ECEN 4013 Project 2

Determining Boltzmann’s Constant

Spring 2021

Group 7

By:

Michael Thompson

Steven Gaiko

Jackson Ball

Bo Rogers

Index

Problem Statement

The goal of this project was to build a circuit that can estimate Boltzmann's constant using thermal noise across a resistor. After designing a circuit that can do this, we were also tasked with estimating the resistance of an unknown resistor connected to our circuit by using our data from known resistor tests. In order to accomplish this task, we were given a basic design prompt which included two amplifier circuits, one frequency filter, and an ADC. We ended up adding another amplifier, an offset summing amplifier, and our software development to these required components for our final design.

Key Project Parts

(1) Preamp

(2) 2 Post-amps

(3) Filter

(4) Offset Summing Amplifier

(5) ADC & Software

Project Function Summary

I mentioned the components which were recommended for this project along with the components we added in the outline above, but I didn't explain why each of these components is needed or what they do for our project.

Our initial signal comes in the form of noise across a resistor which is attached separately within an isolated box. This noise's signal tends to have an rms value in the nV range. The preamp then takes this noise signal and ideally amplifies it by a factor of 1000. Though, our preamp's exact specifications and capabilities can be found later in the report and are affected by part efficiencies. The signal from the preamp should then have an rms value somewhere in the mV range.

The signal is then sent through a filter circuit. Our design used a Bessel filter with a bandwidth of about 10kHz, exact data on our filter can be found later. The filter should also ideally have a gain of 1 so that it doesn't affect our signal other than cutting off the high and low frequencies. We ended up having a different gain through our filter, which is also referenced later in the report.

The next step of the circuit should normally be just a single preamp circuit that also amplifies our signal by a factor of around 1000, but we had quite a few troubles with getting enough gain out of just one amplifier early on and added another. So, we used two cascading preamps to amplify our signal necessarily. Later, we realized we would also need to change the amplification of our preamps deepening on our test resistor, but that is explained in more detail later.

Next, before we can send our signal through an ADC, we determined we would need an offset summing amplifier because we realized the ADC we would be purchasing would only read in positive voltages. Therefore, we designed a summing amplifier after our last preamp that could be adjusted so that our final signal was an oscillating wave completely above 0 and within our ADC's measurable range.

Finally, as hinted at previously, an ADC which we purchased, thank goodness, would need to be at the end of our circuit to read the amplified noise signal into a computer. The computer would then be used to appropriately store and evaluate the data to give us Boltzmann's constant estimates. Now, I talked about both the ADC and software here because of the way we tackled our need for an ADC. Initially, we bought a surface mount ADC and had an outside 10MHz oscillator to function as its clock, but we were unable to solder the surface mount piece, so we fell back on a chip one of our members already owned. The chip was placed on a breadboard and was capable of acting like an ADC, using an internal clock, and could even be coded in arduino. In the end, our ADC's implementation and software implementation were merely one and the same.

Now, each of these components is explicitly described and picked apart further in this report, but the surface level importance and main functions of each component are important to know going into this report. Furthermore, all math and data analysis techniques used in calculating Boltzmann's constant will also be explained as you go on.

Team Structure Roles

Point of Contact and Progress Report Head...Jackson Ball

Project Report Main Author/Coordinator………………………………………..…..Steven Gaiko

Part Purchasing Manager………………………………………………….............Jackson Ball

Project Design Roles

Head of Pre and Post-amp Design……………………...…....Jackson Ball & Michael Thompson

Head of Filter Design……………………………………....................……………Bo Rogers

Head of Offset Summing Amplifier Design..…...……...Steven Gaiko

Head of ADC and Software………………………………………....………...Michael Thompson

Head of PCB Design……………………...……………………………...............Jackson Ball

Head of Circuit Building and Testing……………….............................……………Bo Rogers

Assistant-Head of Circuit Building and Testing………….....................……..Michael Thompson

Circuit Building and Testing Assistant………………………………...………….Steven Gaiko

Design Strategy, Constraints, & Implementations

Pre and Post-amps

Introduction

Our preamp will be placed at the start of our circuit, just after our test resistor, and will be used to amplify our initial noise signal from the nV range up into the mV range. In order to achieve this without disrupting our input noise signal in any way other than amplification, our preamp needs to be able to limit noise as much as possible while amplifying our signal as much as possible. Because this amplification will be over a large range of frequencies, we also want a very high gain bandwidth product, and with the huge gain needed, any offset produced by the circuit's op-amp will be amplified and become very visible. In order to limit the offsets and other effects to our noise signal, we were forced to be a bit picky with our op-amp selections while also needing to be a bit tricky with the way we designed our op-amps. Then, even with the problems that arose later, almost all of the same qualifications and restrictions were then applied to the post-amp design. [Steven]

*It is also important to note that both Michael and Jackson worked on these amplifier designs, and the following information includes both people's own wordings out together into a single section.

Pre and Post-amp Design by Michael Thompson and Jackson Ball

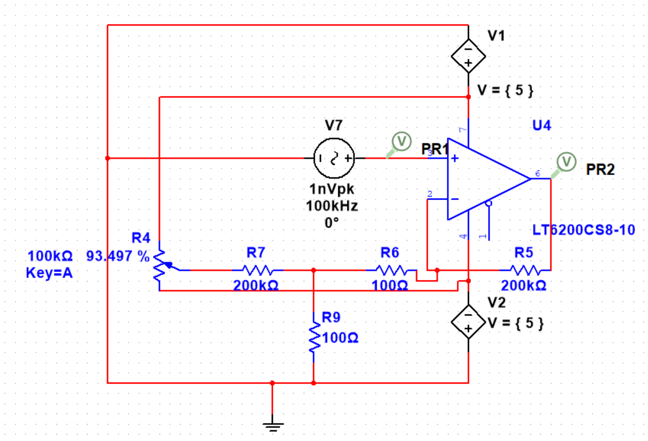
Part Selection:

Taking all the restrictions and qualifications from above into account, we decided on a LT6200CS8-10 op-amp for our initial amplifier designs. This op-amp has a noise voltage of about 1.9nV/sqrt(Hz), a max offset of 500uV, a gain bandwidth product of 100MHz, and a slew rate of 25V/us. Though we didn't mention the importance of slew rate above, we decided on a high slew rate op-amp in order to take advantage of our initial ADC's absurd processing speeds. All in all, this op-amp seemed to be the perfect op-amp for our application. [Michael & Jackson]

Addressing Limitations:

Though we found a resistor with an offset as low as 500uV, whenever we used multisim to test our amplifiers, the offset from our preamp seemed to be causing issues with our final noise signal, so we added a potentiometer to help calibrate the circuit. The new addition to the circuit would act as a current injector and consisted of an extra 100Ω, 200kΩ, and a potentiometer, all of which can be seen implemented in **Figure 1**.

Figure 1



The use of a potentiometer would allow us to adjust the current into the -Vin pin on the op amp. By being able to control the -Vin pin, we would be capable of adjusting the potentiometer until -Vin and +Vin have almost exact input currents. This would then allow us to adjust for the op-amps internal offset effect. Later, we also added a DC blocking capacitor in series with the output of the amplifier to adjust for the remaining offset produced by the op-amp. However, we found that the DC blocking capacitor could block most undesired offsets on its own, and we ended up removing the potentiometer and resistor design from our final circuit. [Michael & Jackson]

<u>Testing and Adjustments:</u>

While testing the amplifiers, we could not get the LT6200CS8-10 to work properly. With an extremely low input, like the nVs of noise we plan to amplify, the amplifier outputs would shoot immediately to the positive rail value. Therefore, we had to find a different op amp that would work. Luckily, we had two leftover LT1007 op-amps from some of our filter tests. These op-amps worked perfectly for the preamp, but they would not work for the post-amp. It seemed that the intended gain was too high for the LT1007 in the post-amp and resulted in the LT1007 dragging its output to the negative rail value. The same issue happened when we tried an ADA4622-1 op-amp. This did not make a lot of sense to us because the preamp had a gain of 1000 and the post-amp had a lower gain, but we eventually found that if we lowered the gain of our post-amp, the LT1007 would not pull the output to its rail value. However, we had to lower the post-amp's gain so much that we were not getting enough gain for our tests. Therefore, we added an ADA4622-1 op-amp in series with the LT1007 and split the gain between the two op-amps. This allowed us to amplify the post-filter signal as desired without running into the rail issue we were having before.

