**Hands-on Lab:**

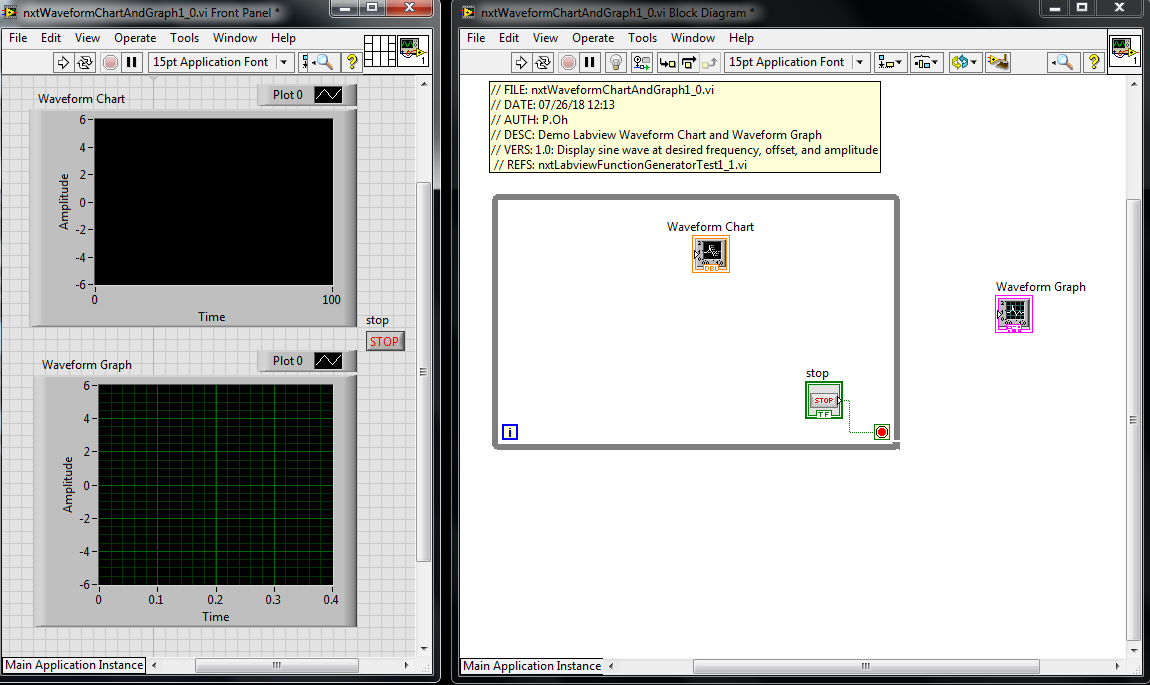
**LabVIEW – Charts, Graphs and Files**

LabVIEW provides Waveform Charts, Waveform Graphs and XY Graphs to display data. Additionally, one can capture data to a computer file that can be opened with Excel. These will be important for future work in tasks like system identification.

**Concept 1:** **Labview Waveform Charts**

**Step 1:** Create Front Panel

Open LabVIEW, File – Save All with nxtLabviewFunctionGeneratorTest1\_0.vi. In your front panel, add a Waveform Chart and Waveform Graph as seen in **Figure 1A (left)**. Re-arrange the controls as seen in **Figure 1A (right)** and then add a while-loop structure and wire the STOP button to terminate the loop. It’s also good practice to comment your code as seen in **Figure 1A (right)**.

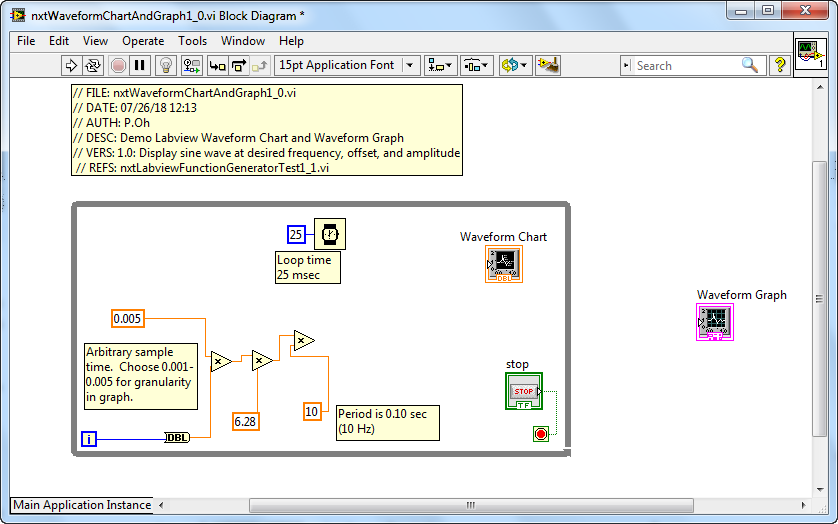


**Figure 1A:** Front panel (left) contains both Waveform Chart (top) and Waveform Graph (bottom). The block diagram (right) shows the controls positions respectively I and out of the while-loop.

