

# DSP LABORATORY: ANALOG TO DIGITAL AND DIGITAL TO ANALOG CONVERSION\*

Erik Luther

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 2.0<sup>†</sup>

## Abstract

NI LabVIEW DSP is a block diagram-based DSP development platform that allows the user to quickly set up complex DSP algorithms. The true power of LabVIEW lies in its ability to interface with external DSP devices and/or internal sound cards that are installed on the PC. The developed algorithms are downloaded to the DSP board, which then runs the algorithm in a real-time environment. In this lab, we will only scratch the surface regarding LabVIEW DSP's capabilities. For starters, we will look at how LabVIEW DSP interfaces with the A/D and D/A operations of the DSP board. Specifically, we will simply connect the A/D converter to the D/A converter so that the DSP system plays back audio signal sent to it.

## 1 A/D and D/A Conversion

NI LabVIEW DSP is a block diagram-based DSP development platform that allows the user to quickly set up complex DSP algorithms. The true power of LabVIEW lies in its ability to interface with external DSP devices and/or internal sound cards that are installed on the PC. The developed algorithms are downloaded to the DSP board, which then runs the algorithm in a real-time environment. In this lab, we will only scratch the surface regarding LabVIEW DSP's capabilities. For starters, we will look at how LabVIEW DSP interfaces with the A/D and D/A operations of the DSP board. Specifically, we will simply connect the A/D converter to the D/A converter so that the DSP system plays back audio signal sent to it.

### 1.1 Simple Sampling/Reconstruction System

1. Keeping the Surround Mixer open, start LabVIEW Embedded Edition. On the startup screen from the Execution Target dropdown menu select your hardware device. Refer to Figure 2 below for details. Execution target determines where your application will execute when you run it.
2. Note: All the screenshots of LabVIEW windows are taken with the SPEEDY-33 selected as the execution target. However, the instructions still apply for any other target unless otherwise specified.

