Lab 1: Elements of Electronic Systems

Instructor: Xiaoguang "Leo" Liu
lxgliu@ucdavis.edu

The main objective of this lab is to understand the basic components of a typical electronic system. To this end, we will build a simple electronic system shown in Fig. 1. In this system, we will use a micro-controller and a digital-to-analog converter (DAC) to generate a triangle wave. This triangle wave signal will be low pass filtered and digitized by an analog-to-digital converter (ADC) whose output will be sent to a computer for analysis and display.

Although this system may seem rather useless, it does serve the purpose of introducing the basic concepts and skills in building an embedded system that needs to interface with the analog world. In addition, some of the components in this system will become useful in later labs.

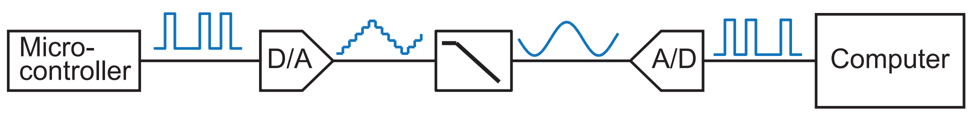

Figure 1: Lab 1 electronic system.

1 Objectives

1. Understand the functionality and characteristics of linear and switching voltage regulators;
2. Learn how to use simple micro-controllers, such as the Arduino Uno and the Teensy 3.1;
3. Learn the functionality and use of DAC and ADC;
4. Learn how to use the serial programming interface to control electronic components;
5. Learn the basic skills of designing and laying out a printed circuit board (PCB);
6. Learn the basic skills PCB assembly involving surface mount (SMD) components.

2 Prelab

2.1 Micro-controller

Micro-controllers are computing systems integrated within a single package. Compared with general purpose computers, micro-controllers find widespread use in applications where low-power consumption, small physical size, low cost, and predictable response time are required.

Traditionally, programming a micro-controller was an art that only well-trained electrical engineers can perform. This all changed with the invention of the Arduino micro-controller platform in 2005 by a group of Italian educators who were frustrated with how difficult it was to learn and use microcontrollers. They decided to develop a hardware and software platform so easy to use that even artists were able to use them. In addition, the Arduino platform is completely open source. Because of this, there has been a large number of library and add-on boards (called "Shields") developed and made available by the community, making it possible to build complex systems without having to reinvent the wheels. The huge popularity of the Arduino platform has motivated the development of numerous similar platfomrs with a wide spectrum of capabilities. These are the reasons that we choose to use Arduino compatible micro-controllers in this class[1]. We hope that even students who are not interested in embedded systems will be able to master how to use them in a relatively short of time.

Although the focus of the EEC 134 class is on high frequency systems, you will find that micro-controllers indispensable whenever you need to control a component electronically.

- To get a quick introduction to the Arduino platform, watch Episodes 1–10 of Jeremy Blums "Tutorial Series for Arduino ". It is advised that you, or your group collectively, go through all of the videos in this list. `https://www.youtube.com/playlist?list=PLA567CE235D39FA84`

2.2 Voltage regulator

In the late 1880s, a heated battle over the best mechanism to transport electricity over long distance broke out between proponents of direct current (primarily Thomas Edison) and alternating current (primarily George Westinghouse and several European companies). History eventually settled on ac current as the preferred method for long distance distribution because of its ability to be easily transformed into high voltages to reduce resistive loss along the wires. So today we all have ac outlets at home and in the lab. However, most if not all the circuits we have studied in our curriculum are powered from dc supplies. Have you ever wondered how dc voltages are generated from an ac supply? The following videos may be instructive. Make sure you watch them carefully.

[1] If you have experience using a microcontroller different than the Arduinos, you should feel free to use that platform to implement this lab; obviously you may want to make sure that you have a consensus within your group.