Later, we found that the changing of our test resistor would also change our output's amplification. To address this, we made one of our last post-amp's gain resistors hot-swappable. In the end, we used this feature to quickly rotate through different post-amp gain values and find the resistors necessary for each test resistor value. This technique helped us quickly find all the resistors we would need for our Boltzmann's approximations. More shockingly, we were also able to use this technique when trying to estimate the value of a mystery test resistor by eyeballing the oscilloscope while cycling through post-amp resistor values.

Finally, though our built design includes all of these features, like the two post-amps and swappable parts, it is worthy of mentioning that these parts were never included in our multisim schematic designs or PCB design and were all implemented as we went through the project. [Michael & Jackson]

Band-pass Filter

Introduction

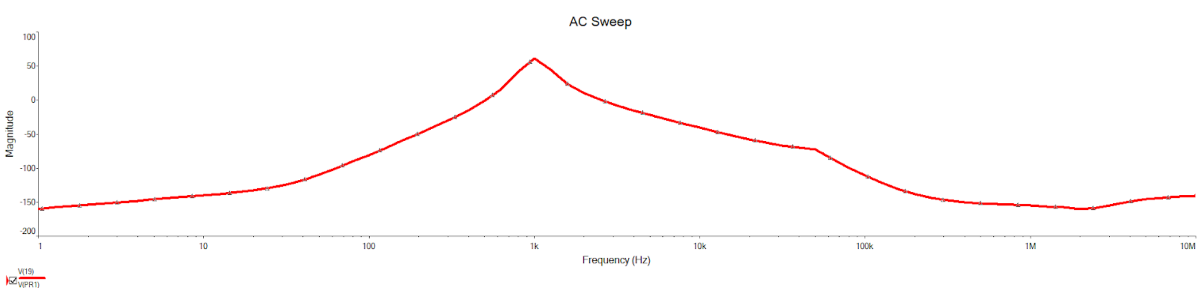
The filter in our circuit will be designed in order to limit the frequencies which will be in our final noise signal. First, we must amplify our initial noise in order for the signal to be filterable. We then use the filter to create a predictable frequency range, bandwidth, that will only allow signal frequencies within that range to be output. In the end, we want our filter to have consistent gain throughout its bandwidth with quick frequency cutoffs. What kinds of filters can do that for us? [Steven]

Filter Design by Bo Rogers

<u>Design Decisions:</u>

For designing the filter, we quickly decided that we wanted to use a Bessel filter because of its steep and consistent magnitude response. From there, we considered a narrow band and wide band filter. Our initial idea was to look into a narrow band filter. The narrow band filter was beneficial due to its easy design with standard capacitor and resistor values, but the pass band was too small to work with for what we need. The magnitude response of the narrow band filter can be seen below in **Figure 2**.

Figure 2



Our main goal with our filter was to have a flat, consistent gain response section for the filtered noise to travel through, and as you can tell, the narrow band filter's gain response depicted more of a spike than a flat and consistent gain response. This would be much harder to work with in our circuit's design, so we chose to use a wide band filter.

Once we decided on the wide band filter, we also had to determine how wide we wanted the pass band to be, and this bandwidth choice rellys on both the gain bandwidth of the pre and post-amps, as well as the sampling rate of the ADC. Our initially chosen ADC had a sampling rate of 500kHz while the pre and post-amps had gain bandwidths of 1.6GHz, and these values would allow us to go quite large on our bandwidth. However, we chose a pass band from only 1kHz to 10kHz in order to be on the safe side. This would allow us to over sample the noise signal while being well within the bounds of our gain bandwidth.

Once we decided on a bandwidth, I used the normalized resistor and capacitor values for a Bessel filter to create the base schematic. From there, I denormalized the circuit by using several different resistor and capacitor values until I found values I could implement with standard parts. The final design was made to have no gain; this allows us to control all of the circuit's gain through the pre and post-amps only. [Bo]

<u>Part Selection:</u>

I decided to use 1% resistors on the filter design to try and keep the filter's gain and bandwidth as close to the simulation's; the capacitors were eventually picked up from the stockroom supply and not purchased online. Then, the op amps used in the filter didn't have extremely high performance requirements, so we could be a bit more flexible with that decision. We ended up going with and ordering LT1007 op-amps because of their solid, well rounded function. [Bo]

Final Design:

The schematic of our final filter design is shown below in **Figure 3** and its magnitude response is shown in **Figure 4**, ranging from 1Hz to 1MHz. [Bo]

Figure 3

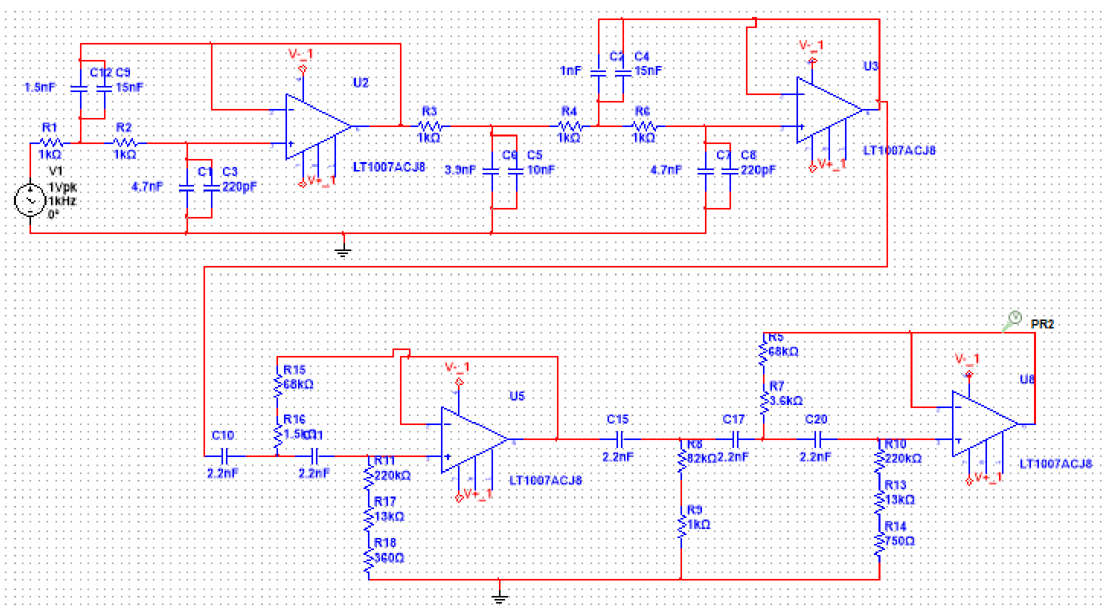
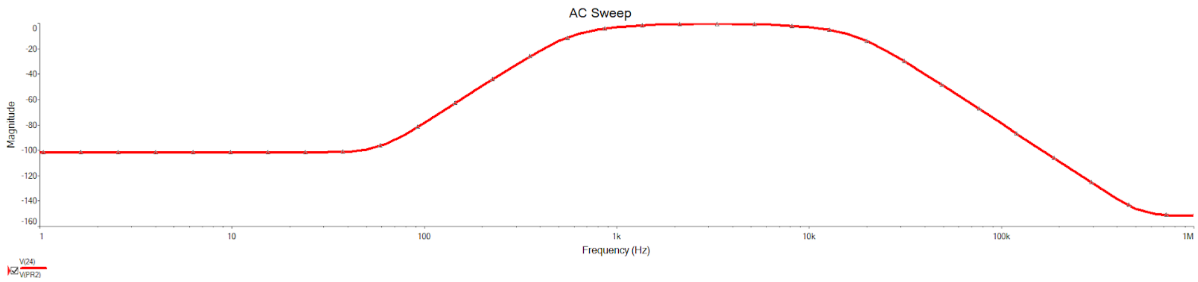


Figure 4



Offset Summing Amplifier

Introduction

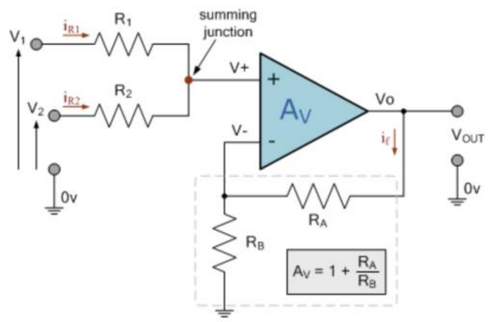
Initially, we thought an ADC would be capable of taking in positive and negative voltage inputs, but all of the ADCs we could find online were only capable of taking positive voltage inputs. Therefore, we decided to implement an offset value for our noise signal so that it would range, for example, from 0v to 5v instead of -2.5v to 2.5v. Our goal with this offset was to offset the signal by half our ADC's max voltage input because our goal was to amplify our noise close to the ADC's boundaries as possible, and this would be the easiest way to shift our noise. Really, our goal was to build any sort of offset that was capable of shifting our AC-like noise wave positive, but it was our design decision to use a summing amplifier for this offset. [Steven]

Offset Summing Amplifier Design by Steven Gaiko

<u>Design Restrictions and Part Selection:</u>

After choosing to use a summing amplifier as an offset for our noise signal, we had to address a few key parts of the summing amplifier. The base design for making a summing amplifier can be seen in **Figure 5** below.