**Step 2:** Create Block Diagram

Next, add a timer (MINDSTORMS Robotics – Programming – Time – Wait (ms)) in the while-loop and add a numeric constant. Set the constant’s value to 25. The Wait control synchronizes the while-loop to execute at the specified number of milliseconds.

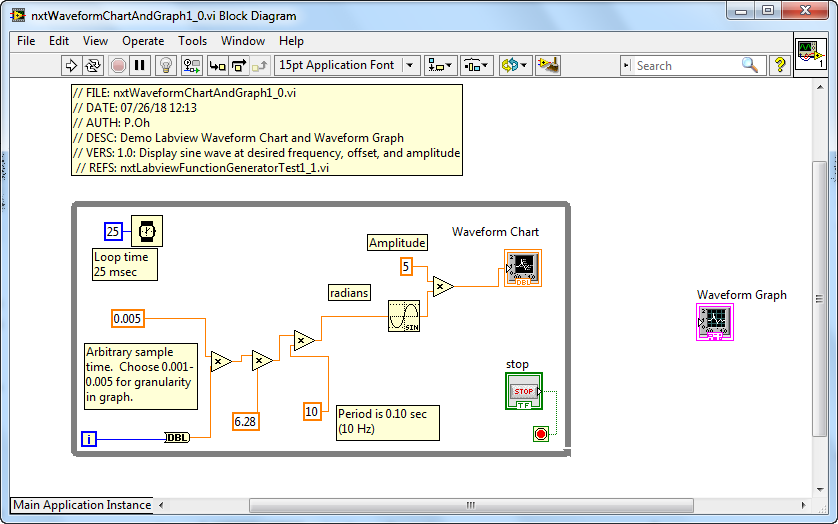
To demonstrate the difference between Charts and Graphs, a sine wave will be used. This wave will be where is the amplitude, is the frequency in Hertz, and is the elapsed time. Create the block diagram as seen in **Figure 2A**.



**Figure 2A:** Add Wait and Multiply controls, and associated numeric constants to the block diagram. These controls are building towards the function.

**Step 3:** Complete Waveform Chart wiring and Test run

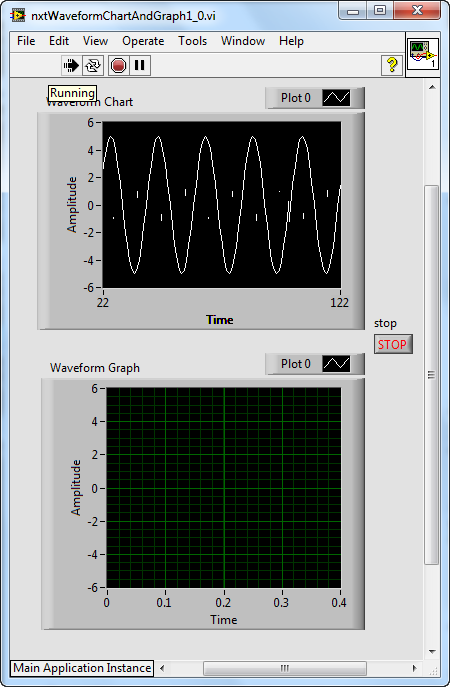
Next, add the sine control (MINDSTORMS Robotics – Programming – Numeric – Trigonometric Functions – sin) and complete wiring as shown in **Figure 3A**. The end result is a sine function with amplitude of 5 and frequency of 10 Hz. Note that the while-loop index variable i multiples a constant of 0.005 to create the elapsed time variable *n*.



**Figure 3A:** Add the sine control and wire to the Waveform Chart

Figure 3 goes here

As a sanity check, running the program should show the Waveform Chart displaying the sine function being continuously generated (see **Figure 4A**). This sanity check reveals the essence of Waveform Charts, namely that it acts much like a real-world strip chart. Such strip charts have paper rolls. A pen simply marks or traces data as the paper unrolls. The horizontal X-axis doesn’t really represent time per se. The paper roll speed is simply set at a constant rate.

