ECEn 487 Digital Signal Processing Laboratory

Lab 1
Introduction to the Software Development Environment and Signal Sampling

Due Dates

This is a two-week lab. All TA check offs must be completed before 3:00 p.m. January 31.

Laboratory report submission, beginning of lab class:

Friday, January 31

Objectives

The purpose of this lab is for each student to become familiar with the MATLAB data acquisition and processing environment, and to study the effects of signal sampling, aliasing, and basic signal input/output and plotting operations.

Introduction

In many real-world DSP applications a specialized microprocessor called a "Digital Signal Processor" is used to implement the structured, repetitive, high speed mathematical operations needed for "real-time" operation. Other high-speed DSP processor options include field programmable gate array (FPGA) implementations, or graphics processor units (GPUs) which were originally designed for high speed graphic display calculations. For lower sample rates (i.e. in the audio range) and fewer input/output channels (e.g. stereo) most modern PCs are capable of keeping up with some significant signal processing in real time. We will use the lab PCs as our real-time platforms, even though the Windows operating system is not well-suited for such applications.

You will be using the built-in sound card of you lab PC as your data acquisition (ADC) and output (DAC) device. Processing will be performed using MATLAB on your general-purpose microprocessor. MATLAB does have some limited provisions for real-time interfacing with the sound card, and though MATLAB is a relatively slow computational environment (due to its default double precision floating point arithmetic and its implementation as an interactive interpreter) we will generally be able to keep up with the sample rate. We will provide you with some template code for FIFO frame-buffered DSP input/output.

This lab is intended to familiarize you with the analog signal sampling and output functions in MATLAB, and to help you understand some of the implications of decisions you make in designing a data acquisition system.

Reading Assignment

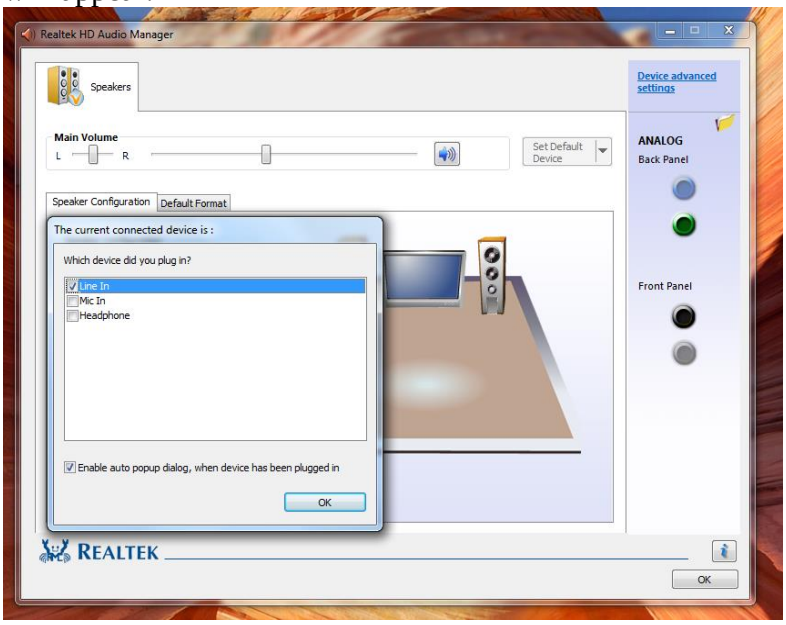
1. MATLAB Help documentation for the following functions: dsp.AudioRecorder, dsp.AudioPlayer, step, upsample, downsample "debugging with the debugging window," and optionally: "creating graphical user interfaces."

<u>Task 1</u> Familiarization with MATLAB ADC - DAC Operation: loopback program.

