lab1

January 30, 2017

1 BioE 101 Lab 1

1.1 Electrical Test Equipment and Arduino Coding

1.2 Objectives:

- Learn how to use an oscilloscope and function generator. - Learn to use breadboard to build and test simple circuits like resistive voltage divider, and RLC filter. - Learn how to use an Arduino and Python to acquire signals.

1.3 Lab Report

Answer the questions in the lab procedure with your partner in a separate document.

1.4 1. Setup

- We should generally have 2 people per lab station.
- Your team should have the following items on your station: oscilloscope (with one test probe), function generator, power supply, Arduino Uno (with grey serial cable), multi-meter, breadboard, jumperwire kits, pliers, BNC cable and adaptors.
- You can get the electronic parts such as the resistors, capacitors, diodes, etc. from the supply bins near the cable rack.
- Please keep all the lab equipment in lab only.
- Before you leave the lab, make sure the power for all the equipment is switched off.

1.5 2. Introducing the Oscilloscope and Functional Generator

- The oscilloscope allows us to view electrical signals (voltages or currents). Sometimes, it is used in conjunction with a function generator to observe the effect that a test signal has at one or more point(s) in our circuit. This is very useful in determining, for example, the input/output characteristics (i.e. the transfer function) of an amplifier. Despite their age, these scopes are very powerful tools. These oscilloscopes are digital, meaning they convert the analog signals at their inputs into numeric waveforms with the sampling rate and scaling parameters that you specify. This offers some very important advantages over analog oscilloscopes of the old days.

- The function generator allows us to apply test signals to our circuit such as sine waves, square waves, pulses, triangle waves, and even waveforms which we define ourselves (referred to as “arbitrary waveforms”). We can control frequency and amplitude, and we can even sweep through many frequencies/amplitudes in order to test our device with a range of conditions.

- Take a good look at your oscilloscope and answer the following questions:

Question 2.1: What is the bandwidth of this oscilloscope? What does that mean for the signals you will be able to observe with it? What would the sampling rate of the ADC inside the scope have to be in order to achieve this bandwidth?

Question 2.2: What is the input impedance of this scope? Why would that matter?

- Use the BNC cable to directly connect the output of the function generator to channel 1 of the oscilloscope. Make sure channel 1 is on and the scope is acquiring signals.

- Set the function generator to a 100 Hz, 1V sine wave with 0V offset. Turn on the output by pressing the "disable" button so the light goes off.

- Adjust the scope time and voltage scales to comfortably see several periods of the waveform. To do this, you will need to familiarize yourself with 3 controls:

- Volts/Div: This sets the Y scale of each channel (channel 1 and 2 are set independently). The current scale is in the upper left corner of the screen. A value of 500mV, for example, means that each horizontal line in the grid corresponds to 500mV.

- Time/Div: This sets the X scale of both channels (they must be the same). The current value is in the upper right corner of the screen. Note that as you "zoom out" on the waveform, the sampling rate drops.

- Voltage Position: Each channel has an additional knob that lets you move the waveform up and down on the screen. This is useful when you want to display multiple waveforms.

- Trigger Source, Mode, and Level: The trigger is useful for synchronizing your display to the signal so it doesn't scroll across your screen. Edge triggering (also called "normal" triggering) causes the display to refresh only when the signal crosses the threshold amplitude. You can trigger on either channel 1 or 2, and you can adjust the triggering threshold. You can also specify whether you want the signal to trigger on a low-high transition or a high-low transition. If "auto" is selected, the trigger values are ignored and the screen constantly refreshes. Auto triggering can be useful initially for setting up scope parameters. Try adjusting the trigger parameters and see what happens to your signal.

Question 2.3: What is (roughly) the largest and smallest voltage that you could observe with this scope (hint: try changing the volts/div knob to see the limits)? What is (roughly) the fastest and slowest frequency that you could observe with this scope?

Question 2.4: What happens when you set the trigger to a voltage that your sine wave never reaches?

