# ESE 150 – Lab 01: Sampling and Quantizing Audio Signals

LAB 01

In this lab we will do the following:
1. Listen to pure audio tones produced by a function generator
2. View and capture pure audio tones using an oscilloscope
3. Use a microphone to convert pressure to voltage
4. View and capture voltage produced by microphone
5. Build an analog-to-digital converter using an Arduino microcontroller

Background:

In lecture, you learned that sound is a vibration of particles in the air. A microphone is a device (transducer) that converts that vibration into voltage. If we sing a pure tone into a microphone and plot the output of the microphone with time, we'd see a sinusoidal pattern as shown in the figure below. We note the amplitude of the voltage is changing as time progresses. We recall that this voltage is continuous with time.
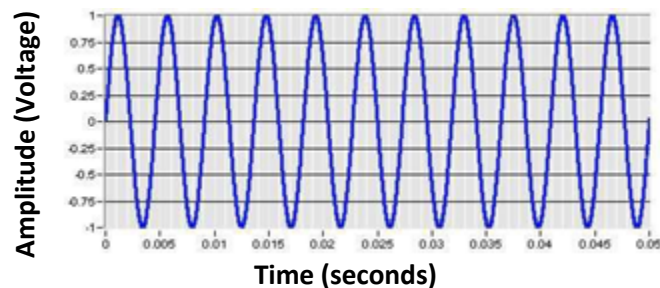
In lecture, you also learned that in order to "digitize" this signal, we need to break it up into discrete segments, that the process of digitization is taking a continuous signal and converting it into a discrete signal that can then be manipulated by a digital computer. A device known as an analog-to-digital converter (ADC or A2D) is the component needed to digitize our signal. An A2D will have a sampling rate: meaning how the x-axis (time) will get partitioned. An A2D will have a quantization value, meaning how the y-axis (voltage) will get partitioned. The A2D will then take a "sample" or measurement of the voltage at regular intervals and produce a digital representation of those samples.

In lab today, we'll take in audio using a microphone and then build our own A2D using an Arduino microcontroller to digitize a continuous audio signal!

ESE 150 – Lab 01: Sampling and Quantizing Audio Signals

Lab – Prelab:

1. What is the relationship between Sampling Frequency and Delay between samples? [provide a symbolic equation] For context, read through Section 3 of the Lab.

2. Complete the following table (you will need these values for the later part of Section 3)

Frequency	Delay
500Hz	
1000Hz	
5000Hz	

3. Read through Section 3. Use the Arduino Language reference guide (https://www.arduino.cc/reference/en/) as necessary to understand the operation of the code in Section 3.

When you arrive in lab, compare your answers with your assigned partner. If your answers differ, discuss amongst yourselves and try to resolve your differences.

Early during the lab session, a TA will check you off on prelab. Go ahead and start working on the lab.   It will take some time for the TAs to get around to all the groups.

ESE 150 – Lab 01: Sampling and Quantizing Audio Signals

Lab Procedure:

Lab – Section 1: Visualizing Audio from a microphone

- An oscilloscope is a piece of lab equipment that can measure voltage vs. time.
- In this section of the lab, you'll attach a microphone to an oscilloscope and view the output of the microphone.
- A "power supply" is a piece of lab equipment that can provide power to turn on equipment, we'll use it to "power up" a microphone.
- The various pre-manufactured wires are identified by their connector ends: BNC, banana, and grabber/alligator, as shown below:

BNC Splitter   Banana   BNC   Alligator/Grabber

1. Obtain a microphone from your TA or lab staff
2. Obtain a breadboard (it's a white board with holes in it at your workstation).
   a. A breadboard allows you to prototype by connecting various electric components when inserted.
   b. Each row is designated with a number (1-62).
   c. The five holes in each row are electrically connected.
   d. The long columns on the sides of the breadboard, marked by a + or -, are the power rails of the board, where each column is electrically connected.
3. Attach the microphone to the breadboard as follows:
   a. You'll notice the microphone has electrical pins on the bottom of it.
   b. Insert it into the breadboard as show in the picture below.
   c. Next, obtain 3 wires (any color) from the lab equipment area (TA will help).
   d. Wire them to the microphone using the breadboard as shown in picture below.
   e. You'll need to "strip" the wires at the ends to make contact (TA will help).
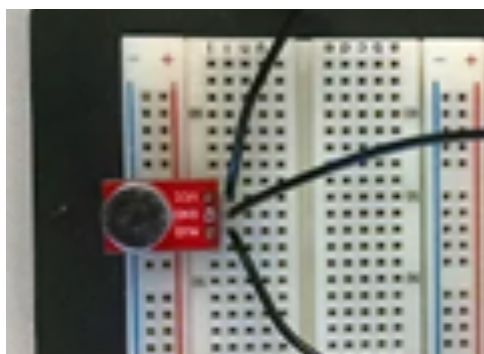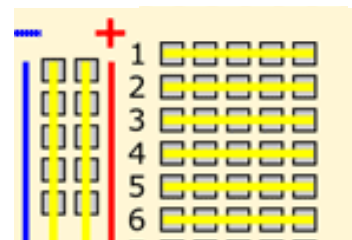
4. Next, we'll power our microphone up by using a power supply as follows:

a. The power supply is on top of your lab station, it is shown in the picture below:
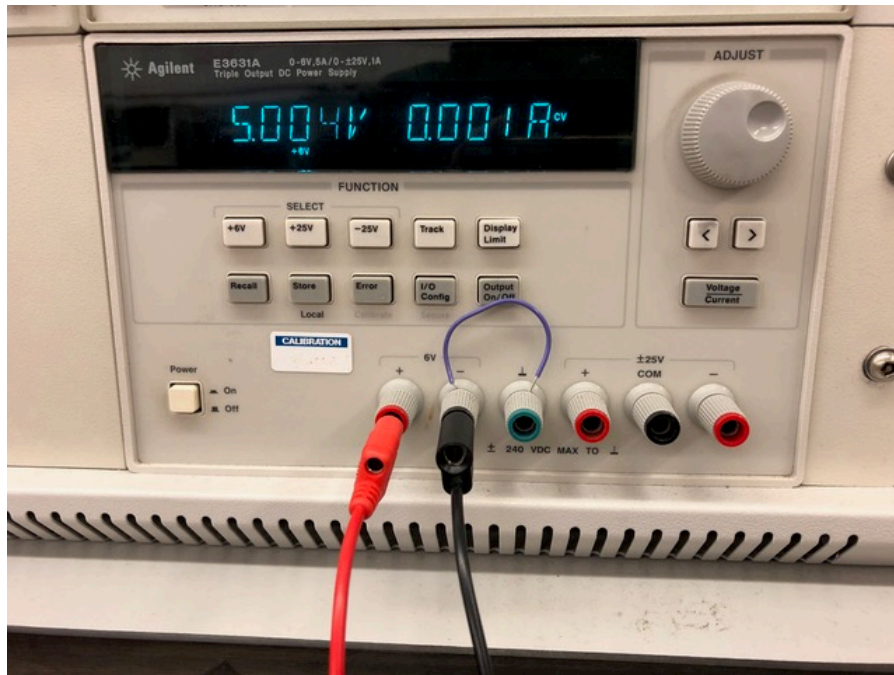
b. Attach a red and black wire (banana to grabber wire) as shown in the picture.
c. Attach one wire between the negative (-) terminal and the GND (⊥)terminal
d. Also attach the two wires coming from the power supply to your microphone as shown in this picture (these are the grabber ends of the wire):
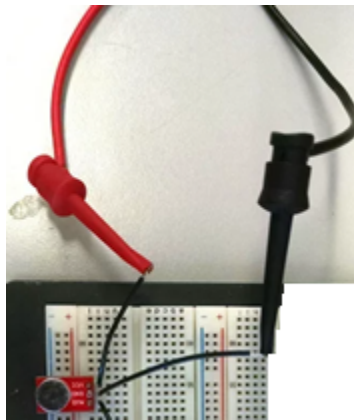
e. Next, turn on the power supply and set it to deliver 5V as follows:
    i. Press the "+6V" button, use the dial change the voltage to read: 5V
    ii. Press the "Output On/Off" button to turn the 5V on.