Figure 5



Traditionally, summing amplifiers are built as inverting amplifiers because they are easier to build and more adaptable, but for our case the non-inverting amplifier worked best.

The main things that need to be addressed when looking at this design are the resistor values along with selecting the op-amp for the circuit. We want this adder to work as an offset, so we want Vout = V1 + V2. In order to accomplish this, we only have to do two things; we need to make sure R1 = R2 so that the summing junction functions properly and creates V+ = (V1 + V2)/2, and we also need to make sure that Ra = Rb so that the circuit has a gain of 2, Av (gain) = 1 + Ra/Rb, and Vout = 2V+ = V1 + V2.

In the end, we went with four 10kΩ resistors with 1% tollerence from the stockroom and one LT1007 op-amp, which we ordered along with the op-amps for the filter. [Steven]

Implementation and Adjustments:

Though the basic design of the summing amplifier stayed the same from our initial plans and schematic, as seen before in **Figure 5**, to the final implementation on PCB, there were still a few different things we had to address. First off, we had to find a way to get half of the ADC's maximum input voltage as one side of the amplifier. Oddly enough, finding half the ADC's max input wasn't hard because it's max input was equivalent to the voltage of it's high supply rail. Using it's high rail supply and a simple resistor divider, we were able to easily implement the necessary value of our offset.

Later, we ran into a few other issues that also needed a bit of work. At one time, we noticed that our offset was not showing nearly as high on the oscilloscope as it was supposed to be. We think that our filter was causing some sort of offset that might not have been completely blocked by the DC-blocking capacitors throughout our design. In order to combat this, we increased the offset amount by changing the values of the resistor divider before the amplifier's input. This in turn caused some issues in the math of our code because we had previously accounted for the offset by subtracting by a set offset, but we changed the code to determine the signal's offset on its own. Though, Michael will go a bit more into detail about the code and ADC later. Eventually, we even added a buffer op-amp circuit between our resistor voltage divide and summing amplifier in another attempt to clean up our signals at one point.

Therefore, our final implementation of the summing amplifier consisted of a resistor based voltage divider which gave us a ratio of the ADC's max voltage. This voltage divider would then connect through a buffer op-amp circuit to the summing amplifier. The summing amplifier takes both the desired offset DC voltage and the AC-like noise signal from the post-amps and offsets the signal for our ADC's use. [Steven]

Testing:

When testing the summing amplifier in the lab before connecting it to the rest of our circuit, the amplifier was giving us nearly perfect results with no gain or offsets induced by the op-amp. However, it is possible that once the summing amplifier was connected to everything else on the PCB that it could have been one of the causes of random offsets and inconsistencies in our final noise signal. [Steven]

ADC and Software

Introduction

We have talked about both our ADC and Software throughout the entirety of this report because they are really the spine of our project. Everything that our project accomplishes physically has to be converted to data and properly handled by the ADC and our software. Our initial ADC plan, as mentioned before, was to place a surface mount ADC onto our PCB. After trying to apply the ADC for two days of almost 4 hours each at the Endeavor, we decided to give up on the surface mount ADC. Luckily, Michael had a perfect replacement for us that could be connected using breadboards and coded using arduino. In the end, we had a breadboard ADC which we connected serially to our noise signal at the end of our circuit, after the summing amplifier. The ADC reads in the noise voltage signals as binary values and then our software is able to properly assess that data. [Steven]

ADC Design By Michael Thompson

Early Design Issues:

The initial plan was to use a 16 bit 500ksps ADC (ADS8322) that would be surface mounted to the PCB. However, once all the parts were received, we found that the ADS8322 was so small that the precision soldering machine at ENDEAVOR could not even solder it. So we had to come up with an alternate plan. The next option was an ESP32 with it's built in 12 bit ADC.

The ESP32 has two cores for multithreading. Initially, we used one core for reading the analog pin and the other for serialy sending the data. However, the main core must have some time to perform the watchdog routine or else the ESP32 will restart. This restart sends an emp which is picked up by the resistor and is shown as a large couple of pulses in the noise. This would cause a lot of issues with our data, so we set the ESP32 to read and serialy send the data on core 0 while leaving core 1free to run the watchdog routine. [Michael]

Early Testing:

With the ESP32 set up as an ADC, the first thing to test was the speed. To do this, time was taken before the sample was taken and sent, and then after. The difference between the times in microseconds was then sent serially. The time was between 10-20us which is 50k-100ksps. That value will be a bit long because taking the time and sending it also takes time.

The second thing tested was the accuracy of the voltage reading. To do this a constant voltage was applied and the samples were converted from bits to voltage to compare. In the end, everything looked to be in place, and though the sample rate wasn't what we might have gotten from the ADS8322, we were just glad to have a working ADC. [Michael]

Software Design by Michael Thompson

<u>Original Design:</u>

The original plan was to write all the code in python and have it run on a raspberry pi 4. To control this pi without a monitor, a SSH tunnel was set up to be used with a LAN connection created with an ethernet cable from the laptop to the pi. The major parts of the code were: UI, Log (calculation), ADC read and the MASTER. Once the code was complete, the ADC read was tested and found to be way too slow. To fix this problem, that section was recreated in C++ and a bash script was made that compiled the C++ file and then ran the main python file. Instead of having the python script call the python ADC read function it would now create a temporary "txt" file with all the information the C++ script needs. Then the python script would call the C++ "exe" file that is created when the file is compiled. This C++ script would then read in the temporary "txt" file and then delete it before reading and logging the ADC data. [Michael]

<u>Secondary Design:</u>

In the end the ADC we were planning to use was too small to fix to the board, so the C++ script was no longer needed. This no longer needed to be ran on the pi so all code was moved to a laptop. Instead of reading the ADC directly, a serial stream from an ESP32 needed to be read. The easiest way to do this is using a free program called PuTTy.

PuTTy is used to open all kinds of connections and log incoming data. In this case it opened the serial connection to the ESP32. PuTTy was chosen because it allowed for an easy way to log the serial data without having to dive into Python serial libraries. To run PuTTy, the line that ran the C++ “exe” was replaced with PuTTy’s “exe” path.

<u>Software Functions:</u>

The final code is split into three files: UI, LOG and MASTER. The UI file holds the page class which is used to populate the terminal screen with the desired information in the correct locations. Each page has a title, task, general contents and command line variables which are shown when the show() command is called.

The LOG file holds the log class which is used to manipulate log files and calculate the rms voltage from the log and both the boltzmann constant and unknown resistor. First, the test resistor, tested filter gain, temperature, noise equivalent bandwidth and resistor values that make up the gain are entered in a file called info.v. Every time the log class is created, info.v is read in and all gain values calculated. To calculate rms voltage, first all the data in the log is averaged to establish the offset. Then the code takes in each data entry from the log and subtracts the offset. Then this sample is converted into a voltage value using 1V = 1575 bits and divided by the total gain and then squared. Each of these converted samples are then added together, averaged and then the square root is taken. This result is the rms voltage. The following equation is used to calculate Boltzmann’s constant:

$$\square_{\square} = \frac{\square^{2}_{\square\square\square}}{4 * \square\square\square\square * \square\square\square * \square_{\square\square\square\square}}$$

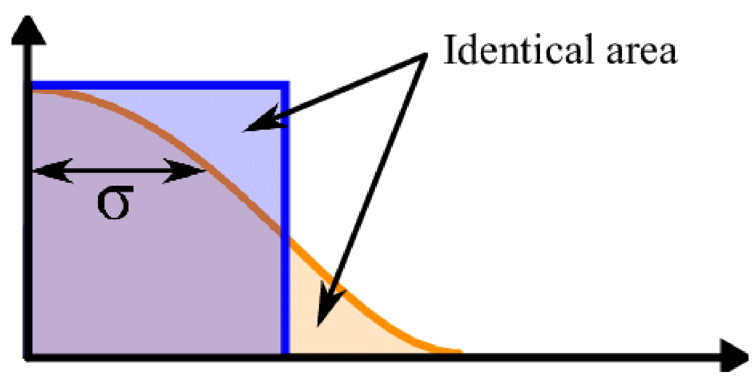
The following equation is used to calculate the unknown resistor:

$$□_{□□□□} = \frac{□^{2}_{□□□}}{4*□□□□*□□□*□_{□}}$$

The unknown resistor is calculated twice. First the $□_{□}$ is the average of all previously calculated Boltzmann’s constants. Because these constants very greatly over the range of 1k-100kΩ, the second calculation takes the first value and finds the Boltzmann’s constants from the closest resistor values to the first guess and uses those for $□_{□}$. [Michael]

One important variable in the calculation for Boltzmann’s constant is the noise equivalent bandwidth (NEB). It is essentially a representation of a brick wall filter that allows the same amount of power to pass through as our filter design. In **Figure 6** below, the NEB is represented by the blue line.