---

\*Version 1.1: Sep 20, 2005 1:17 pm -0500

<sup>†</sup><http://creativecommons.org/licenses/by/2.0/>

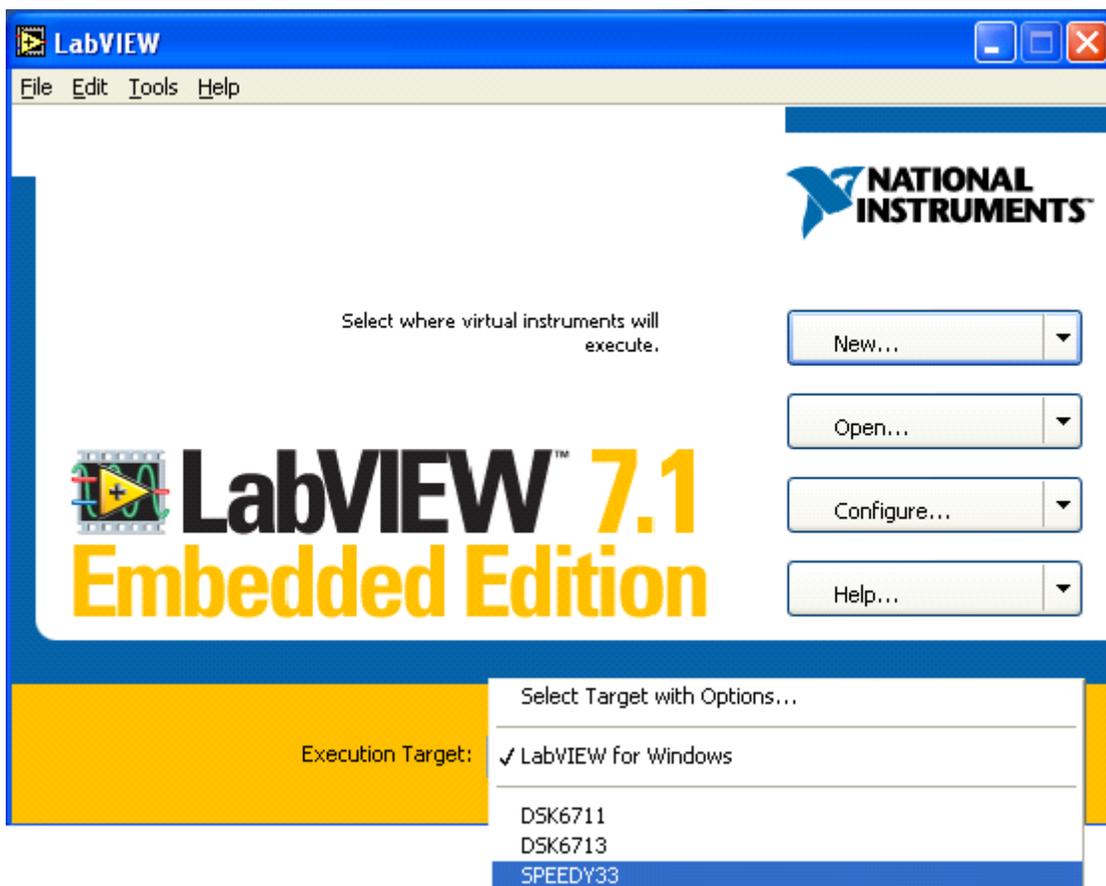
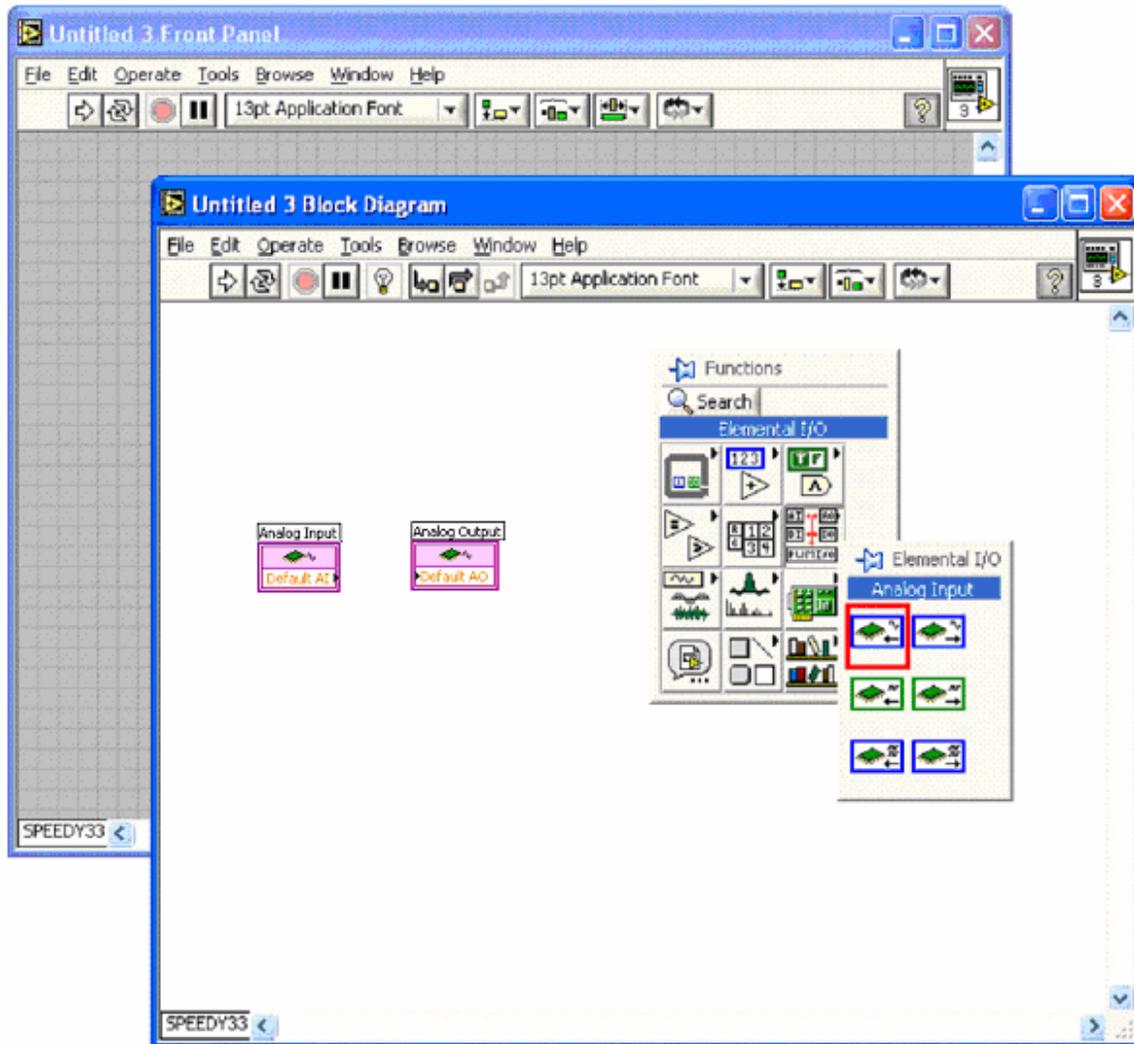