5. We want to see the output of the microphone, so we'll attach the output of the microphone to our oscilloscope by doing the following:
   a. Obtain a "BNC to grabber" cable.

b. Attach the grabber ends of the cable to your microphone's output and ground as shown in this picture:
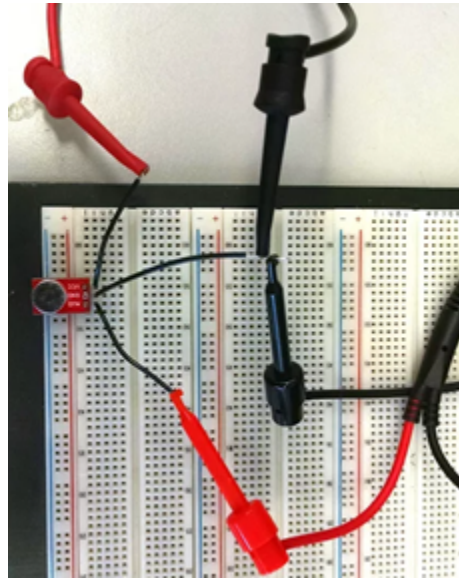
c. Ensure that both the red and black grabber ends of the BNC cable are connected, where the red is attached to the output, and the black is connected to the ground. This will allow the oscilloscope to measure the voltage difference.

d. Attach the other end of the cable (the BNC end) to the oscilloscope's channel 1.
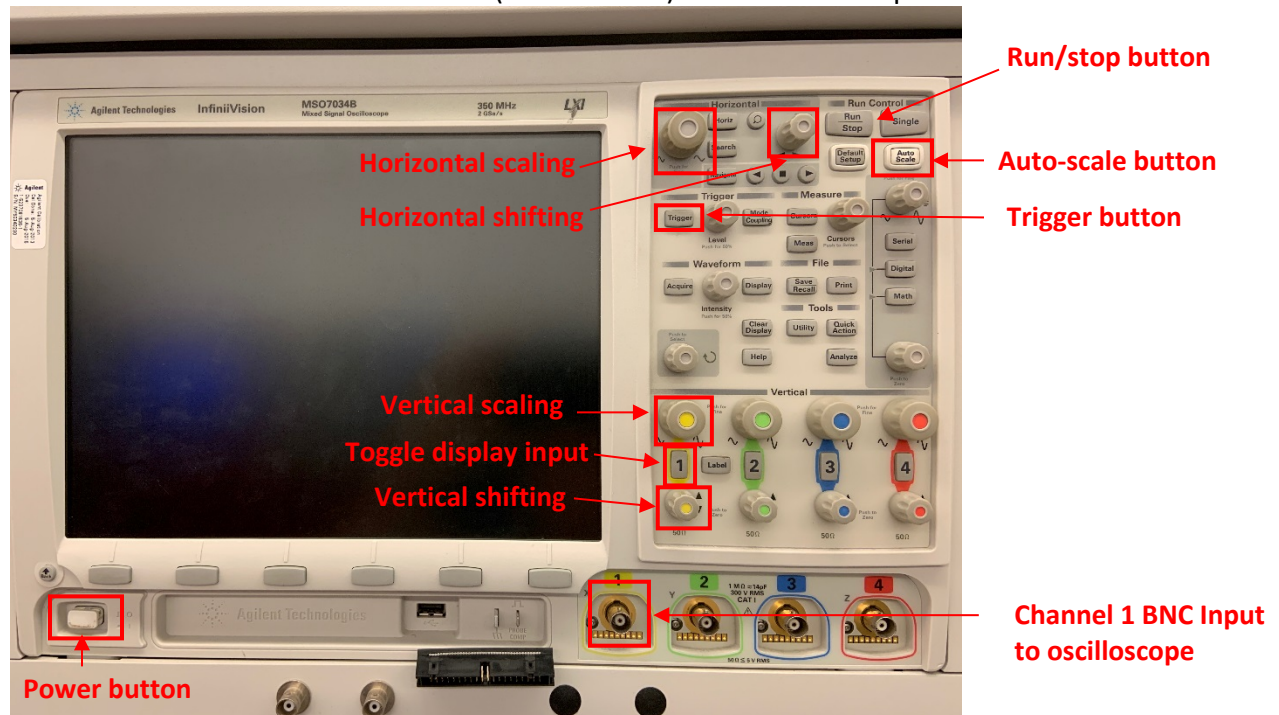
e. The oscilloscope is shown above, look for the "yellow" BNC connector on it labeled "1" – attach your cable there.

