Johnson Le

January 23, 2019

CMPE 167/L

Lab 1 Report

**Introduction**

This lab introduces two new sensors: flex sensor and vibration sensor. The flex sensor works like a potentiometer by changing resistance based on bending angle. The vibration sensor detects vibrations and outputs higher voltages the stronger the vibration. The goal of this lab is to incorporate these sensors into one circuit to control the speaker used from last lab.

**Part 1 – Linearize the Flex Sensor**

Method:

The goal here is to linearize the flex sensor to get consistent readings based on the degree of flex. The first step I took was to get readings from the flex sensor onto an A/D pin.

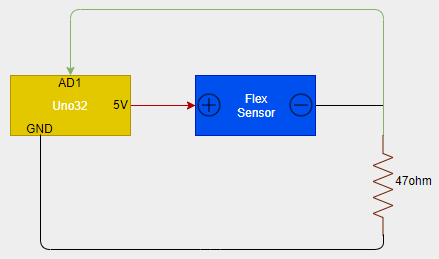


Fig 1 Flex Sensor Wiring

I notice the values ranged to a maximum of 1023. Printing the values onto the OLED, I could easily note down A/D values onto excel. I printed out a protractor to measure the degree of flex and noted degree of flex vs A/D reading. The results are shown in figure 2.

Fig 2 Flex Sensor: Degree Vs A/D Reading

Utilizing excel, I added a best fit line from all the data and got an equation. The equation can be seen in figure 2. The y denotes A/D reading and x denotes the degree of flex. Because the uno32 can’t read the degree of flex as an input, I solved for x. This gives me an equation I can code up as a function allowing the A/D readings from the flex sensor to be converted to degrees providing me with linearization of the flex sensor.

**Part2 – Capture the Taps**

Method:

Moving from the flex sensor, part 2 has us explore the vibration sensor. The vibration sensor generates voltage based on the amount of vibrations it goes through. Before sending any signal from the vibration sensor into the uno32, I had to make sure that the signal generated was within working range for the uno32. I hooked up the vibration sensor in parallel with a resistor and diode. I then hooked up an oscilloscope for the output which is the positive terminal. Striking the sensor, I get peak to peak values ranging from 1.4V to 2.8V averaging 2V depending on how much force I put into striking. This is a safe voltage range to input into the uno32. I tested this by continually striking the vibration sensor with some trigger delay from the oscilloscope.



Fig 3Average Striking Voltage

Now I hooked connected the vibration sensor to the uno32 for some coding. The wiring is shown in Figure 4. I outputted the AD value of the vibration sensor onto the OLED to see what values would be appropriate for considering it high. Once that was determined, I made a while loop with a condition that an LED would light up briefly if the vibration sensor was hit.

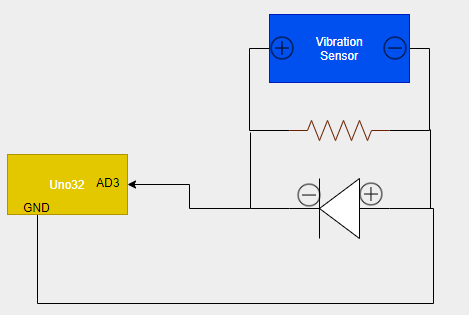




Fig 4Wiring of Vibration Sensor

**Part3 – Tone Out Speaker Based On Flex**

Method:

Now that I have linearize the flex sensors into degrees from part 1, I can set the tone of the speaker based on how much degree of flex. I decided to only have the sensor flex from 10 to 100 degrees. Anything above 100 would have the same tone as the one at 100 degrees. It then mapped that range to the range I wanted the speaker to have which is 100-880. This range was decided in the previous lab. 100-880 is gives nice, clear sounds from the speaker. I incorporated the wiring from Lab0 into my circuit as seen in Figure 5.