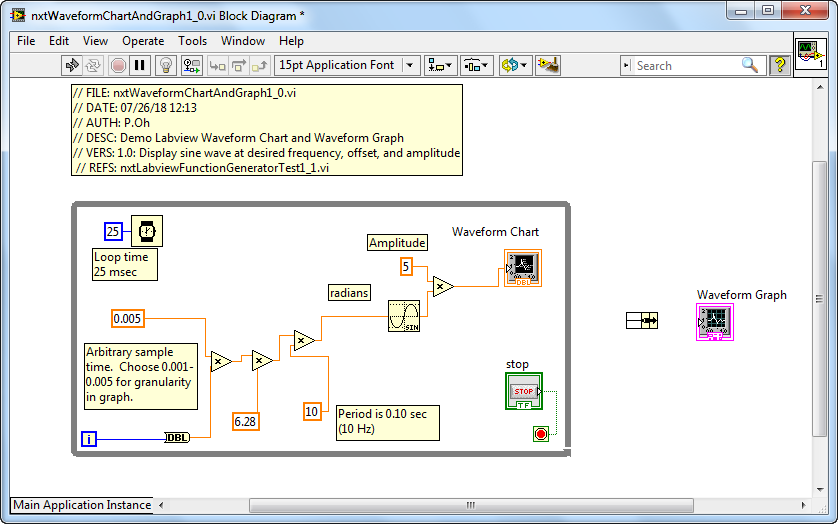


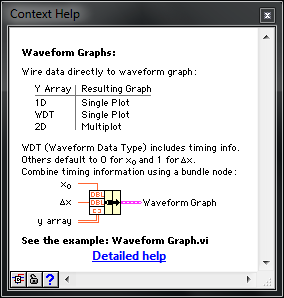
**Figure 4A:** Sample “sanity check”. The Waveform Chart (top) plots the sine function.

**Step 4:** Complete Waveform Graph wiring

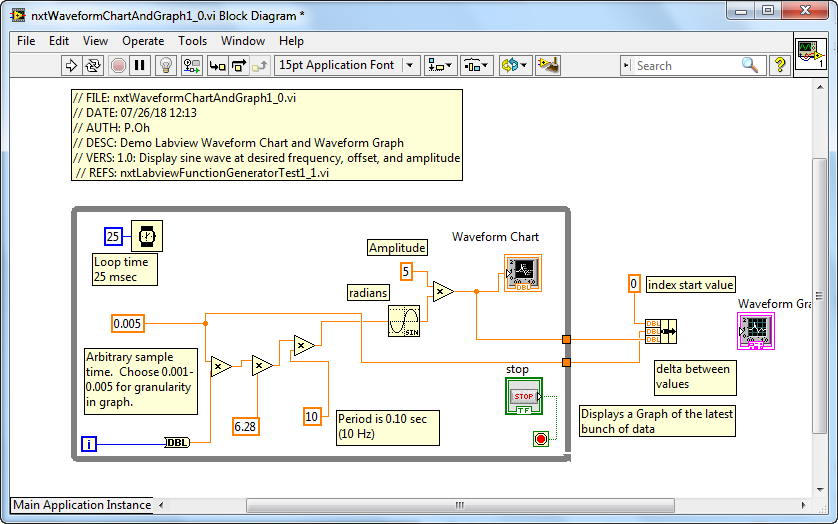
To demonstrate the Waveform Graph, add a bundle control (MINDSTORM Robotics – Programming – Cluster – Bundle) as seen in **Figure 5A left**. Using Context Help on the Waveform Graph shows the necessary inputs (**Figure 5A right**).

**Figure 5A**: Bundle control added before the Waveform Graph control (left). Context Help for Waveform Chart (right) shows expected inputs



