

BENG402 Lab
September 4th, 2008.

Recap from first week:

During the first week, we went through the LabView 6se tutorials and designed a temperature VI, as well as a function generator. Concepts reviewed included: loops, case structures, frame structures, variables (integers, doubles, strings, arrays); making an executable application (e.g. digital clock).

Part A: Creating sub-VIs, modifying example VIs.

1. Find out, through the LabView help, how to make a sub-VI.
2. Modify the VI you developed last week, temperature.vi, to be a sub-VI. Add an input and an output to it. Input should be the direction (temperature going up or down); output is the temperature (a number: double)
3. Add your own customized icon to it.
4. Start a new VI. Copy the code from figures 1 and 2. In figure 2, the icon with a thermometer is your sub-VI.

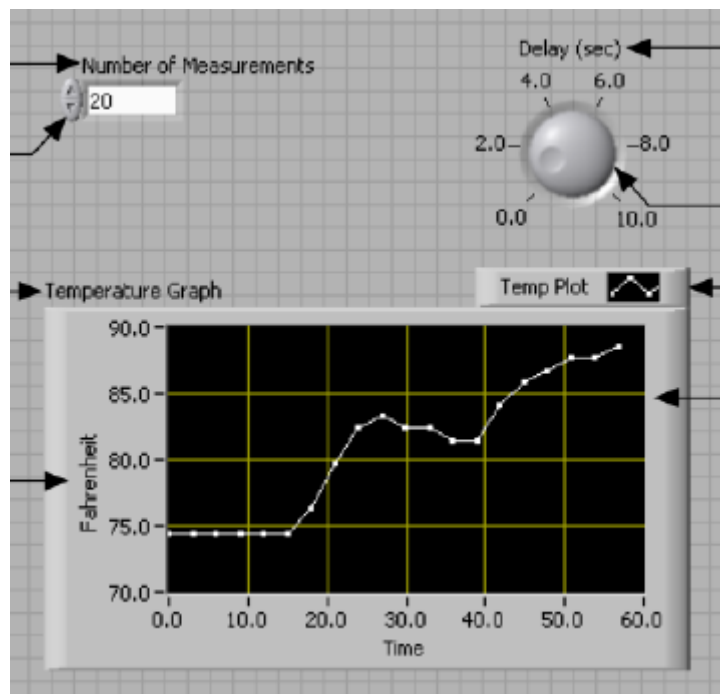


Figure 1. Enhanced temperature VI. Front panel.

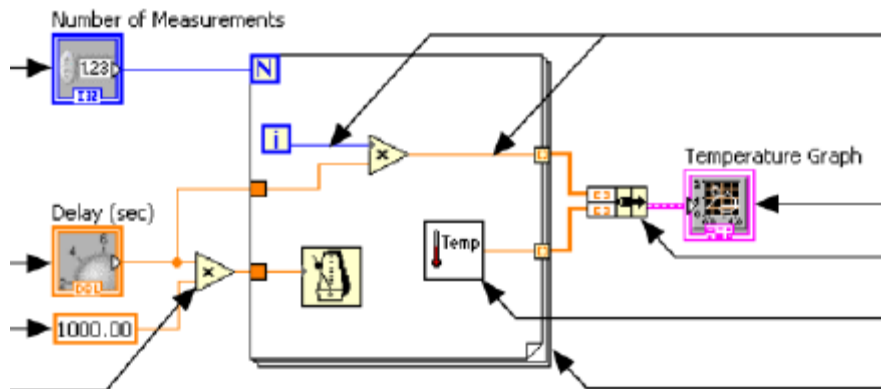


Figure 2. Enhanced temperature VI. Block diagram

5. Now save this data to a file. After the temperature is displayed on the graph, save those numbers, along with the time, to an ASCII file. Make sure you are saving it correctly by opening the file in a text editor. Save the vi (temperature_to_file.vi), along with your name and date.
6. Add a chart to your VI, in parallel to the graph. Explore the differences between the two.
7. Modify the VI so the user can stop it at any time. Make the VI run forever, until the user presses the “stop” button. The file being saved should be appended to in each iteration. Save the vi as temperature_forever.vi, along with name and date.
8. Email these two VIs (from tasks 5 and 7) to you and me (npeixoto)

Part b: Demo (getting acquainted with sensors in preparation for coming labs)

1. Look at signals generated by the accelerometer and the pressure sensors. Find datasheets for both. Write down the accelerometer: _____. The pressure sensors (force sensing resistors, FSRs) are made by Interlink. (<http://www.interlinkelectronics.com>).
2. Later in this course you will use these components to build more sophisticated VIs and mini-projects. Be aware of power requirements, speed, and response time for them.

Part c: Read in a binary file and analyze waveforms.

3. The file testsignals.bin was created with the “write to binary.vi” example from the tutorials you read last week. Your task is to read that binary (1D) file in and display it with LabView.
4. Start by finding the read binary.vi example. Make sure you understand the “write to binary.vi” too. Save an example file, and then read it in.
5. Modify the read binary.VI so you read in only 100k points at a time (per iteration). Add a “next dataset” button which allows you to control the samples which are being loaded to the front panel. Assume the samplerate is 1k (or any other number you would like).
6. What are the three waveforms you see?
7. Which amplitudes do you read?
8. What’s the relative frequency from one to the other?
9. Email me (npeixoto@gmu.edu) the vi from task 3, and the answers to questions 4 through 6.