Introduction To LabVIEW and the DSP Board

1 Overview

The purpose of this lab is to familiarize you with the DSP development system by looking at sampling, analysis, and reconstruction of continuous-time signals. We start with the sampling and reconstruction of continuous time signals. We will then look at displays in the time and frequency domains. We also examine the importance of the sampling frequency and its effects on aliasing.

2 DSP development overview

2.1 PC Details

The PC is required to run the LabVIEW DSP module which is required to design and implement and download code to (run on) the DSP.

2.2 DSP Board Highlights

A self-contained, high-performance, programmable DSP board for signal processing applications. The SPEEDY-33 comes equipped with a Texas Instruments DSP for high speed DSP calculations and Analog to Digital (A/D) and Digital to Analog (D/A) converters.

- High-performance 32-bit floating-point digital signal processor (DSP)
- 150 million floating-point operations per second (MFLOPS)
- 13-ns Instruction Cycle Time
- 16/32-bit integer and 32/40-bit floating-point operations
- 32-bit instruction word, 24-Bit Addresses
- Parallel arithmetic/logic unit (ALU) and multiplier execution in a single cycle
- Supports standalone operation

The SPEEDY-33, which connects to a PC through a USB host port is programmed in LabVIEW. The program as well as other data such as tables, sounds etc. are stored in the flash memory. Once the FLASH is programmed, the SPEEDY-33 can run in standalone mode, without the PC.

The TMS320C33 is one of the SM320C3x generation of floating-point DSPs from TI. The SPEEDY-33 is a self-contained, programmable board for signal processing applications.

2.3 Software

The software used in the lab is National Instruments LabVIEW DSP Module; a block diagram-based graphical programming package for digital signal processing that will be used for: data acquisition, time domain data display, frequency domain calculation and display, filter implementation, convolution and correlation
calculations, speech/signal modeling, and real-time filtering. Using LabVIEW DSP Module along with the LabVIEW Digital Filter Design Toolkit and the DSP board allows you to perform data acquisition, analysis, and filtering, it will also give them the ability to listen to the filtered output.

3 System Setup

Follow the instructions below to set up the system correctly.

- Log on to the computer
- Connect a USB cable between the PC USB Port of the SPEEDY-33 (Figure 2-7) and the USB port of your PC. The SPEEDY-33 is powered by the USB port. Once you connect the cable, the green Power LED (Figure 2-2) on the SPEEDY-33 will light.
- Confirm that the Power LED on the SPEEDY-33 is lit.
- Connect the speakers to the Audio Stereo Output Port (Figure 2-4) on the SPEEDY-33.
- The on-board microphones will be the input to the DSP board.

4 A/D and D/A Conversion

The NI LabVIEW DSP module 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 real time environment. In this lab, we look at how LabVIEW DSP interfaces with the A/D and D/A converters on the DSP board. Specifically, we simply connect the A/D converter to the D/A converter so that the DSP plays back audio signal sent to its input.

4.1 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. The Execution Target specifies where the code will run. Refer to Figure 3.
Figure 2: SPEEDY block diagram

Figure 3: Switch Execution Target to SPEEDY 33
4.2 Start With a New Blank LabVIEW VI

1. A LabVIEW application is called a Virtual Instrument or VI. Every LabVIEW VI consists of two windows: a Front Panel which is the user interface and a Block Diagram which contains the blocks inherent to the application. On starting a new VI you will see the Front Panel and Block Diagram windows for your application. Refer to Figure 4.

2. Switch to the Block Diagram window. You will 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 WindowShow Functions Palette. The Functions Palette contains functions (VIs) required 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 5.

3. The Analog Input and Output blocks control the flow of data between the analog input/output channels and the DSP. The Analog Input block is equivalent to an A/D converter. The Analog Output block is equivalent to the D/A converter.

4. We need to configure the Analog Input Output blocks. Double click on the Analog Input (Output) block to bring up its configuration dialog. In this dialog specify the following setting(s):

   Analog Input General tab
   - Resource: 2 channel multiple samples

   Analog Input Configuration tab
   - Sample Rate in Hz: 48000
   - Framesize: 256

   Analog Output General tab
   - Resource: 2 channel multiple samples

   Analog Output Configuration tab
Figure 5: Functions Palette with the Elemental I/O blocks
5. Before we finish, the Analog Input and Output blocks need to be wired together. Hover your mouse over the Left(Right) channel output of the Analog Input block, the mouse cursor will change to a wire spindle. Left click and drag the wire to the Left(Right) channel input of the Analog Output block.