- How to build an AC-DC power supply:
  `https://www.youtube.com/watch?v=cyhzpFqXwdA`
- Linear voltage regulator:
  `https://www.youtube.com/watch?v=GSzVs7_aW-Y`
- Adjustable linear voltage regulator:
  `https://www.youtube.com/watch?v=IjJWWGPjc-w`
- Switch-mode voltage regulator:
  `https://www.youtube.com/watch?v=CEhBN5_f05o`

2.3 DAC and ADC

DAC and ADC are the interface between the analog and the digital world. A DAC takes digital input ("1"s and "0"s) and convert them into an analog signal. An ADC does the just the inverse of that, converting an analog signal into a digital one. A simple DAC can be built with only resistors, such as the R-2R ladder shown in Fig. 2-a. A simple ADC is not much more complex using a linear resistor ladder integrated with comparator and decoders (Fig. 2-b)[2]

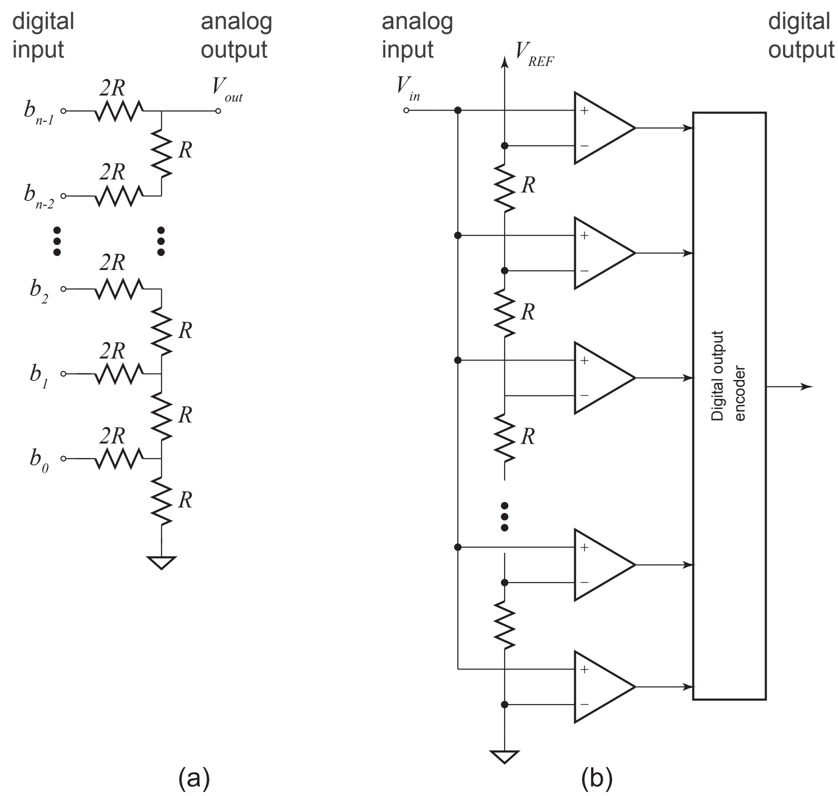

Figure 2: (a) R-2R ladder DAC. (b) ADC using a linear string of resistors.

To learn more about DACs and ADCs, read the following documents.

[2]A variety of other architectures exist for both DACs and ADCs with respective advantages and disadvantages.

- Bill McCulley, "Bridging the divide: Part I - DAC Introduction[3]," National Semicondcutor, 2010
- Chris Pearson, "High Speed, Digital to Analog Converters Basics," Texas Instruments, 2012.
- Mike Kester, "DAC Interface Fundamentals," Analog Devices, 2008.
- Chris Pearson, "High-Speed, Analog-to-Digital Converters Basics," Texas Instruments, 2011.

2.4 Serial interface

For micro-controllers that don't have built-in DACs and/or ADCs, an external DAC/ADC IC needs to be used when conversion between analog and digital is required. Even micr-controllers with on-chip DAC/ADC may run into situations where more than conversions channels are needed. When using an external device, a digital interface for transmitting and receiving data is needed between the micro-controller and the device.

Digital interface generally comes in two flavors: *parallel* and *serial*. In a parallel interface, each physical pin corresponds to one digital bit. For example, an 8-bit word would need 8 physical pins for transmission/reception. In a serial interface, the digital bits are transmitted/received sequentially in time and theoretically only two pins are needed (one for signal and one for ground). Depending on whether a clock is needed to time the transmission/reception of digital bits, serial interface can be further categorized into *synchronous* and *asynchronous*. For synchronous transmission, an additional pin for the clock signal would be needed.

Obviously, at the same physical bit rate[4], parallel interfaces is much faster than a serial one. But the serial interface requires far less pins and therefore a smaller chip footprint and lower PCB routing complexity, both of which translates to lower cost!. Therefore, for medium to high rate (kp/s to a few Gp/s) data transmission, serial links are more popular.

In this class, we use synchronous serial interface, in particular the Serial Peripheral Interface (SPI) protocol, as the interface between micro-controllers and external devices, such as DACs, ADCs, and some other electronically controllable RF components.