f. Turn on the oscilloscope with the power button in lower left corner.

g. Now the fun part…

        i.   SING a pure tone…say "AHHHHHHHHH" into the microphone.

      ii.   Press the "auto-scale" button the oscilloscope.

h.   How does the scope work?

        i.   There is a horizontal knob that you can use to stretch out the x-axis.

      ii.   There is a vertical knob that you can use to stretch out the y-axis.

     iii.   But you are viewing your voice converted from pressure vs. time to voltage vs. time.

     iv.   Play around a bit and try different "pitches" to see the results.

      v.   Trying talking into it!

     vi.   Take a screenshot or two of interesting figures with Excel (explained below); you'll need to turn these in with your lab.

          1.   Open Excel on the lab computer and go to the Add-ins tab.

          2.   Click , and select the Address and Identified Instruments that show up, before clicking OK.

          3.   To capture the oscilloscope screen, click and uncheck the greyscale box before clicking OK.

          4.   The screenshot should now be in your Excel workbook.

    vii.   The trigger button can be used to stabilize and focus the oscilloscope output, with many different types of triggers that are suited to various inputs.

   viii.   The run/stop button can be used to (un) freeze the screen, allowing for accurate screen captures.

     ix.   The cursors button and knob can be used to turn on and move the cursors vertically and horizontally across the screen to find exact values in the waveform.

Lab – Section 2: Measuring pure tones with an oscilloscope

- Since it's hard to keep saying "AHHH" during this whole experiment, we'll use a more dependable audio source, a "function generator".
- A function generator can generate signals whose voltage varies with time, much like the output of our microphone.
- We'll have the function generator generate a few sine waves for us, with an exact amplitude and frequency, so we can measure them with the oscilloscope.