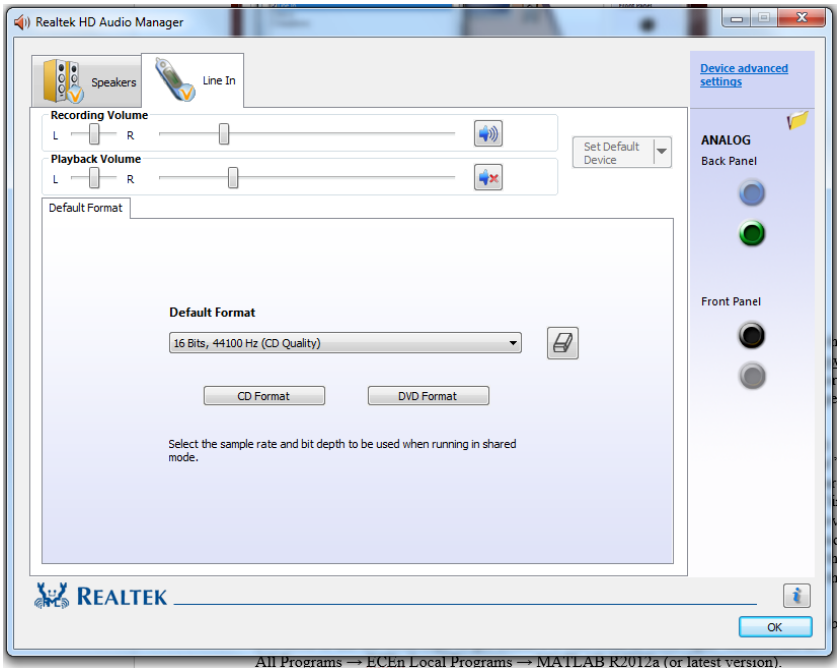
In this task you will write a MATLAB script to sample an audio function generator using the built-in sound card, and play the signal out to speakers. This loopback program will be the basis for future code development.

Procedure

1. On the front panel of your computer are a power button, 4 USB ports, a microphone/headphone jack, and a headphone jack. Take the special cable with the red and white BNC connectors on one side and a microphone plug on the other out of the box. Insert the microphone connector into the mic/headphone jack. The following screen will appear.



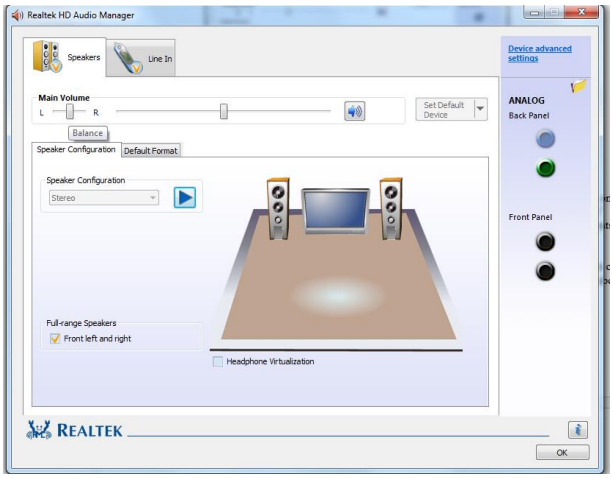
Select the check-box for “Line In” and then press the button for OK.



All Programs → ECEn Local Programs → MATLAB R2012a (or latest version).

Then, you can select "16 bits, 44100 Hz (CD Quality) for the Default Format. Also, on the "Line In" tab, disable the speaker icon on the right side of the "playback volume" slider. This keeps the input analog signal from bleeding through to the speaker outputs. Make sure the Recording Volume is set to 50.

2. Plug the other microphone/red/white BNC cable into the headphone port on the front of the computer. A notification icon should appear in the bottom portion of the screen above the orange speaker icon. Right click this icon and select "Sound Manager". Now the settings of the speakers appear. Adjust the "Main Volume" slider so that it has a gain of 66.



3. Now connect a BNC "T-connector" to the DSO3102A oscilloscope port 1. This will allow you to see the signals that are being presented to the audio input. Then connect a

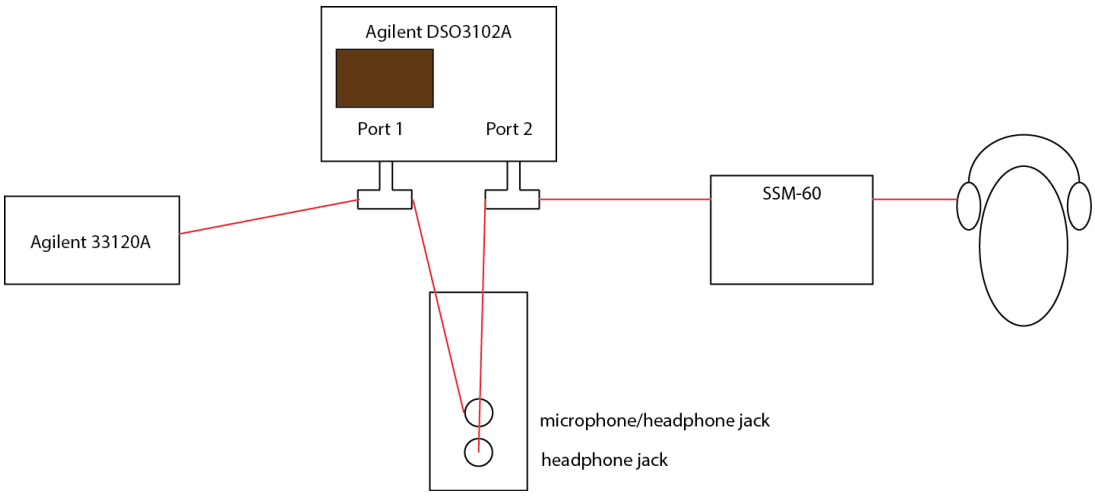
short BNC-BNC cable from the output of the Agilent 33120A Function Generator to one end of the "T". Turn on the oscilloscope.