To learn more about the SPI interface, read the following documents:

- Mike Grusin, "Serial Peripheral Interface (SPI)," Sparkfun. `https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi`
- F. Leens, "Introduction to IC and SPI protocols," Byte Paradigm, 2009. `http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/`

[3] Following the link will lead you to EEC134's repository on Github. If you want the actual PDF file, you can clone the entire repository to your harddrive. The file is under the foler /labs/lab1/references/.

[4] how often the voltage changes from logic 1 to 0 (or the other way) on a pin.

2.5 Printed Circuit Board Design

A major goal of this lab is to help you get familiar with printed circuit board (PCB) design techniques. There are numerous PCB design software packages available on the market. In this quarter, you will start with a simple tool called KiCad. Once you are familiar with the basic concepts, it is easier to transition to a more sophisticated tool, such as the Cadence Allegro which is made available to us by a donation from Cadence.

KiCad is an open-source electronic design software with very good PCB layout capabilities. KiCAD is becoming more popular in the hobbyist electronics community because it doesn't have any restrictions on the number of layers or boards sizes, nor does it lock you down to a particular PCB manufacturer.

To learn how to use KiCad and design PCBs in general, please go through the following materials.

- Video tutorials from Contextual Electronics: `https://www.youtube.com/playlist?list=PLy2022BX6Esr6yxwDzhqYZyuuenJE2s5B`
- Video tutorials form Yoonseo Kang: `http://vimeo.com/user9565582`
- A simple KiCad example:

  `http://exploreembedded.com/wiki/A_simple_example_for_beginners:_LED_Breakout`

Pre-lab Assignment 1.1
Due: Sep. 25th, 2015

Please answer the following questions:

a) What are the advantages and disadvantages of using switch mode voltage regulator vs a linear voltage regulator?

b) For the circuit of Fig. 3, with sing an input voltage of 9V and R1=510 Ω, what's the value of R2 such that the output voltage is 5 V? What is the efficiency of the regulator in this case? "LM317" refers to the Texas Instruments LM317 voltage regulator IC. You should be able to find its datasheet online.

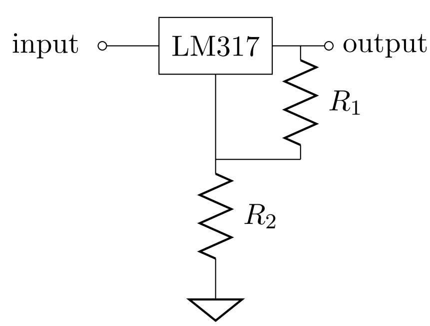

Figure 3: LM317 linear regulator circuit.

c) According to the datasheet you found above, what is the typical drop out voltage for the TI LM317? If an input voltage of 12 V is used, what range of output voltage can be considered regulated?

d) What is the maximum efficiency of the TI LM2694 switch mode voltage regulator for an output voltage of 5 V? Under what conditions is this efficiency achieved?

e) What do *LSB* and *MSB* mean in a DAC? For a 12-bit DAC with an output reference voltage V_{ref} of 2 V, how much voltage does an LSB correspond to?

Pre-lab Assignment 1.2
Due: Oct. 2th, 2015

Please answer the following questions:

a) What does *SFDR* mean for a DAC? What is the typical SFDR of the Analog Devices (ADI) AD9788 DAC?

b) What is the *image* signal for a DAC output? If a DAC is operating at a clock rate of 200 Msps and the output fundamental signal is a 50 MHz sine wave, what are the frequencies of the first three images above the fundamental?

c) What does *SNR* mean for a DAC? What is the typical SNR for the Linear Technology (Linear) LTC2641 DAC?

d) What does *THD* mean for an ADC? What is the typical THD for the TI ADS5400 ADC ?

e) What does *SINAD* mean for an ADC? What is the typical SINAD for the Linear LTM9008-14 ADC?

f) What does *ENOB* mean for an ADC? How is it calculated? What is the ENOB for the Maxim Integrated (Maxim) MAX11270 ADC?