1. Look for the function generator on the top shelf of your lab workstation:
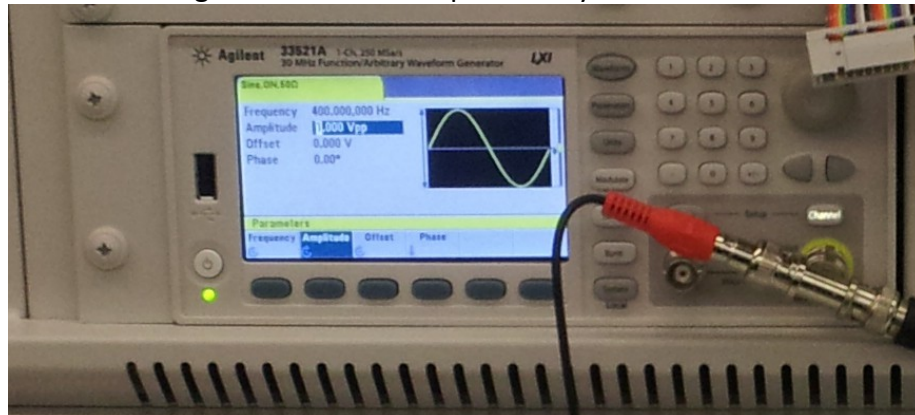
2. Obtain a "BNC to BNC" connector wire from your lab bin, connect the yellow output from the function generator to CH1 on your oscilloscope. You can disconnect your microphone's output from the scope for this experiment.
3. Configure the function generator as follows:
   a. On the function generator, press the "**Channel**" button right above the output:
      i. A menu will be displayed along the bottom of the screen; you can select the options using the blue buttons beneath each one.
   b. Press the "**Waveforms**" button. You can set the type of wave (square, sine, etc.).
   c. Options will be displayed on the bottom of the screen: select "***sine***".
   d. Now you can set the frequency, amplitude, etc. for the sine wave:
      i. Press the button below **frequency**.
      ii. The field with the value of the frequency should now be highlighted.
      iii. Enter 400.
      iv. After you enter a #, a list of units appears at the bottom of the screen.
      v. Select Hz.
   e. Next, adjust the amplitude and offset the same way that you adjusted frequency:
      i. Set the amplitude to 1 VPP (1 Volt – peak to peak).
      ii. Set offset to 0.5V.
   f. Turn the output on by:
      i. Press the channel button: in the menu that appears at the bottom of the screen, select the leftmost button to switch the **output on**.

4. Measure this on the oscilloscope
   a. On the scope, press "autoscale" and you should see your 400 Hz 1VPP (1V peak-to-peak) sine wave.
   b. Measure the period of the wave using the vertical lines on the screen
      i. Convert it to frequency, is it 400 Hz?
   c. Measure the voltage range of the wave using the horizontal lines on the screen.
      i. Is it indeed 1 Volt Peak-to-peak?
   d. Once you are sure its correct, take a screenshot of the waveform to save to Excel.
   e. Adjust the function generator to 1kHz.
      i. Scale your scope and repeat previous steps to measure the period (verifying that it is now 1kHz) and take a picture of the waveform using Excel.

Lab – Section 3: Sampling a pure tone using the Arduino

An Arduino is a microcontroller. This basically means it a little computer that has lots of input and output and you can control by writing a simple program (we call them sketches on the Arduino platform)

The goal of this section is to:
- Attach the output of the function generator to the analog input of the Arduino
- Modify a small Arduino sketch to sample the input from the function generator
- Output the samples taken over time to the Arduino's terminal output window
- Capture the output and save it for a future lab!

In this section of the lab, we will use the following code to sample the output of the function generator with the Arduino:

```
int incomingAudio[800];
int startTime;
int run = 1;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600) ;
 }


void loop() {
  // put your main code here, to run repeatedly:
    if(run){
      for(int i = 0; i < 800; i++) {
        incomingAudio[i] = analogRead(A0); // read input from A0
        delayMicroseconds(1000);
        // change line above to change sampling rates
      }
      for(int i = 0; i < 800; i++) {
        Serial.println(incomingAudio[i]);
      }
     run = 0;
     }
  }
```

An Arduino program is called a sketch and is based on C/C++. For the specifics of Arduino programming, you can reference the Arduino Language Reference documentation online. The language is split up into structures, values, and functions. Use this resource to understand the code snippet shown above, both to familiarize yourself with the language and to understand

ESE 150 – Lab 01: Sampling and Quantizing Audio Signals

the purpose of the code. Note that the Arduino sketch is comprised of these two functions, setup() and loop(). Effectively, the Arduino sketch behaves like this:

```
 main() {

   setup();

   while(true) { loop(); }

}
```

The Arduino sketches are making it easier by just letting you specify the contents of these two functions. The setup() is the one-time code. Here, we use this section to set the speed at which the Arduino communicates to the host computer over the serial port. The loop() code is the body that is invoked repeatedly, and is usually the main heart of the program. We don't actually want the body code to infinitely repeat, so the run variable is used to ensure the program only executes our logic once. Each time you reset the Arduino (or download new code), it starts over and runs the implicit main routine, re-running setup() then repeatedly executing the contents of loop().