6. We require the VI to run continuously. To do this, we use a While Loop. Place the While Loop (in the FunctionsStructures Palette) around the two nodes by left clicking on the top left corner and dragging it to the bottom right. The final VI should look similar to Figure 6. The Stop button that appears is wired to the conditional terminal of the loop. This also appears on the front panel. Once the VI is run, this button aborts execution.

7. Save your VI by using the File >> Save As dialog.

8. Your audio input will be the on-board microphones.

4.3 Run LabVIEW Code On The DSP

1. When you press Run, you should see something like Figure 7.

2. What happens when the framesize and/or sampling frequency are changed?

3. Does sampling freq. more affect music or speech?
5 Time and Frequency Displays

LabVIEW has the ability to display data in the time and/or frequency domains.

5.1 Time Domain

When an analog signal is passed through an A/D converter, its samples are stored as an integer. If we use a 16-bit system, there are $2^{16} = 65,536$ possible values that can be stored. These stored values can be displayed either as discrete samples or as a smooth line connecting the samples. We use the Waveform display to set up a time domain display. Switch to the Front Panel and right click anywhere to bring up the Controls palette. From the Graph palette up the Waveform Chart and place it on the Front Panel, as shown in step 1 of Figure 8. Right click on the Waveform Chart and select Properties from the popup menu as shown in step 2 of Figure 8. In the Chart Properties window on the Appearance tab change the Label to Left Channel Time Domain. Specify the settings as shown in step 4 of Figure 8. Deselecting Autoscale will ensure that the Y-axis of the chart does not change with the data values. Minimum and Maximum values of -32000 and 32000 will cover all the values for a 16-bit A/D channel.

The Waveform Chart maintains a buffer (History) of values that are plotted on it. The number of points stored can be adjusted by right clicking on the chart and selecting Chart History Length from the popup menu.

1. Connect the left output of the Analog Input block to the Waveform Chart as shown in Figure 9.
2. Run the program and speak into the microphones. You should be able to see your voice, in addition to hearing it.
3. Stopping the program will freeze data on the Waveform Chart.

5.2 Displaying the magnitude of the Fourier Transform

1. To display the frequency content of the audio signal use the Fast Fourier Transform (FFT). Add a second Waveform Chart Display to the Front Panel. Change the label of this Chart to Left Channel Frequency Domain and enable Autoscale for the Y-axis by checking the box in the Waveform Chart Properties window on the Scales Tab. On the Scales Tab switch over to the X axis properties by selecting it in the dropdown menu and change the X-axis range to 0-255.
Figure 8: Time Domain Display

Figure 9: Block diagram with time domain display
2. In Block Diagram, resize the while loop as shown in Figure 10. Place the Spectral Measurements Express VI from the Functions>>Signal Processing>>Frequency Domain Sub-Palette.

3. Connect the blocks as shown in Figure 11.

4. Speak into the microphone and look at frequencies in the signal.

5. The frequency spectrum is always displayed from 0 Hz (DC) to Fs/2. For voice signals, which have most of the energy ≤ 2000 Hz, the voice frequencies only cover a small portion of the display (assuming Fs = 48000 Hz). To get a more detailed view of your voice spectrum, change the sampling frequency to 8000 Hz (in both the Analog Input and Output blocks). Re-run the VI.

6. Speak into the microphone using a single held-out note (like AHHHHHHHHHH or EEEEEEEEEEE). Examine both the time-domain and frequency domain displays. If necessary, change the Chart History Length so that about 4-5 periods of the signal are on the time-domain display. Because a note like this is periodic, you should notice a fairly steady time signal as well as a well-defined frequency spectrum consisting of large peaks at your notes fundamental frequency and its harmonics. Change the frequency (pitch) of this note and examine how it affects the frequency spectrum. Try experimenting with different notes. Now, hold your note and stop the program to freeze the displays. You can now zoom into various parts of the spectrum to examine it closer.
Figure 11: Final block diagram