Figure 6



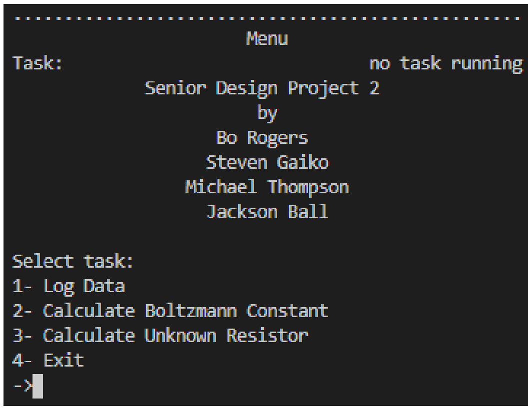
The NEB allows us to use a number to represent the entire signal that the filter allows to pass through. To calculate the NEB, I used the following equation.

$$□□□ = 1.11 * (□□□□□□□□□)$$

$$= 1.11 * (□□□\ □□□□□□□\ □□□□. - h□□h\ □□□□□□□\ □□□□.)$$

[Bo]

The MASTER file is a finite state machine with four states which each have their own page object which contains information about the state. The first state is “menu”. This state is used to select the next state and to select the log file to be used at that state via the command line, and this state can be seen in **Figure 7** below. [Michael]



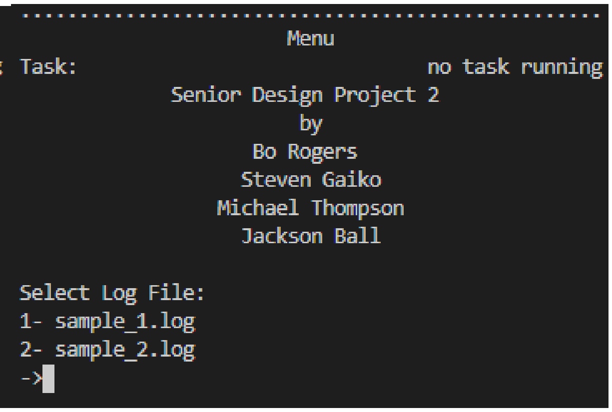
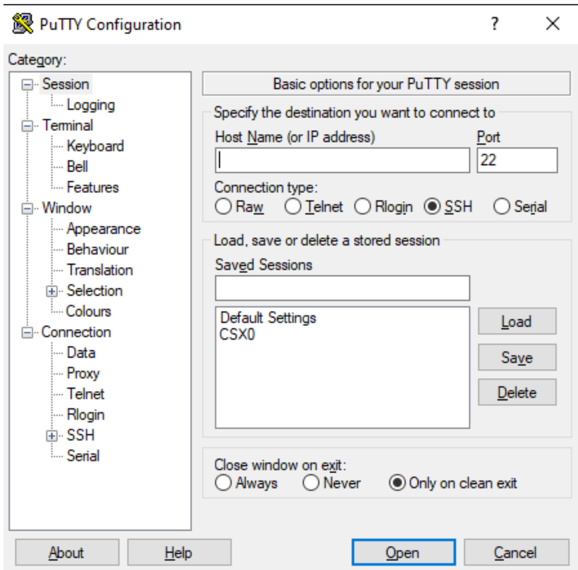


Figure 7 **Figure 8**

The second state is “log data”, as seen in **Figure 9** below. This state opens PuTTy and displays the settings that need to be set in PuTTy. Once PuTTy is closed the PuTTy output log is cleaned of characters and placed in the desired sample log file and then the state is set back to “menu”, as seen in **Figure 8** above. [Michael]



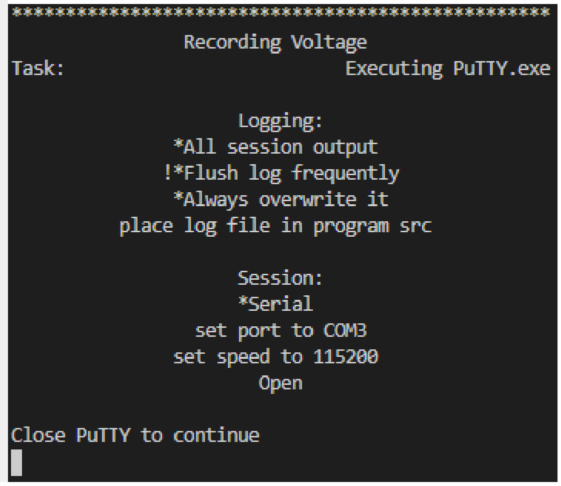


Figure 9

Figure 10

The third state is “calculate boltzmann’s constant”. This state calculates the rms voltage from the selected log file and displays the screen shown in **Figure 10** above. [Michael]

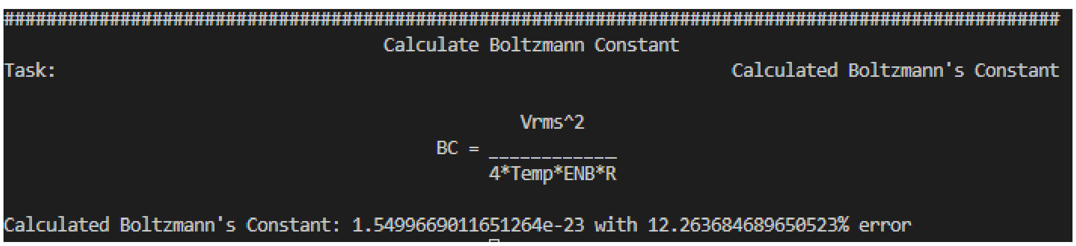


Figure 11

Then Boltzmann’s constant and % error is calculated and returned to the page as seen in **Figure 11** above. The fourth state calculates the unknown resistor using the log of Boltzmann constants and the selected log and displays it as shown in **Figure 12** below.

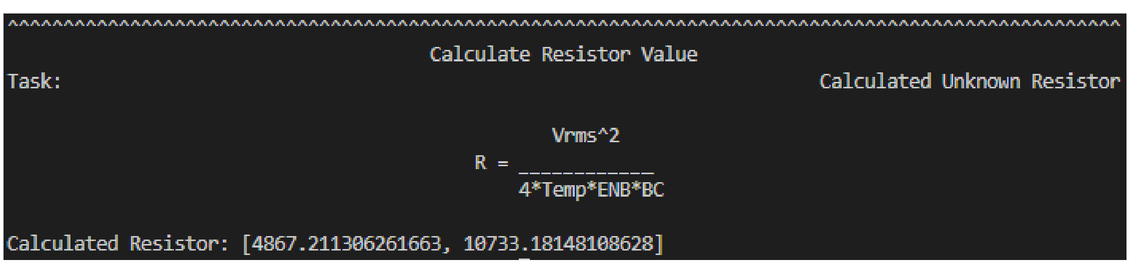


Figure 12

Testing:

Throughout the creation process, the code was constantly tested. The UI was tested by altering page values and then showing the page. The log was tested by using print statements to verify info.v was read in correctly and to make sure the values were being logged and calculated correctly. PuTTy was tested by comparing the serial terminal to the PuTTy log. The finite state machine was constantly tested by running the program and going through all the states in multiple different orders. [Michael]

PCB Board

Introduction

From the start of our project, we knew we were going to have a final design on a PCB board. This was because of our need to minimize noise and outside interferences on our noise signal. Therefore, we couldn't use breadboards for the entirety of our project, and we needed lower tolerance parts. In this case, we decided to design our circuit on a PCB board. In the end, we had to add our ADC using a breadboard; one of our op-amps' pads got ruined on the PCB and had to be implemented using a perf board; we had to add an additional post-amp on a perf board; and we had our test resistor connected using a shielded cable on a perf board inside an aluminum casing. However, the majority of our project was designed onto a PCB designed by Jackson, and the exceptions to this were simply solutions to problems that arose throughout our project. [Steven]

PCB Board Design by Jackson Ball

Initial Design Issues:

Once the multisim simulation of the project was finished, I used Ultiboard to create a PCB schematic we could get printed. The ADS8311YB was the most difficult to place because of how small it was and that it is a surface mount chip. We had some concerns about soldering it but we hoped to use the Endeavor precision soldering facilities. However, they were unable to work with our part and we had to work around it. We also had to work around the fact that our 20-pin connector did not have a model in Ultiboard, but we were able to use two 10-pin connectors instead and aligned them to work as one.