4. Turn on the 33120A function and have the function generator output a 100 mVPP sine wave at 1 kHz. Observe this on the oscilloscope. Note that the output impedance of the function generator is 50 Ohms and the input impedance of the oscilloscope is 1 MOhm. Thus, you will probably see 200 mVPP on the oscilloscope. Change the output impedance of the function generator by pressing the "Shift" then the "Menu" key. Hit the ">" button until you see "D: SYS MENU" appear. Then press the "\/" button twice until you see "50 OHM" or "HIGH Z". If the unit says "50 OHM" then hit the ">" button once and "HIGH Z" will appear. Then hit the "Enter" button and the unit will now be set to a High output impedance configuration. IT IS VERY IMPORTANT TO KEEP SIGNALS BELOW 1 VPP TO AVOID DAMAGING THE SOUND CARD.

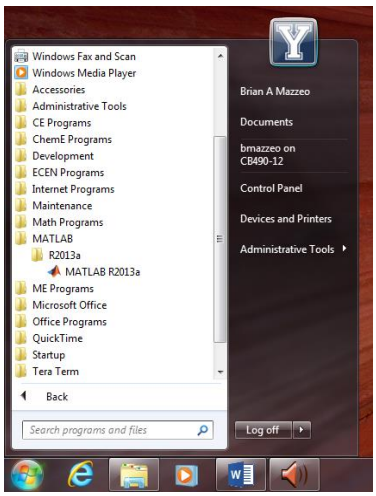
5. Now that you are outputing a 100 mVPP, 1 kHz signal from the waveform generator and have verified it on the oscilloscope, now connect the white terminal from the microphone input to the other end of the "T" attached to port 1 on the oscilloscope.

6. Connect another "T" to port 2 on the oscilloscope. Now connect the white headphone output from the computer to the "T" on port 2 of the oscilloscope. All you should see on this second output is noise at this point.

7. Connect a BNC-to-RCA cable from the "T" on port 2 of the oscilloscope. Connect the RCA cable to the line input on the Radio Shack SSM-60 Stereo Stound Mixer. Also plug in the Sound mixer using the transformer. Also, plug in a set of headphones to the "PHONES" input on the board. Make sure that the Source for Monitoring is set to Source 4. This setup will allow you to listen to the signals that you are monitoring. Your completed setup should look like the following schematic.



8. Launch MATLAB from the Start Menu.



9. Make sure that you write and save all your scripts and data to either a removable USB memory stick, or to your personal CAEDM account network mounted J: drive. To insure this you must browse to your folder using the ". . ." button on the "Current Folder" window of the MATLAB desktop. If you write your files on the local machines hard drive, it may be erased when you return, or you may not have access to that machine next time. Place the "loopback.m" file from the course website into a folder where you will have access to it. Open the file in the MATLAB editor.
10. Use the provided "loopback.m" MATLAB script which samples the line input in stereo, then plays that block through the DAC to an output such as amplified speakers or your RadioShack amplifier board. You modify your code to adjust the specified sample rate, sample window block length, and how long it will run before terminating. Read the MATLAB help documentation on the dsp.audioRecorder and dsp.audioPlayer functions.
11. Evaluate how the loopback program operates using the function generator as you input. Use the sound card sample rate of 48 ksamples/s. Try different sinusoidal frequencies and other waveforms. For this task, and all experiments below, observe your input and output signals simultaneously with two channels of the oscilloscope. Note differences you observe. Do this for a rate of 10 Hz and observe the delay between the input and output. Why is this delay present? How is this affected by the frame size and sample rate?
12. Provide a MP3 player input (from your phone/player) to the line connection on the line-in back panel connector. Use the sample rate of 48 ksamples/s. Verify proper digital-audio loop input and output and note what volume levels lead to distortion. Can you detect any degradation in sound quality through the loopback? Should there be any noticeable change? Document your observations in your lab book.