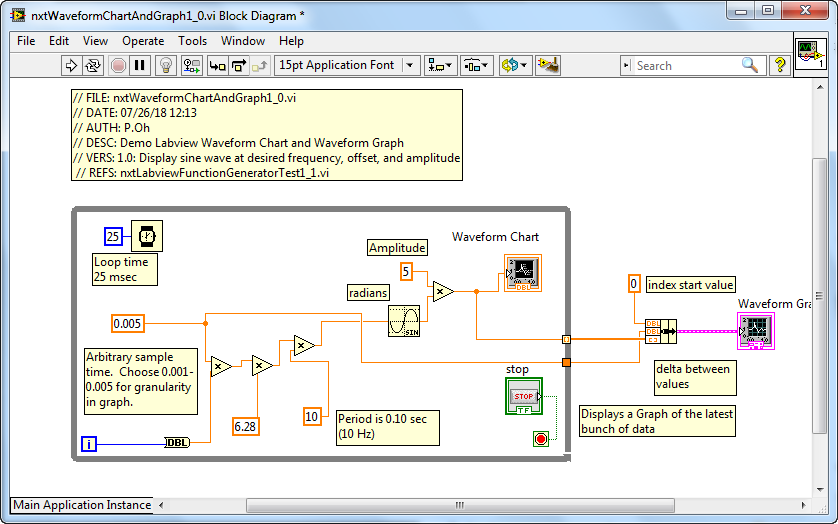


The input is a 3-element bundle that defines the starting time, the sampling time, and output to be plotted. By default, a bundle has 2-elements. One hovers over the bundle and clicks-and-drags vertically down to add more elements. Complete the block diagram as shown in **Figure 6A**. One observes that the starting and sample times are set to 0 and 0.005 seconds respectively.



**Figure 6A:** Wiring of bundle results in two solid orange boxes on the right border of the while-loop.

In **Figure 6A**, one observes two solid orange boxes on the right-side of the while-loop border. In Labview, solid orange boxes represent last known values. The top orange box is the calculated sine function value. Context Help (recall **Figure 5A right**) says that the third element in the bundle is an array. So, one must convert the sine function value into an array value. One does this by hovering over the top orange box, right-clicking and choosing Tunnel Mode – Indexing. The solid orange box turns into an orange box with square brackets in it (see **Figure 7A**). Complete wiring the bundle to the Waveform Graph control (**Figure 7A**).

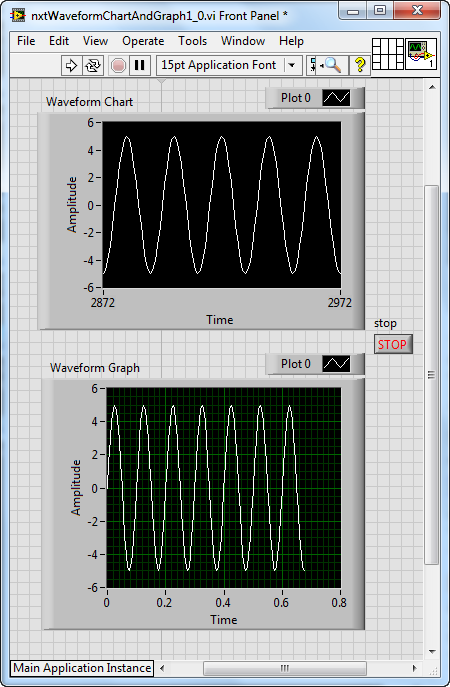


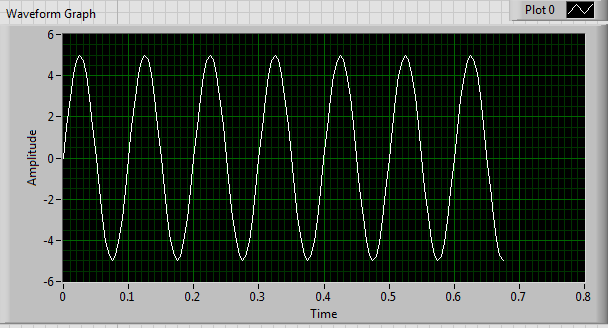
**Figure 7A**: The top orange box turns into one with square brackets in it. This makes the values indexed. This allows final connection to the Waveform Graph.

**Step 5:** Execution and Analysis

First click the Run button to run nxtLabviewFunctionGeneratorTest1\_0.vi. One notices that the Waveform Chart plotting the sine function (like the sanity check from Step 3) but the Waveform Graph remains empty. After a few seconds, click the STOP button. One will then see the Waveform Graph appear as shown in **Figure 8A left**.

**Figure 8A:** One the STOP button is pushed, the collected data is displayed in the Waveform Graph (left). One can re-size this graph (right) to see values in more detail.





One can re-size the Waveform Graph. Doing so (as in **Figure 8A right**) shows values in closer detail. The amplitude is indeed 5 and period is 0.10 sec (or 10 Hz).

A Waveform Graph differs from a Waveform Chart. Here, data must be collected first. A Waveform Graph can then plot this data.

**Exercise 1:** In Labview create programs to:

* 1. Add numeric controls so that you can change the frequency and amplitude of the sine wave
  2. Add a numeric control so that you can have the Waveform Chart display data faster or slower

**Concept 2:** **Labview XY Graphs**

**Step 1:** Create Front Panel

Open LabVIEW, File – Save All with nxtLabviewSimplePlot1\_1.vi.