Final Decisions:

We ordered the board from JLCPCB after researching several options. We chose JLCPCB because of their pricing, shipping speeds, and their reviews impilded better quality than the competitors.

When the board arrived we were able to solder everything but the ADS8311YB. However, some of the LT1007 were not working as expected during testing, and we found that their shutdown pins were not grounded properly and had to solder an extra wire on top of the board to fix them. In the end, some parts of our circuit had to be moved to a perf board when new parts were added or old parts were replaced by parts with different footprints, but the majority of our project stayed on the board. [Jackson]

Final Design:

The two figures below show the final design of our PCB board in Ultiboard.

Figure 13

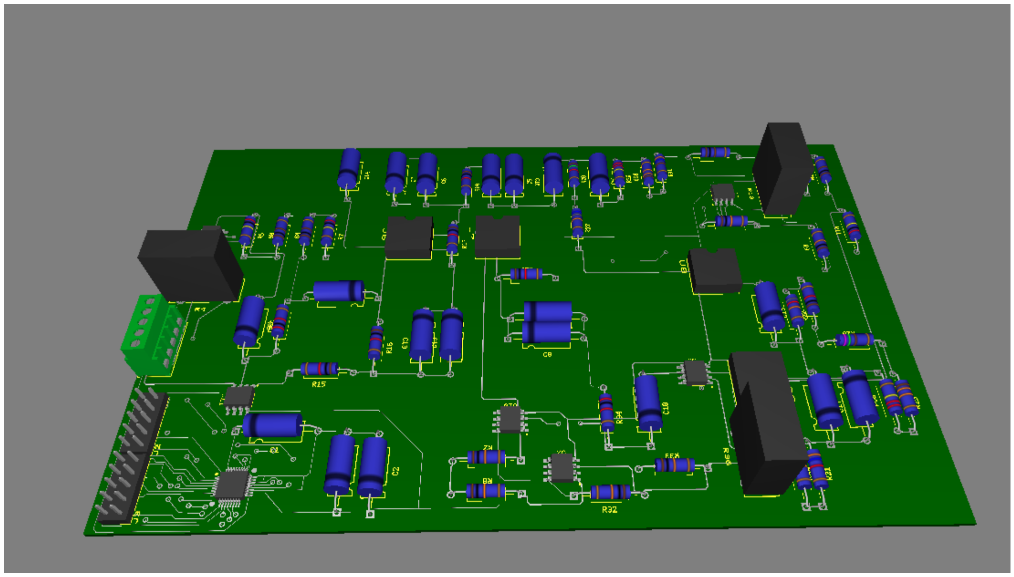
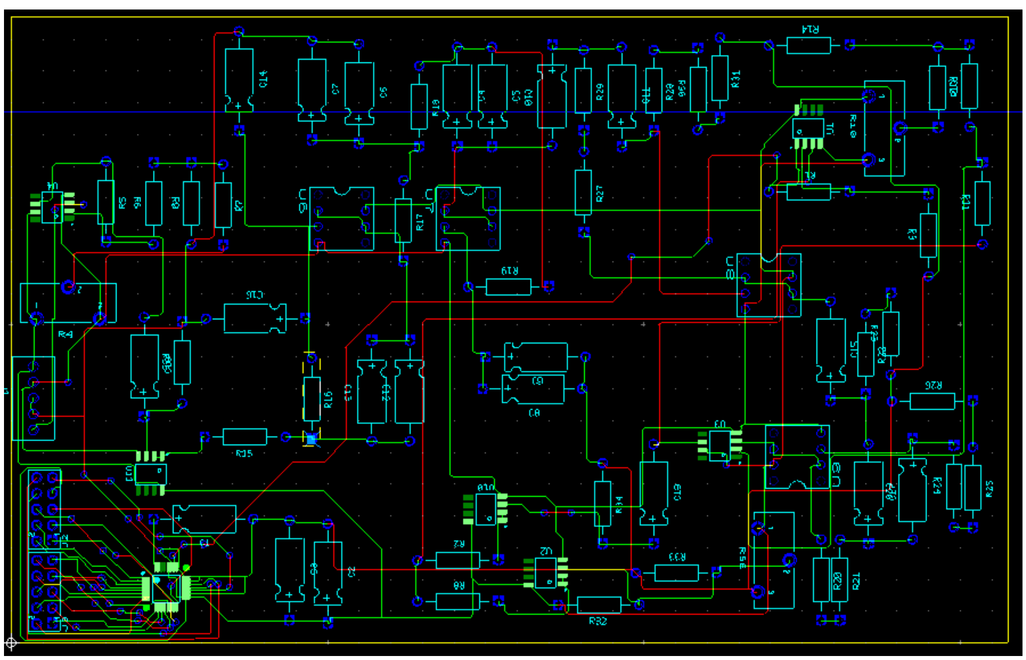


Figure 14



Project Assembly and Testing

Introduction

The majority of our project was assembled and tested by Bo. Michael also contributed quite a bit of work on the assembly and testing of the final circuit, and Steven helped with the final set of testing and troubleshooting.

Most of our tests revolved around checking if a piece of the circuit, summing amplifier or preamp for example, worked on its own, and then we would connect that section to our PCB and see how it functioned in the circuit. After everything had been implemented and the parts which needed to be added using perf boards were all in place, we began testing our system as a whole. By analyzing different checkpoints in our circuit using an oscilloscope and analyzing the data gathered by our ADC and computer, we were then able to hunt down issues throughout our design and slowly shape our project into its final form. [Steven]

Pre-ADC Building and Tests by Bo Rogers

<u>Filter Setup and Testing:</u>

I tested the preamp, filter, post-amp, and adder in specific stages to minimize possible errors and make circuit integration easier. First, I built the filter on the PCB. This would allow me to test the filter without possible issues with the preamp. After each component was soldered on the PCB, I tested the filter by inputting a 1V sine wave from the function generator and reading the output from different frequencies. Frequencies below 1kHz and above 10kHz should be filtered out, so the test was simple and the filter worked well. However, the gain was smaller than we expected in the pass band. It only reached 0.92V as a max and slowly decreased until it hit the low pass cutoff. This was odd, but we averaged the values from 2kHz to 9kHz and used it in our final gain calculation. The table displaying the pass band gain is shown below in **Figure 15**. [Bo]

Figure 15

Frequency Input	**Volts P-P**
1k	Cutoff
2k	0.92
3k	0.92
4k	0.92
5k	0.9
6k	0.86
7k	<u>0.84</u> - 0.88
8k	0.8 - <u>0.82</u>
9k	0.78
10k	Cutoff
<u>AVERAGE</u>	**<u>0.87</u>**

-

To calculate the NEB from the built circuit, I found each of the cutoff frequencies by measuring where the output of the sine wave measured 0.707V. The values were calculated as ~1kHz and ~11.4kHz which resulted in an NEB of about 11.544kHz. [Bo]

Preamp Setup and Testing:

After the filter was working properly, I moved on to test the preamp. I could test the preamp separately from the other portions of the circuit because it was the first main stage. The preamp is just a simple op amp with a gain of 1000, so it was easy to build on the PCB. My testing plan was to use the function generator for a sine wave input and read the output. Once that worked, I planned to input our test resistor signal into the preamp to see the amplified noise.

When I began my first stage of testing the preamp, I ran into issues. For some reason, the output of the amplifier would go directly to the positive rail. There was a floating pin on the amplifier, so we tried connecting the floating pin to both power and ground. Unfortunately, neither attempt solved the issue. We are still unsure what caused this to happen. As a backup, we used an extra LT1007 and soldered it to a perf board because it wasn't in the same package as our initial preamp.

From there, we jumped the output of the new preamp to the blocking capacitor on the PCB. Once we replaced the amplifier, both tests worked. The noise output was amplified to ~12mV peak to peak, around what we wanted. [Bo]

Post-amp Setup and Testing:

Once the preamp and filter were working correctly, I soldered the post-amp to the PCB. We replaced the LT6200 with an ADA4622-2 from the stockroom because it was in the same package and had the same pinout. Initially, the post-amp also had a gain of 1000. My plan for testing the post-amp was to use the noise signal from our test resistor. We knew that the signal we wanted would be pushed through the filter so there was no reason to test it separately.