Pre-lab Assignment 1.3
Due: Oct. 9th, 2015

Please answer the following questions:

a) What is the highest speed 8-bit ADC you can find? What is its power consumption? What would be the power consumption of an 8-bit ADC with half of this speed?

b) What is the frequency domain representation of the following triangle wave signal?

$$x(t) = \sum_{k=-\infty}^{\infty} \left[\left(\frac{2t}{T} - \frac{kT}{2} \right) \Pi(t - kT) + \left(1 - \frac{2t}{T} \right) \Pi \left(t - \frac{(2k+1)T}{2} \right) \right],$$

where $\Pi(t)$ is the rectangular pulse function

$$\Pi(t) = \begin{cases} 1 & 0 \leq t \leq 1; \\ 0 & \text{elsewhere.} \end{cases}$$

If we pass the signal through an ideal low-pass filter that keeps only the first three harmonics (including the fundamental, that is, the fundamental, the 2nd harmonic, and the 3rd harmonic), what would the filter output signal look like in the time domain?

c) Design an analog low-pass filter that meets the following specifications:

(a) In-band gain: 10 dB;

(b) 3-dB cut-off frequency: 20 kHz;

(c) Attenuation at 100 kHz: 30 dB.

You may find online filter design tools, such as the TI WEBENCH Filter Designer and the ADI Filter Wizard, to be useful. Verify the performance of your filter by simulation. You may use a SPICE simulator, such as LTSpice, or a high frequency circuit simulator such as Agilent/Keysight Analog Design Systems (ADS). Both software are available on lab computers.

3 Equipment & Supplies

- breadboard
- jumper wires
- Arduino UNO development board
- Teensy 3.1 development board (optional)
- MCP4921 digital-to-analog converter (DAC)
- misc resistors
- misc capacitors
- 8x AA rechargeable batteries
- battery pack

4 Procedures

4.1 Power supply/voltage regulator

We will use batteries to power up our system. Batteries provide the cleanest (i.e. very little noise) type of supply but with one major disadvantage, their voltage keeps dropping due to increased internal resistance under discharge. A voltage regulator is therefore often used to provide a relatively stable dc voltage supply.

In our design, a LM317 adjustable regulator IC will be used to provide the supply voltages for several parts in the system. LM317 is a linear voltage regulator, which can provide a very clean output voltage and is quite simple to use. A linear regulator can only provide regulated output voltage lower than the primary voltage source. Another disadvantage of a linear regulator is its low efficiency when the difference between the input and output voltages is large. Because the same amount of current flows through the regulator and the load, the power being dissipated is roughly.

$$P_d = I_{load} * (V_{in} - V_{out}) \tag{1}$$

When the load current is high, this power dissipation can be significant. It is therefore important to provide good heat sinking to the regulator IC. In fact, many regulator ICs, such as the LM317, have large exposed metal pad with low thermal resistance to facilitate mounting a heat sink. However, it is very important to note that the metal pad on some regulator ICs, such as the LM317, is connected electrically to the output of the IC. If the heat sink may potentially come in contact with any other part of the circuit (including the enclosure, which often is tied to ground), proper isolation is needed between the IC and the heat sink.

1. Using Fig. 4 as a reference, build the regulator circuit on the breadboard.

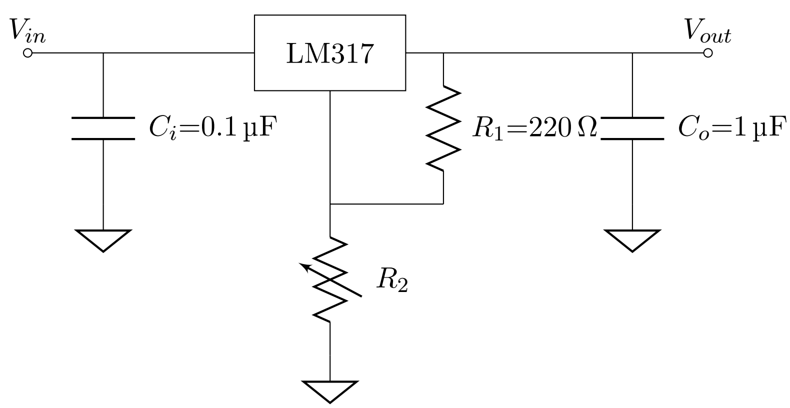

Figure 4: Schematic of the voltage regulator circuit using LM317.

2. Use the batteries as your input power. Adjust R_2 until the output voltage is 5 V. For the LM317, you will need 8 AA batteries in total (2 packs). You should use the multimeter to measure the output voltage.

3. Use the oscilloscope to measure the regulator output vs. time. Is the output stable?

4. Now use the bench power supply as the input to the regulator, adjust R_2 until the output voltage is 5 V. Observe the output voltage on the oscilloscope. Compare with the case when batteries were used as the input power. Which one gives a more stable output? What could have caused the difference?