<u>Task 2</u> Use of MATLAB debugging tools.

Procedure

1. Practice using the MATLAB built-in debugging tools to evaluate your code's operation, including setting breakpoints, inspecting data array values, and single stepping.
2. Practice your use of breakpoints to stop the loopback script after acquiring the 5th window of data. Plot the sampled data window, then continue non-stop operation.
3. Document your observations in your lab book. Include tips and instructions to help you build your next program.

<u>Task 3</u> Build a digital oscilloscope.

Procedure

1. Write a MATLAB script to implement a two channel digital sampling oscilloscope using the parts of the loopback script and the "plot" functions. (If you want to have your scope actually operate in real-time, which is possible to do, don't actually use the "plot" function. You can use the "line" function or update the data in the plot and then use the "drawnow" command). The sample rate should be fixed at 48 ksamp/s, and the plot window will have 480 points. Make the sample window block 0.5 seconds long. The choice of which 480 samples in the block are plotted depends on the trigger detection and time base operations described below.
2. Provide a triggering function where your code detects the input signal (say on the left channel) crossing a threshold amplitude level. The first sample in the data block that crosses the threshold becomes the first sample in your plot. This insures that successive plot windows properly align the signal waveform. You will need to detect the difference between a rising and falling slope as it crosses the threshold. Single sweep and continuous run options must be demonstrated. If no crossing is detected in the widow block, set the first sample as the trigger sample.
3. Provide user-controlled ability to set internal digital gain (scaling) levels for the two channels. The plot axis scale will need to be set to fixed values (rather than the default auto scale) depending on your gain setting.
4. Provide user-controlled selection of time base for the horizontal sweep. Required setting options are 0.2, 0.1, 0.05, 0.02, and 0.01 seconds to span the horizontal axis of the full 480 sample plot. For example, when the 0.1 second setting is chosen, the plot displays only every tenth sample in the data block, beginning with the trigger threshold crossing sample. <u>Demonstrate your working oscilloscope to the TA.</u>
5. Optional (no extra credit, but it is cool if you can do it). Use GUI interface functions built into MATLAB to implement the user control settings for channel gain, trigger level, and time base.

Task 4 Sampling principles

Procedure

1. Modify your program to scale the audio data by a user specified fixed scale gain factor. Run the program with CD audio input and scale gains of 0.1 and 100. Record in your lab book what you observe. Explain what happens and why there may be distortion.
2. Using a combination of the volume control setting on your audio source, and the line input setting in the Windows recording control panel, set the signal level very low so that you can just hear it by using a scale-up factor of about 5,000 in your loopback code. Do you hear any noise? What is causing it? Why can't you simply compensate for an arbitrarily low level input signal by using a sufficiently high internal digital gain factor?
3. Modify the sample rate to 2 kHz, 6 kHz and 12 kHz. The sound card is pretty smart and won't allow aliasing on the input. Thus, to effectively change the sample rate, you want to instead downsample the data digitally. Leave the input sample rate at 48 kHz. Then change the output by applying the command downsample(y_data, factor) and also change the sample rate of the output to the appropriate frequency. Now use different sinusoids from the function generator on the input, say 1 kHz, 2 kHz, 3 kHz, etc. and listen to the output. You may have to trigger on channel 2 on the oscilloscope to see what is happening. Document the aliasing phenomena and the observed signals.
4. Now perform digital upsampling on the signals. Change the input sample rate to 2 kHz, 6 kHz, and 12 kHz (You can do this with aliasing if you use the above procedure, or you can actually change the sample rate and you won't get aliasing). Then use the upsample command in MATLAB to digitally add in zeros to the signal. Change the corresponding output rate to 48 kHz or other values. Document the upsampling phenomena and the observed signals.
5. Run the program for each case with CD/MP3 audio input and record in your lab book what you observe. Is there any signal quality change? Distortion? Explain what happens. Demonstrate your running program to the TA.

Conclusions

Write a paragraph or two of conclusions for your lab experience. Discuss any additional implications of what you observed. Describe what you feel are the important principals demonstrated in this lab, and note anything that you learned unexpectedly. What debug and redesign procedures did you need to perform to get it to work?

Laboratory 1 TA Check-off Page (to be submitted with laboratory report)

Student Name: _____

_____ Task 3.4 – Demonstrate your working oscilloscope to the TA.

_____ Task 4.5 – Demonstrate downsampling and upsampling to the TA with aliasing.