When I began testing, the output from the post-amp would go directly to the negative rail. After running numerous tests to find the issue, I concluded that the gain of the post-amp was too high. When I lowered the gain, the signal was read correctly on the output. However, it wasn't amplified as much as we wanted. From there, I cascaded another amplifier with our initial post-amp using another LT1007 that we had left over. The gain didn't have to be very high and the two post-amps worked well.

I later soldered female pins in place of the feedback resistors for both post-amps in case we needed to change the gain for different test resistors. This proved to be extremely helpful when testing different gain values.

Adder Setup and Testing

The final stage of the circuit manipulating the signal before it reached the ADC was the summing amplifier, also called the adder. The adder was a simple design so it was easy to solder onto the PCB and test. My test plan was to use the output signal from the post-amp and read the voltage shift on the oscilloscope.

The voltage offset was initially coming directly from the ADC, so I soldered a wire connecting the two, and the output was offset like we had anticipated. However, it was not offset as much as we planned it to be. We tried changing the voltage divider on the adder to produce a higher offset voltage, but it didn't help much. Our final solution was to fix it with the code for the ADC.

Resistor Testing:

Once the entire circuit prior to the ADC was working, we moved on to testing different resistor values as our test/noise resistor to see how we needed to manipulate the post-amp gain. Our plan was to plug in different test resistor values and change the feedback resistors to the post-amps to produce the amplitude we desired. I thought this process would be simple, but it turned out to be very frustrating.

Any time a change was made, our output would jump to the negative rail. It seemed to do this randomly, but I found that it happened every time something was changed on the PCB specifically. After testing numerous ideas on what caused the problem, we found that it was from a floating pin on the ADA4622-2. Any time we would make a change, I would need to solder or desolder that floating pin to the positive supply voltage. Once I did this, the output would go back to normal.

We have no idea why this kept happening, but we found a decent solution to our problem. From what we could find, that floating pin on the ADA4622-2 was a disable-not pin, so theoretically soldering that pin to the V+ input of the ADA should keep it enabled. This just didn't seem to be the case for us, but we were still able to get around the issue in the end.

From there, we tested different input resistors and determined the gains we needed. Our goal was to get an output peak to peak of roughly 1.5V or so for each test resistor. We found that fixing the first feedback resistor at 1.762kohm allowed it to always stay the same while we were within our 1kohm to 100kohm test resistor range, which simplified our testing and overall circuit. **Figure 16** below shows an example of our final noise signal and **Figure 17** shows the different feedback resistors we used for each test resistor.

Figure 16

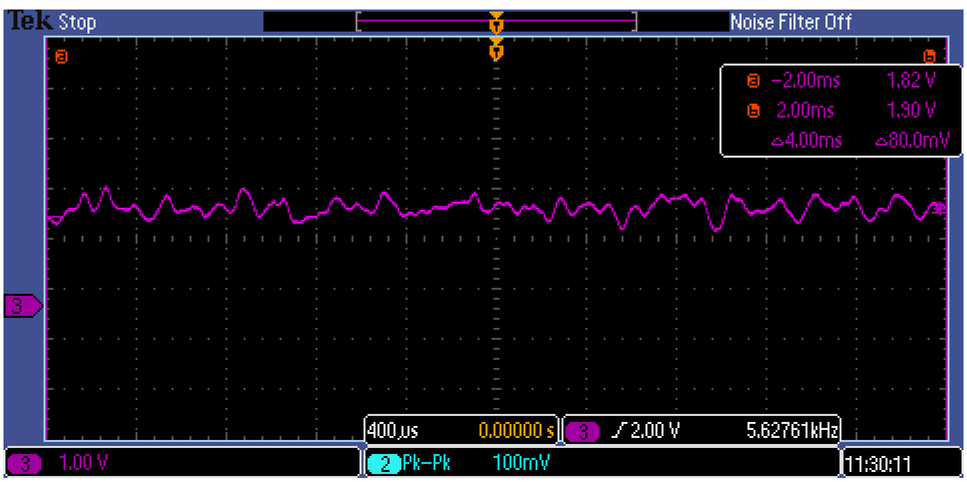


Figure 17

Test Resistor (ohm)	Post-amp Feedback Resistor (ohm)
97.24 k	117.6 ohm
80.6 k	177 ohm

67.01 k	380.7 ohm
55.29 k	545.9 ohm
46.65 k	675.6 ohm
38.62 k	811.65 ohm
32.700 k	1.176 k
26.575 k	1.464 k
21.825 k	1.464 k
17.64 k	1.749k
9.863 k	2.658 k
983 ohm	11.91 k

Later, while testing our system at guessing a mystery test resistor, we discovered how to determine the feedback resistor for the unknown test resistor. We would cycle through this same list of feedback resistor values above until our oscilloscope gave us a wave with enough amplification while also not hitting the op-amp’s rails. This would not only give us an idea of what the mystery resistor value is, but it is also pretty quick to swap around after you get the hang of it. [Bo]

Parts and Purchasing

Below is a parts list of all the parts we purchased from Mouser and JLCPCB for our project. All of these parts cost quite a bit, and they still weren't quite everything we needed. Luckily, the table below this one shows how many parts we were able to retrieve from the parts storage room in GAB for free. [Steven]

Part Name	Part Number	Quantity	Price (pp)	Total
ADC	ADS8322	2	$23.80	$47.60
Op-Amp	LT1007CN8#PBF	6	$4.80	$28.80
Op-Amp	LT6200CS8-10#PBF	4	$3.79	$15.16
Screw Terminal	282834-5	2	$3.04	$6.08
Screw Terminal	282834-2	2	$1.21	$2.42
Header Terminals	1-825433-0	4	$1.39	$5.56
10k Resistor	MFR-25FBF52-10K	7	$0.13	$0.91
100 Resistor	MFR-25FBF52-100R	4	$0.10	$0.40
200k Resistor	MFR-25FBF52-200K	4	$0.10	$0.40
68k Resistor	MFR-25FBF52-68K	4	$0.12	$0.48
3.6k Resistor	MFR-25FTE52-3K6	2	$0.12	$0.24
82k Resistor	MFR-25FTE52-82K	2	$0.12	$0.24
1k Resistor	MFR-25FTF52-1K	8	$0.12	$0.96
220k Resistor	HHV-25FR-52-220K	4	$0.45	$1.80
13k Resistor	MFR-25FTE52-13K	4	$0.12	$0.48
750 Resistor	MFR-25FTE52-750R	2	$0.11	$0.22
1.5k Resistor	MFR-25FBF52-1K5	2	$0.10	$0.20
360 Resistor	MFR-25FTE52-360R	2	$0.11	$0.22

Op-Amp	ADA4622-2ARZ	6	$5.96	$35.76
Oscillator	ECS-2200BX-100	2	$3.28	$6.56
Resistor Box	54-CBSA-3.5X5.5X0.5	1	$24.77	$24.77
Blank PCB	PR2H1	2	$3.15	$6.30
Adapter	PA0091	1	$7.39	$7.39
Total Price	192.95			

Part Name	Quantity
0.1uf Capacitor	5
10uf Capacitor	2
2.2nF Capacitor	7
0.015uF Capacitor	3
1000pF Capacitor	2
0.01uF Capacitor	2
3900pF Capacitor	2
4700pF Capacitor	3
220pF Capacitor	3
1500pF Capacitor	2
Shielded Cable	1ft

Final Design

Final Schematic

The following figure, **Figure 18**, shows our final multisim schematic design directly above our final PCB design, **Figure 19**, which is also directly above a photograph of our final circuit design and implementation, **Figure 20**. [Steven]