Figure 1: Switch Execution Target to your DSP hardware device

### 1.1.1 Start With a New Blank LabVIEW VI

1. Select File»New VI to start a new LabVIEW application. (Note: A LabVIEW application is called Virtual Instrument or VI). Any LabVIEW VI consists of two windows; a Front Panel where you will design your user interface and a Block Diagram where you will add various block components or VIs and wire them together to design your DSP system. On starting a new VI you will see the Front Panel and Block Diagram windows for your application appear on the screen.
2. Switch over to the Block Diagram window by left clicking on it and start developing your application. You will now develop a simple VI to read in an analog input from the A/D channel and play it back using the D/A channel. From the menu select Window»Show Functions Palette. The Functions pallet contains functions or VIs that you will wire together to develop the application. Click on the Elemental I/O palette and bring out the Analog Input Elemental I/O node and place it on the block diagram. This node is used to read in analog input from the A/D channel. Similarly place the Analog Output Elemental I/O node on the block diagram. Refer to Figure 3 below for details.



**Figure 2:** Analog Input and Analog Output Elemental I/O nodes.

3. The Analog Input and Output nodes control the input/output flow of sound waves/data between the analog input channels and the DSP. The Analog Input node controls the A/D converter, which converts the analog input signal (from the microphone, function generator, or any other analog sound source) to a sequence of numbers (digital data) and passes the data on to the DSP. The analog Output node controls the D/A converter, which takes the digital data from the DSP and converts it back to an analog signal.
4. Now we need to configure the Analog Input and Analog Output nodes. Double click on the Analog Input node to bring up its configuration dialog. In this dialog specify the following setting:
  - General Tab:
  - Resource: 2 channel multiple samples
  - Configuration Tab
  - Sample Rate in Hz: 48000
  - Framesize: 256