- Play with the function generator and see how the signal changes. Try changing the frequency, amplitude, offset, and waveform type (sine, square, triangle). When you're done, set it back to 100 Hz, 1V sine.

- Cursors are a handy way to measure signal voltages and timing on the oscilloscope. Pressing the "cursors" button will turn them on. Make sure the "Source" for the cursors is set to channel 1. Use the buttons below the screen to select which cursor you want to move and the knob near the cursor button to move the cursor. The positions and distances appear at the bottom of the screen. Using the cursors, verify that the amplitude is 1V and the period is 1/100 = 10ms.

Question 2.5: Is 1V the peak-to-peak amplitude, the RMS amplitude, or the peak amplitude?

- You can also have the scope automatically measure voltage and timing parameters by pressing the Voltage/Time buttons under the "Measure" label. You can have it display 3 different quantities at a time. Try displaying the average voltage value, RMS voltage value, and frequency of your waveform. Make sure "Source" is set to channel 1.

- Try changing the function generator to a square wave, then to a sawtooth wave.

Question 2.6: How do the 3 measurement quantities on your scope (average value, RMS value, and frequency) change for a sine, square, and triangle wave? Explain what you see.

1.6 3. Introducing the Breadboard

- The breadboard is a powerful tool to test and debug the circuit design since no soldering is required. A typical breadboard consists of a perforated block of plastic with metal spring clips under the perforations. The clips are interconnected following a certain pattern underneath the plastic block (see the example below).The holes connected by the imaginary lines means that they are shorted. Electronic parts can be connected by plugging into the interconnected holes or by jumper wires.

- Take a look at the interconnection pattern above, the layout of a typical breadboard is made up from two parts: terminal strips and bus strips. Terminal strips are the main areas to hold the most electronic components. The holes are usually grouped into five in a column. Bus strips, usually run on the long side and are typically used for providing power to the circuits being built within the terminal strips area.

Question 3.1: How can you test your own breadboard to determine its interconnection pattern? Draw a simple diagram to explain how the holes in your breadboard are interconnected.

- Jumper wires with 22 AWG (American wire gauge) can be used to connect two electronic parts. The wire ends should be stripped 3/16" to 5/16". Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of shortcircuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards. One tip for keeping your circuit neat and readable is to use different color wires for different signal paths.

- The breadboard is reusable after each time by removing all the parts and jumper wires from the board.

1.7 4. Build Circuits and Measure Circuit Parameters

In this lab, we will build two circuits – (1) resistive voltage divider, and (2) a simple RC filter. Keep your circuits on the breadboard until the end of the lab, since we need the circuits throughout the rest of the experiment.

- Voltage divider: it is a simple linear circuit that produces an output voltage (V_{out}) that is a fraction of its input voltage (V_{in}). A simple example of a voltage divider consists of two resistors in series. It is commonly used to create a reference voltage, and may also be used as a signal attenuator.

- Build the circuit on the breadboard using two resistors. Choose $R_1 = 3.3\text{k}\Omega$, $R_2 = 10\text{k}\Omega$. Use a multi-meter to make sure that the resistors are of the correct values. Note that resistor values are printed on the outside as a series of colored bands. Take a look at the chart below to verify that you have a 3.3kΩ and 10kΩ resistor.

- One hint to manage your power supply and ground system is to use the strip bus: make sure you connect the strip bus to the banana plugs which connect to the external power supply.

Question 4.1 Suppose you accidentally picked up a 33kΩ and a 100Ω resistor. What would the color bands look like instead?

- Feed Vin with 5V from the power supply. Turn on the power supply, press and hold the "6V" button to limit the output range of the power supply from 0V to 6V. Then tune the dial to set the output voltage to 5V. You can tell the actual output voltage by reading the number on the display panel. Hook the output of the power supply to the breadboard. Test the voltage output using the multi-meter.

Question 4.2: What is the output voltage? Explain how this circuit can be worked as a voltage divider? Can I use small resistors (like 1Ω) for this application and why?