1. Set the function generator to a sine wave at 300 Hz, 2V peak-to-peak, and 1V offset.
2. Attach the output of the function generator to the A0 pin on the Arduino as follows.
    a. Attach a BNC to grabber cable to the output of the function generator.
    b. Plug two wires into the Arduino pins: one in ground (GND) and the other in A0.
    c. Attach the black grabber cable to the ground pin and the red grabber cable to A0.
    d. Now your Arduino can read the signal from the function generator!
3. Connect your Arduino into the USB on your computer, specifically to the USB ports on the computer towers (**not** the monitors).
4. Upload the provided sketch to the Arduino
    a. Open the Arduino IDE (open the start menu are search 'Arduino').
    b. Adjust the settings:
        i. Select Tool -> Serial -> COM## (the number will differ depending on your machine, but it will not be COM1)
        ii. Select Tool -> Board -> Arduino Uno
    c. Paste the provided code into the IDE.

5. Click the upload button to upload the code to the Arduino. You will see a message "Done Uploading" once the upload has finished. Once you see this, click to open the Serial Monitor.
    a. You may get an error: "Low memory available, stability problems may occur" – that's expected and does not indicate a problem.
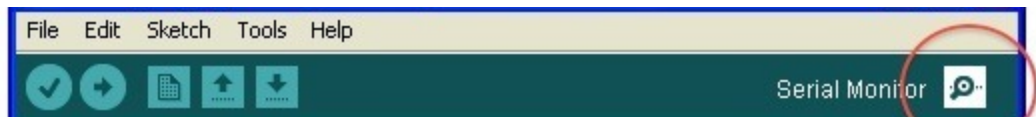
6. After a few seconds, a sequence of numbers will be printed into the Serial Monitor. Wait until it is finished printing numbers, then select all, and copy and paste into Excel to save the output.
   a. Make sure you're seeing numbers that range between 0 and 1023 (not all of the, and maybe not all the way to 1023, but many different values in that range); if you're seeing all 0's or only a couple of different values, things probably aren't connected correctly.
   b. We will use this output for the next lab, so make sure to save the file somewhere (like your eniac home directory and on the shared google drive noted below) you will be able to recover it!
   c. You should also save your Arduino sketch for use in the lab report.
7. Modify the sketch to sample at 500Hz. You will need to change the delayMicroseconds function. What is the relationship between sampling frequency and delay (Prelab 1)?
8. Repeat steps 5-6 with the modified code and save the sketch and output.
   a. Close the Serial Monitor between runs to clear it.
   b. To rerun the sampling code, press the reset button on your Arduino.
   c. The Arduino's analogRead() function has a sampling frequency limit of ~ 8kHz.
9. Modify the sketch to sample at 5000Hz, run, and save the sketch and output.
10. Before leaving lab, show your data sets to a TA and answer a few questions. This is the Lab Exit Check-off.
11. Share your data: *before leaving lab*, make sure both partners have access to the data collected and sketches saved. One way to do this is to setup a shared Google Drive.
12. Clean up your lab station before you leave.

<u>HOW TO TURN IN ANSWER TO THE LAB:</u>
- Answer the prelab questions, assemble the data requested, and answer the questions in the lab.
- Upload a word document or PDF containing your informal lab report including
  - Partner's name
  - Prelab answers
  - Oscilloscope screen shots (1.5.h.vi, 2.4.d, 2.4.e)
  - Arduino sketches (section 3)
- Separately upload your data set samples (3 data sets – 3.6—9)
  - Don't forget to save your data! (You don't need to turn it in yet, but we will use it for the next lab)
- Each lab writeup is individual.
- You can see the grading rubric we are using for the lab on Canvas. Review that to make sure there will be no surprises when your lab is graded.
- Due by Friday 5pm