5. Click OK when done.

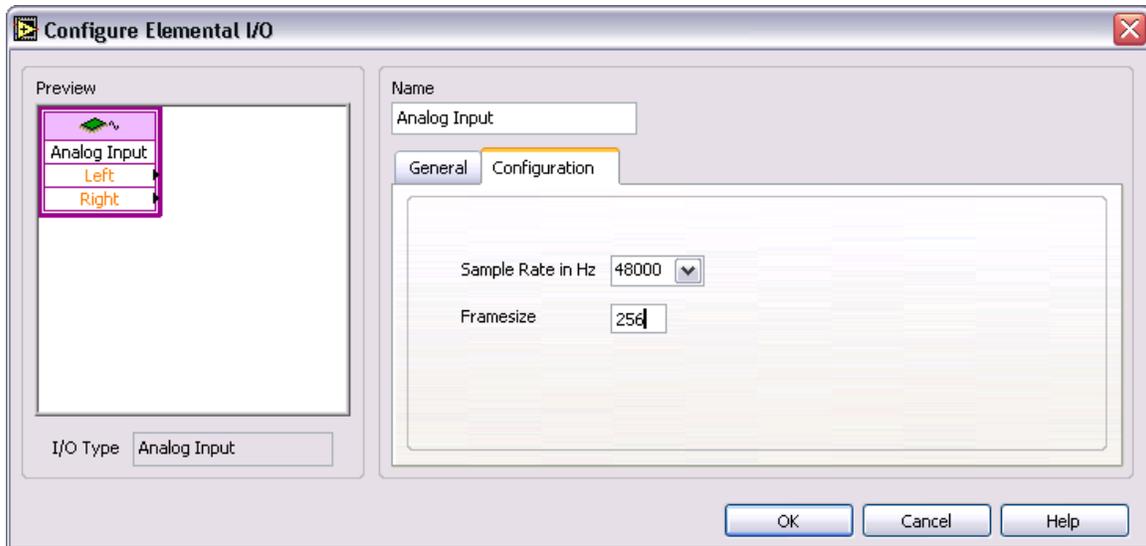


Figure 3

6. Double click on the Analog Output node. On its Configuration dialog specify the parameters as follows:

- General Tab:
- Resource: 2 channel multiple samples
- Configuration Tab
- Sample Rate in Hz: 48000