Figure 18

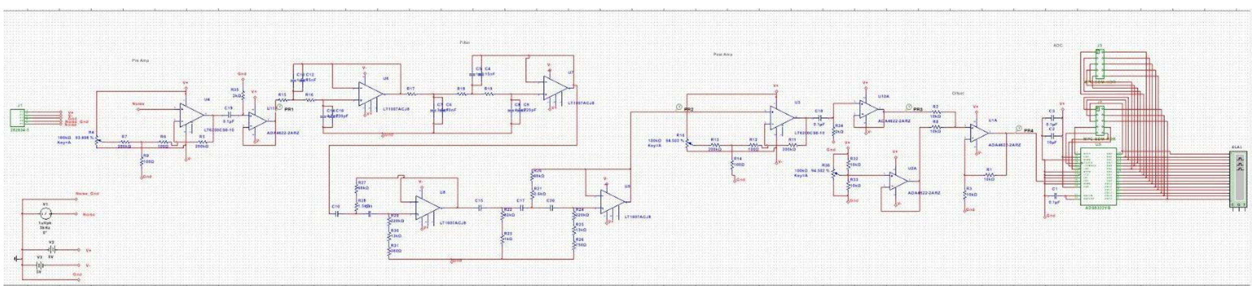


Figure 19

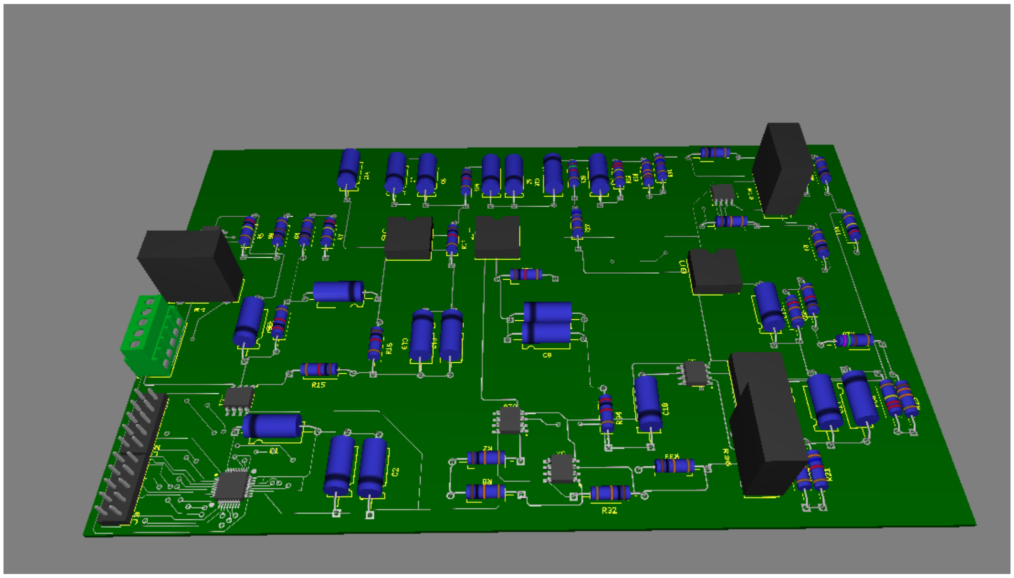


Figure 20

********INSERT PHOTO********

The final designs above show all of our components properly connected and joined together to properly amplify the thermal noise across the testing resistor in its protected box. In fact, this was our goal from the beginning, and it's just dawning on me now that we are finished with this project and this class.

Putting that aside, looking back up at **Figure 18** and the final schematic, you can see each and every one of the major parts of the project, except for the software perhaps. First, we have the preamp and power sources for our circuit on the far left. Next in the circuit is the large middle section which holders the entirety of the filter in two rows of op-amps. Then, we have the post-amp after that, though there are two post amps after that in our final design implementation, as shown in **Figure 20**. The post-amps then feed into a buffer op-amp and then the offset summing amplifier and the ADC on the far right, thus completing our circuit's entirety. The only other addition in our final project implementation in **Figure 20** is the inclusion of the test resistor in its shielded box attached by its shielded cable. [Steven]

Final Testing and Results

Testing

Overall Testing

For our final tests, we had four major stages in the process. Initially, shortly after we had gotten a full working circuit for the first time, we had a long and awful stage of troubleshooting to find why our Boltzmann's constant evaluations were so poor. Eventually, we found that our software had a major flaw in the way it was reading in our data. Though fixing this software issue was what fixed this issue, many other things were changed for the better within our software while trying to solve this issue. I would call our first major step of testing revolved around software.

Our next stage of testing could be thought of as more of a guess and check stage. Since we needed to determine what the feedback resistor on our post-amp would need to be for each test resistor, determining these combinations was our next step to our final results. We had just begun receiving more accurate Boltzmann's constant approximations, and testing these was the next step.

Next, we began measuring our calculated Boltzmann's constants for each predetermined combination of test and feedback resistors. For each combination we collected data and then processed that data for a Boltzmann constant 3 separate times. It quickly became apparent that our lower resistor values would give us significantly better Boltzmann constant approximations than the higher test resistors. In fact, graphing the percent error of our Boltzmann constant approximations shows us that they seemingly get worse and worse along a continuous linear path. However, all of this is talked about in more depth soon in this report.

Finally, our last major step to final tests on our device is practicing the process where we try and guess a random unknown test resistor's value. Since we weren't required to do a full response and understanding of this data, we only tested this process enough to know it would work properly when tested by Dr. Gard in the lab. [Steven]

How to Test Properly

When running our final circuit, there were a few quirks in the way it functioned. The first of these quirks was mentioned by Bo earlier with the ADA-2's floating "disable not" pin. For some reason, whenever we would change something on our PCB, like solder or unsolder something, our post-amp would pull the noise signal to the op-amp's negative rail. Then, we could solder or unsolder the floating pin on the ADA-2 and it would magically fix the post-amp and our noise signal. We still aren't quite sure why this would happen, but at least we were still able to have a properly functioning circuit when this was fixed.

Our second range of testing and quirks came when we realized how much we were going to need to change our post-amp feedback resistor. Not only did we have to test many different resistor values for feedback resistors on our post-amp for every test resistor value we were using, but we also had to make sure we were measuring their exact resistances and inputting them into our software whenever we used them. You see, each time we change a test resistor we have to also change our feedback resistor and both of those exact resistance values need to be input into our software for proper calculations.

Then came the odd quirks with our mystery resistor calculations. We were having issues with our original math for this approximation because of the way our Boltzmann constant approximations constantly increase proportionally with the test resistors and occasionally spike around. These inconsistent Boltzmann constants caused our values to have quite a few issues, so we decided to implement a second round of approximation in our resistor value testing software. This implementation was also mentioned earlier in the report by Micahel.

Finally, some of the weirdest results and quirks arose from the underside of our boards. It seemed as though whenever our boards were lifted from the workbench's surface, the noise signal would become much cleaner and more readable. We ended up rigging our perf board so that it somewhat floated above a plastic circle between it and the table. Then, the added feet to our main PCB and had it lifted above both the table and a box between. Both of these changes dramatically changed our signals, it was quite interesting. [Steven]

Results

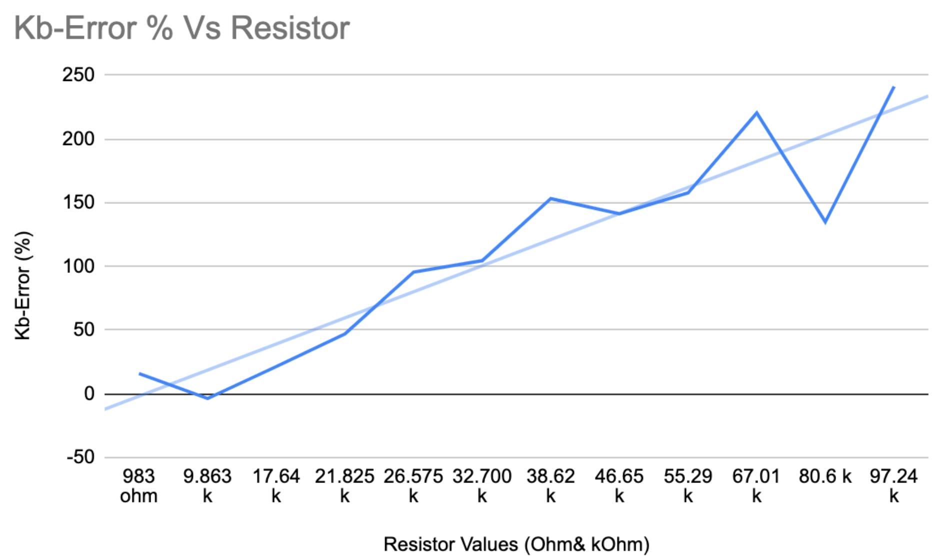
Testing Results

The following chart shows the tests resistors with their matching feedback resistors and the results of testing their data for Boltzman constants. That is a lot of information in one chart, so I have color-coded the resistors green, the test result Boltzman constants yellow, and the percent error of the Boltzman constant blue. [Steven]