5. Line regulation:

   (a) Use the bench power supply as the input; set it at 7 V;

   (b) Adjust R2 until the output is 5 V;

   (c) Adjust the input voltage from 7 V to 5 V at 0.25 V intervals, and then from 5 V to 2 V at 0.5 V intervals, record the output voltage;

   (d) Plot your results. In what input voltage range does the LM317 provide good line regulation?

4.2 Function generator

There are numerous ways to build a function generator. You can use a 555 timer, a ring of inverters, dedicated oscillator ICs, or the more recent invention of direct digital synthesis (DDS) circuits. A DDS circuit takes advantage of the ever-increasing speed of digital circuits to implement fast waveform generators. It does so by outputting values from a look-up table (LUT), which stores the desired waveform in discretized values, and converting them to an analog signal by a digital-to-analog converter (DAC).

In this lab, we are going to use the Arduino UNO to generate and send data to an MCP4921 DAC in order to generate a triangle wave. Since DAC outputs are discrete in nature, we are really just approximating the triangle wave with a stair-case waveform (Fig. 5). Obviously, the higher the resolution of the DAC, the better this approximate is.

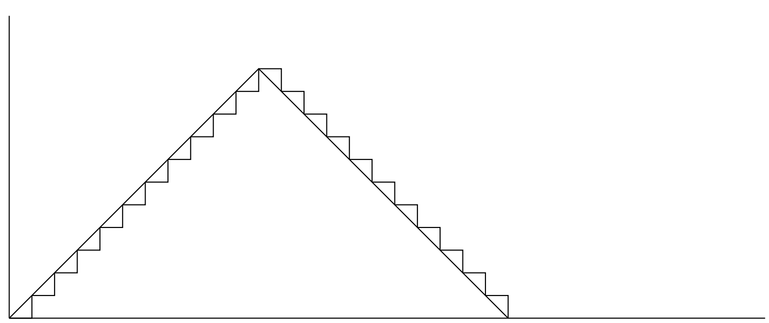

Figure 5: Approximating a triangle wave using the stair-case output of a DAC.

1. Using Fig. 6 as a reference, connect the Arduino UNO to the MCP4921 DAC. Connect VDD to 5 V and both AVSS and $\overline{\text{LDAC}}$ to ground. A

0.1 µF and 1 µF capacitors could be used at VREF to minimize noise at the reference. Connect the UNO with the computer using the USB cable, which is used both to program the UNO and power it up. It is very important to keep your circuit (ICs, resistors, capacitors, wires, etc) organized. Use as short of a wire as possible between two connection points. Refer to Fig. 7 for an example layout.

Figure 6: Schematic of the connection between Arduino UNO and the MCP4921.

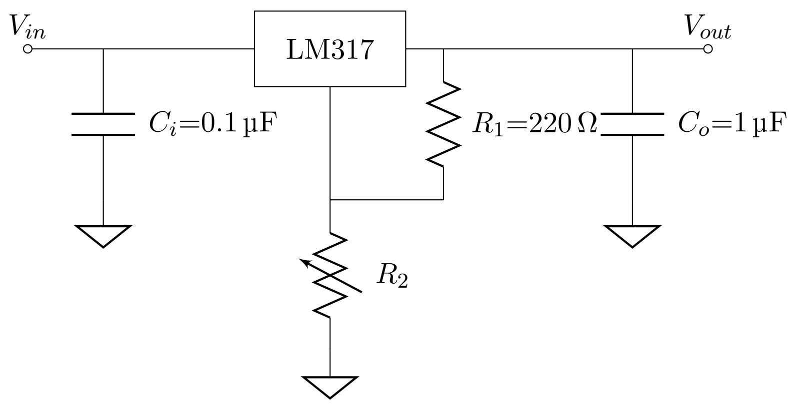

Figure 7: Connection of the Arduino and the DAC chip. Only the top DAC section of the breadboard needs to be assembled for this lab. Power supply is also not shown in this figure.

2. Once the circuit is built, open the Arduino IDE, create a new sketch, and input the code "triangle.ino" (`https://github.com/ucdart/UCD-EEC134/blob/master/Lab1/triangle.ino`). **Make sure you read through and understand the code.** You may want to read the datesheet of MCP4921 to understand its SPI control protocol. Compile and upload the code to the Arduino Uno as explained in the pre-lab tutorials.