- RC filter: you can turn the resistive voltage divider into a filter by simply replacing R2 by a capacitor. Choose R to be 3.3kΩ and C to be 0.01μF. Build the RC circuit on the breadboard. Analyze this circuit and answer the following questions.

Question 4.3: What type of filter is it (high-pass, low-pass, band-pass, band-stop)? What is the cut-off frequency of this filter (here we use the −3dB definition)? Sketch the Bode plot of your filter.

- Connect the input Vin of this filter to the function generator and hook the output to the oscilloscope. Tune the function generator to output a sine wave of frequency 9kHz, 5 Vpeak, 50 % duty cycle, 0V offset.

Question 4.4: What does the output signal look like? How does it compare to the input signal? Is this what you would expect? How would you experimentally get the Bode plot?

1.8 5. Introducing the Arduino

The Arduino is an open-source physical computing platform made up of two parts: a simple microcontroller board and a development environment for writing software to the board. Arduinos can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer. The programming language for the free, open-source IDE is an implementation of Wiring, with libraries written in C++.

There are many very attractive features of arduinos, including low board cost, easy-to-use programming environment, and virtually limitless supply of online resources and tutorials due to its immense popularity and fan-base.

Hardware - Arduino Uno Board: We will be using the Arduino Uno Rev 3 boards, with a schematic and overview of the various pins shown below:

Note the difference between digital and analog pins. Digital pins use 5V (HIGH or 1) and GND (LOW or 0) as it's only signals (with some threshold before 5V and after GND to be considered HIGH or LOW). Analog pins receive/output a value between 0V and 5V encoded between 0 and 1023 (if the ADC is 10 bit).

Here you will learn how to upload code to Arduino. We have given you a .ino file that programs the Arduino to sample at a given sampling rate.

1.8.1 Connecting the Arduino:

- Using a standard USB cable (A plug to B plug), connect the Arduino to computer.
- Launch the Arduino application
- Under the 'Tools>Board' menu select the type of board you're using (Arduino Uno), and under 'Tools>Serial Port', select the serial (USB port) that your arduino is connected to (there should be be only one option).

Follow these steps to test the Oscilloscope and Function Generator:

- Start the Arduino IDE and open the ** 'send_serial.ino' ** file in your lab1 folder
- Click the arrow button next to the check button to upload your code to your board. The text near the bottom of the IDE should say your code was uploaded successfully. During this process you should see the TX/RX leds blink rapidly on your board.

1.8.2 Interfacing Arduino with Lab Equipment

Instead of using the oscilloscope to see the voltage waveform, you can now view the waveform by going to 'Tools>Serial Plotter' and view a real time plot of the function generator wave. You can only view up to a few hundred hertz before the signal aliases. This is okay since biomedical signals are usually at a very low frequency.

1.8.3 Design Problem

You're building a circuit for your biomedical startup. However, you keep finding that there's this huge interference when looking at the voltage outputs. It's the power line interference at 60 Hz! Things to consider: the biomedical signal you want to acquire does not have any frequency components past 30 Hz.

Let's get rid of that pesky 60 Hz interference from the power lines.

** Question 5.1:** Build a first order RC lowpass Filter that would take care of the problem. What will your RC value and cut-off frequency be? (Hint: don't forget that annoying 2π factor)? Is it okay that we're filtering every frequency past that cut-off frequency?

** Question 5.2:** Let's test that the circuit actually filters out the 60 Hz. Let our desired signal be 5 Hz. Send that through the arduino and view the waveform. Now send 60 Hz through the arduino and view it. What is the difference in magnitude of those frequencies, assuming the input amplitude was the same?

** Question 5.3: ** Find the cut-off frequency of your real circuit by tuning the frequency until it is roughly 0.7 times the amplitude of the input (where did that number come from?) Compare that to the one you calculated. Explain any differences.

** (Bonus) Question 5.4: ** Let's say that you want to suppress the 60 Hz interference even further. What are some options you can try? What are the trade-offs /risks to those options?