R-Test	R-Feedback	Kb- T1	T1- % error	Kb- T2	T2- % error	Kb- T3	T3- % error	Avg % error
97.24 k	117.6 ohm	4.84045	250.757	4.72314	242.257	4.56422	230.741	241.25
80.6 k	177 ohm	3.24197	134.925	3.24993	135.50	3.23658	134.535	134.987
67.01 k	380.7 ohm	4.29784	211.438	4.57146	231.265	4.39896	218.765	220.489
55.29 k	545.9 ohm	3.4904	152.928	3.57	158.696	3.60984	161.58	157.735
46.65 k	675.6 ohm	3.41948	147.788	3.33685	141.801	3.23699	134.56	141.385
38.62 k	811.65 ohm	3.42041	147.856	3.44693	149.778	3.61705	162.105	153.246
32.700 k	1.176 k	2.75537	99.6645	2.82162	104.465	2.88934	109.37	104.501
26.575 k	1.464 k	2.67357	93.737	2.69801	95.508	2.72378	97.3754	95.5401
21.825 k	1.464 k	2.049079	48.484	2.024226	46.683	2.012641	45.8436	47.0035
17.64 k	1.749 k	1.6669	20.7899	1.65537	19.954	1.70434	23.5029	21.4157
9.863 k	2.658 k	1.318711	-4.441	1.321129	-4.266	1.35177	-2.0457	-3.5843
983 ohm	11.91 k	1.60983	16.654	1.554849	12.670	1.634398	18.4346	15.9197
							Total Avg % error	**110.824**

Then, the unexplainable linear relationship between the increase in test resistor and increase in our Boltzman's constant approximation, which I mentioned before, can be seen in **Figure 21** below.

Figure 21



Understanding Our Results

Perhaps the fact that we had to constantly change our post-amp's gain caused our Boltzman constant approximations to vary so greatly, but at least we were able to get constant readings from our system.

When looking at any singular resistor and the thermal noise produced and read in by our device, we almost never varied more than 2-6% from any other test of the same resistor combination. Our circuit wasn't the most accurate, but it was consistent and reliable. If you put in a 10k test resistor with its proper feedback resistor, our circuit would almost always give you a Boltzmann constant approximation within 10%. In the end, our best Boltzmann constant approximations came from our 10k test resistor and could get within 2% and once even within 0.5%. However, because of our awful approximations with larger test resistors, our average percent error when approximating Boltzmann's constant was around 110%. This is not great, but like I said before, our system makes up for some of its inaccuracy with its consistency.

Further, our resistor estimations were similarly plagued and gifted. Because of the way we ran through our resistor estimation program twice, as explained before, our system tended to be much more accurate when guessing resistors at either end of our 1k - 100k range and less accurate in the middle. These inconsistencies between different test resistor values put a bit of a dent in our accuracy, but our consistency when testing the same resistor was just as good as when testing for Boltzman constants. In the end, when it comes to guessing our test resistor value, the positives outweigh the negatives. Our resistor estimator was almost never more than 50% off of the test resistor's value, and some of our best tests on the edges of our test resistor range could get within 15%. [Steven]

Planning and Scheduling

After our first meeting, we planned what each person was to look into before our next meeting. Bo was supposed to look into the filter, Jackson into the PCB, Michael into the software options, and Steven into the shielded wire and casing research. After another meeting, we assigned the ADC selection to Michael, the pre and post-amps to Jackson, and the offset adder to Steven. These roles ended up holding throughout most of the project. Though, eventually Michael also helped with the Pre and Post-amps, Bo took over circuit building and testing, and Jackson had to quarantine for the end of our project due to covid. Luckily, most of

Jackson's work came from being the POC and early design work, so we were able to get the building and testing of our project done smoothly with just minimal zoom interactions with Jackson.

In the end, we still had to pull two weeks of constant lab time testing and troubleshooting, but it seemed as though we were one of the earlier groups who was able to wrap up comfortably before presentation day. I can still remember the feeling of being on cloud nine when we left lab that day confident in the effectiveness of our circuit. [Steven]

Budget Summary

By the end of our project we had used almost our entire $250 budget. Our Mouser part sheet earlier combined to a total of almost $200, and those purchases combined with the cost of the manufactured PCB boards from JLCPCB added up to nearly $250. We could have saved quite a bit without purchasing two separate surface mount ADCs that couldn't be used or an oscillator that wasn't needed for our final ADC. However, we didn't go over budget, and we used supply room parts quite a bit to try and stay away from over-purchasing. Furthermore, I was very satisfied with the things we initially decided to spend larger amounts of money on including the PCB, ADC, and op-amps because of how important their functions were. We unfortunately had a few issues where we couldn't use all of those, but I would choose top tier components for each of those if tasked to do this project again. [Steven]

Project Reflections

Michael's Reflection

If I had this project to do over again I would first make all the op amp footprints DIP. This allows the use of converters for smaller size op amps while also allowing for DIP op amps. We had to move our post and preamp to a perf board because they were originally surface mount. This perf board was very susceptible to distortion. When we put our hand close to it, the output wave would alter. Getting rid of this perf board and better designing our PCB could fix this issue. I would also try and make our PCB board smaller board so that the whole board could fit into its own individual shielded box. Another thing would be to ask JLCPCB to attach our ADC themselves. I learned after we received our board that that was an option, and this could have allowed us to use our initial ADC.

For code I would have used C++ to read the serial port instead of using PuTTy. This would allow people to use the code without having PuTTy installed and would remove the need for entering the path of PuTTy on every new computer.

Overall the project went very well and there were only small things that would make it better. [Michael]

Bo's Reflection

If I got to do this project again from the beginning, I would not use op amps that have a floating pin anywhere in the design. Those caused us trouble almost every day we were in the lab. There was no way we could've seen it coming, though. If those op amps would have worked like we planned, I think our circuit would have been more accurate because we wouldn't have used a perf board for the pre amp and there would have been a lot less jumpers across the circuit. Aside from some fixable mistakes, I think we did a great job of ordering extra parts because we ended up using several of them. We also had a solid test plan which made implementation go as smoothly as it could have. Overall, I am pleased with how well our circuit performed. [Bo]

Jackson's Reflection

If we were to do this again I would consider ordering smaller boards for each section of the circuit (per-amp, post-amp, filter, ect.). This would be useful if some part of our circuit failed and had to be redone. We would have also been able to order the boards sooner and find any mistakes or flaws in our design by testing them individually. Overall, despite some setbacks, I believe our project performed fairly well. [Jackson]

Steven's Reflection

My reflection is a bit more influenced by everyone else's since I am writing it after reading and writing out everyone else's reflections just now, but I think I still had a few things that my partners didn't mention. However, I do agree heavily with Bo's opinion on using op-amps with no floating pins if we reworked this project.

I love the way we prepared for the worst in our initial designs. We designed our pcb so that we could include or exclude different parts of our circuit so that we could optimize it as we tested it initially. However, we still could have planned better. We were able to overcome a ton of issues because of how early we got into debugging our circuit, but if we had designed each part of our circuit more separated, like Jackson mentioned, it would have been easier to troubleshoot.

I also hated the shielding box I ended up choosing and the way we had to set up our circuit when testing. I chose very poorly on a shielding box for our test resistor and should have gotten something more robust. I also wish we had designed our pcb board with rubber feet initially so the bottom of the board never touched anything and couldn't cause issues.

Aside from these issues from our project and the few positives I mentioned, I would put more time and effort into being a better teammate next time around. I could see myself being a dead weight early in the project, and I'm glad I stepped it up at the end with our testing, presentation, and report. However, I wish I could have been that kind of a teammate throughout the project. [Steven]

*Note: Much like in my last project report, sections that were worked on by my project partners are labeled and any passages or sections which one of them wrote has their name in [brackets] at the end of said section. Anything that does not have a distinction on who wrote it or a section header with someone's name was written by me, Steven Gaiko, the main report author.

References

ADCs

https://www.ti.com/data-converters/adc-circuit/products.html#p157max=1;50&p1918=Parallel&p89=Pipeline;Flash;SAR&sort=p157max;desc

Chosen ADC

https://www.mouser.com/ProductDetail/Texas-Instruments/ADS8322YB-250?qs=%2Fha2pyFaduhv7HiYQOtyJgGLbGZh8N1gczkzaPOgRscVYHlzYj6eYg%3D%3D

Chosen filter amp

https://www.mouser.com/ProductDetail/Analog-Devices/LT1007CN8PBF?qs=ytflclh7QUWpHOiIqTItEA%3D%3D

Null Offset

https://electronics.stackexchange.com/questions/34071/how-do-i-correct-the-offset-voltage-of-op-amps-which-have-no-explicit-offset-nul

Summing Amplifier (ADDER)

https://www.electronics-tutorials.ws/opamp/opamp_4.html