3. In your report, include a screen capture of both the triangle (at VOUTA) and sync (at SYNC) output signals on the oscilloscope. Record their amplitudes and periods.

4. Disconnect the VREF pin of the DAC from the 5 V regulator and connect it to an external 2.5 V supply. How does changing the reference voltage affect the output signal?

5. How can you modify the code to change the amplitude and period of the output waveform? What is the fastest triangle wave you could generate? What do you think is the limitation to going even faster?

6. Modify the code to generate a sinusoidal wave. What is the highest frequency sinusoidal wave you can generate? The following link may give you some hints. `http://interface.khm.de/index.php/lab/experiments/arduino-dds-sinewave-generator/`

7. **(Challenge!)** Research and propose a design that will allow you to generate a 100 kHz triangle and/or sine wave.

4.3 Low-pass filter

In this part of the lab, we will implement an active low pass filter (LPF) with a gain stage. Fig. 8 shows the schematic of this part of the circuit. Fig. 9 shows an example of the actual circuit on a breadboard. It also filters out any spurious signals above a certain threshold frequency (15 kHz in this design). To get the maximum resolution from the digitizer, the maximum strength of the signal should be less than the allowed input voltage range of the digitizer.

In our simple radar system, we are going to use the soundcard of the PC as the digitizer. The amplified and filtered IF signal is fed to the right channel of an audio card, which is conventionally connected using the red wire of a 3.5 mm stereo jack or the ring of the connector (Fig. 6.10.3).

Figure 8: Schematic of the active low-pass filter.

Figure 9: Schematic of the active low-pass filter.

4.3.1 Gain stage

We will use the Texas Instrument OPA4228 quad Op-Amp for both the gain stage and the active LPF. A pinout diagram of OPA4228 is shown in Fig. 6.10.4.

Identify the gain stage of the audio amplifier from Fig. 6.10.1. What is the expression of its gain? Notice the +5V bias does not affect this result, rather it simply shifts the bias of the overall amplifier since a single 9V supply (battery) is used vs a double +/- supply. Build just the gain stage of the audio amplifier. Remember to power the op-amp chip by using 9V for V+ and ground for V-. Use the 9V battery as the supply for all the measurements to achieve better noise performance. Remember to disconnect the battery when measurements are not being made. The gain is controlled using a 20 k potentiometer (POT). To test the circuit, input a 1-kHz 100 mVpp sine wave to the IF input using a signal generator. What is the maximum gain before clipping occurs, and what is the corresponding peak-to-peak voltage at the output? The audio card on a typical laptop is designed to take in line level signals. Notice that this limits the sound card input signal strength to approximately 3 Vpp. Adjust the pot to achieve 3 Vpp on the oscilloscope at 1 kHz. Tabulate and plot the gain from 100 Hz to 1 MHz while keeping the pot fixed. Find the 3-dB cutoff frequency or the 3 dB bandwidth at this gain. Use at least 20 data points.

4.3.2 Active LPF

Build the rest of the audio amplifier circuit, i.e. the active LPF. To measure just the filter response, disconnect R4 and R14 from the circuit (Fig. 6.10.1). Input a 3-Vp-p signal in place of R4 and measure the output in place of R14.

R14 will only be needed as a current limiting resistor when connecting the amplifier to the laptops audio card. Again tabulate and plot the frequency response from 100 Hz to 20 kHz. What is the cutoff frequency of the filter? The overall filter consists of 2 low pass filters. What is the order of the overall filter and how is it determined from the schematic? Is the bandwidth of the filter tunable? How would one increase or decrease the cutoff frequency? What limitation does the cut-off frequency of the LPF place on the radars performance? (Hint: what happens when your object is too far away)

4.3.3 Gain Stage + Filter

Reconnect R4 (R14 is still disconnected) and measure the overall response of the amplifier. Input a 100 mVp-p sine wave to the IF input. Adjust the pot for a 3 Vp-p output at 1 kHz and once again sweep the frequency from 100 Hz to 20 kHz while tabulating the voltage output and the gain. What is the cutoff frequency? Compare the results to previous measurements. Tabulate and plot all of the results for the report. Reconnect R14. Break the audio cable and connect the LPF output to the right channel and the SYNC signal (from the modulator) to the left channel.

4.4 Analog-to-digital converter

4.5 Signal processing

Use teensy 3.1 to sample and process data? sounds like too much work...

Using the laptop

References

[1] Leslie Lamport, *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.