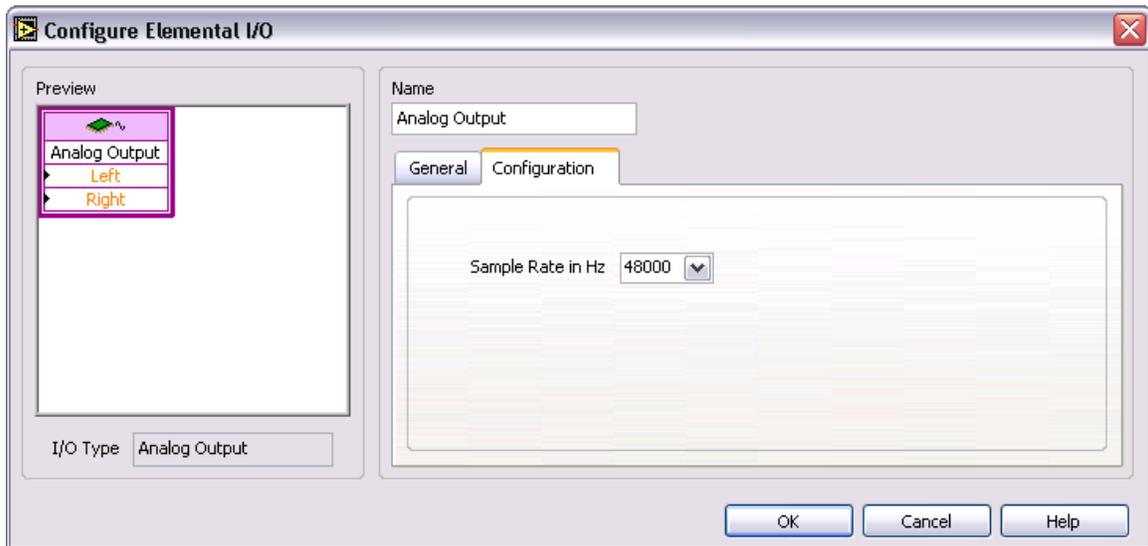
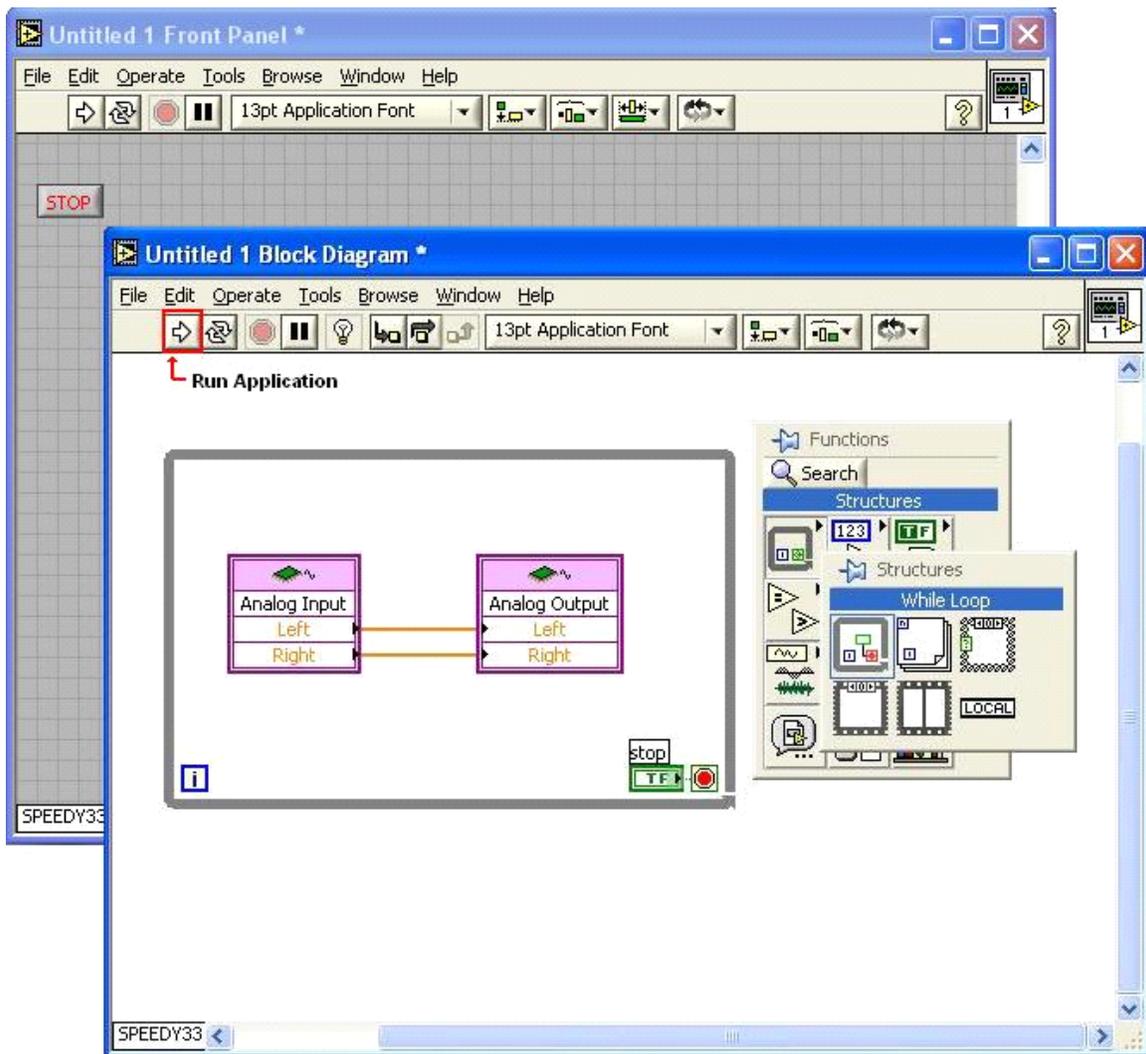


Figure 4

7. Before running any DSP algorithm, the Analog Input and Output nodes need to be wired together. Hover your mouse over the Left channel output of the Analog Input node, the mouse cursor will change to a wire spindle. Left click and drag the wire to the Left channel input of the Analog Output node.
8. We will now place a While Loop so that the process can be run in a loop. Bring out a While Loop from the Functions»Structures Palette. Place the While Loop around the two nodes by left clicking on the top left corner and dragging it to the bottom right. The final application will look like Figure 4. Notice a stop button appears which is wired to the conditional terminal of the loop. You will also see a stop button on the front panel. Once the application is run you will press this button to stop the application.



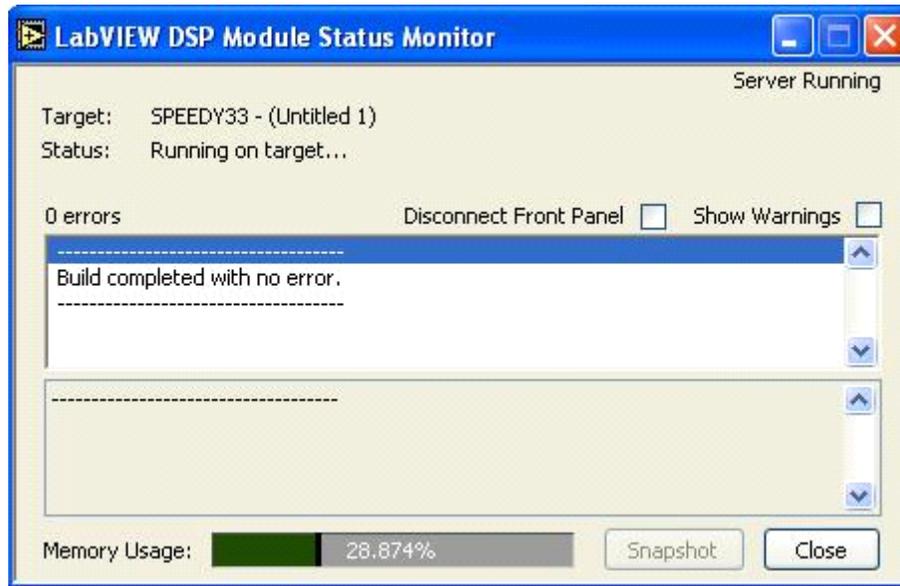
**Figure 5:** Completed Application

9. Save your VI by selecting File»Save As and storing it in the directory you created earlier. You can give the file any name you want (such as lab1setup), but use the default extension of vi.
10. Now you need to prepare an audio input for your system. First, go to the Surround Mixer and activate the Wave/mp3 input. Place a CD in the CD player in the PC, select the Windows Media Player program, and press play. (Note: The program may automatically start when you insert the CD). You should not hear anything yet, because you haven't compiled and downloaded the simple playback program to the DSP yet.

### 1.1.2 Run LabVIEW Code On The DSP

1. You're finally ready to run this very simple DSP system. (If using the SPEEDY-33 make sure that the Jumpers are in the Line position). To run the application left click on the run arrow on the toolbar as shown in Figure 4. Notice on the LabVIEW DSP Module Status Monitor window comes up and

displays the Status as the application is compiled and downloaded onto the DSP. The Memory Usage bar at the bottom displays the amount of memory being used. Once Status displays Running on Target you should hear the music through the headphones.



**Figure 6:** LabVIEW DSP Module Status Monitor

2. To use the microphone input instead of the CD Audio input, simply mute the Wave input button on the Surround Mixer and activate the Microphone input instead. Give it a try. You can very quickly toggle between (or even combine) different inputs by using the Surround Mixer.
3. Stop the program by pressing the Stop button on the Front Panel.
4. Effects of Sampling Frequency on Sound Quality
5. Now we'll briefly examine the effects of sampling frequency on the sound quality of both music and voice signals.
6. Run the program again with CD audio input (using the original sampling frequency of 48000 Hz.) and note the sound quality.
7. Stop the program and change the sampling frequency (in both the Audio Input and Audio Output nodes) to 18000 Hz. Run the program again and listen to the sound quality. Repeat with a sampling frequency of 8000 Hz.
8. Repeat steps 1-2 but use the microphone input instead of the CD input.

## 1.2 Answer These Questions

### Exercise 1

Describe how sampling frequency seems to affect the sound quality for both speech and music. Based on your observations, which seems to be most affected by sampling frequency, music or speech?