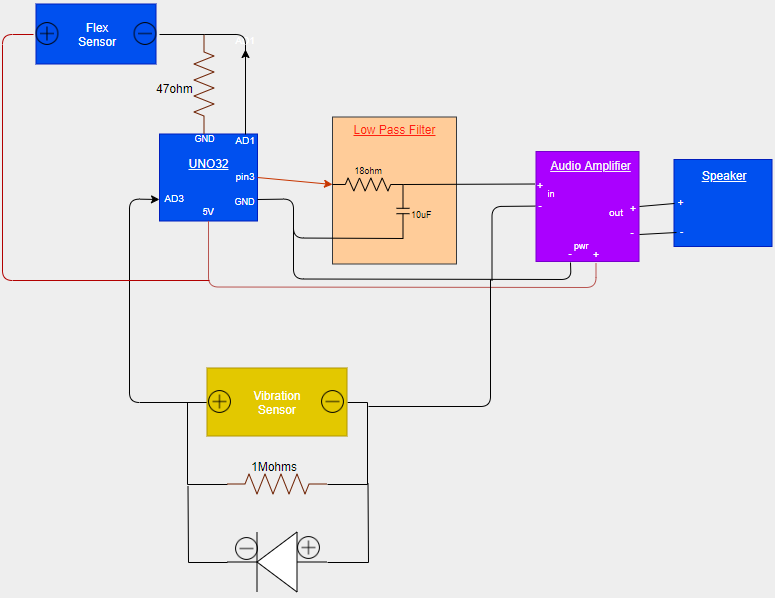


Fig 5Wiring for Everything

Once mapping the range, I used the tone generating library to set the tone based on the degree of flex.

Questions:

Is it noisy? Is it smooth?:

The flex sensor readings are noisy and bounces around rapidly without any filtering. The sound is also not very smooth and the properties of “crackly.”

Is your linearization of the flex sensor correct?:

The flex sensor provides the uno32 with AD values. I take those AD values and throw it into a linear function to get an arbitrary degree of flex. Because I throw it into this function, the flex sensor is linearized.

How could you improve the sound?:

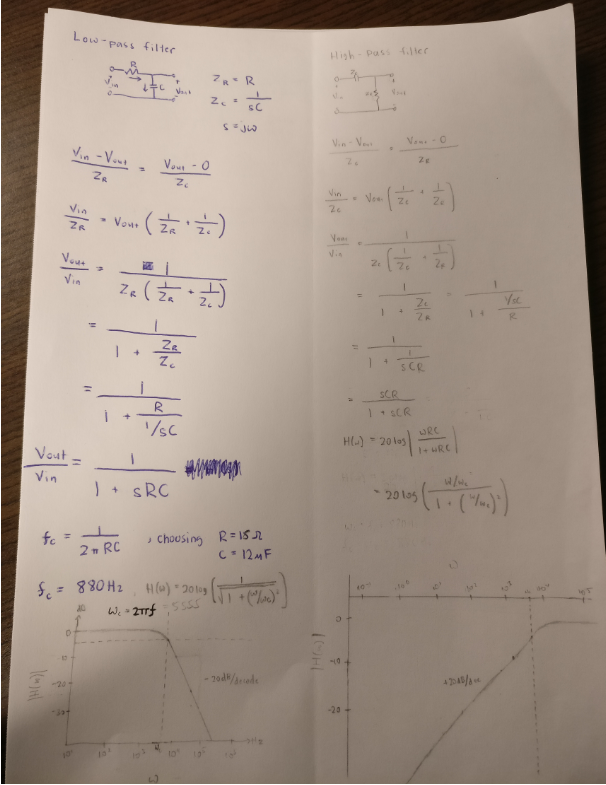
I improved the sound by adding a simple, passive low pass filter at 880hz. Although this decreases the volume, it reduces some noise allowing for smoother sound transitions. I also only allowed tones between 100 and 880 because tones outside of that range gets funky.

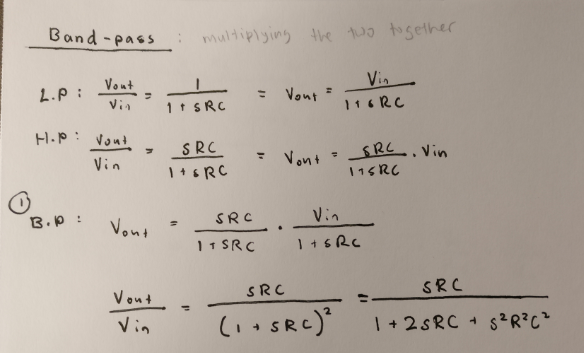
**Part4 – Tone Out Based on Tap**

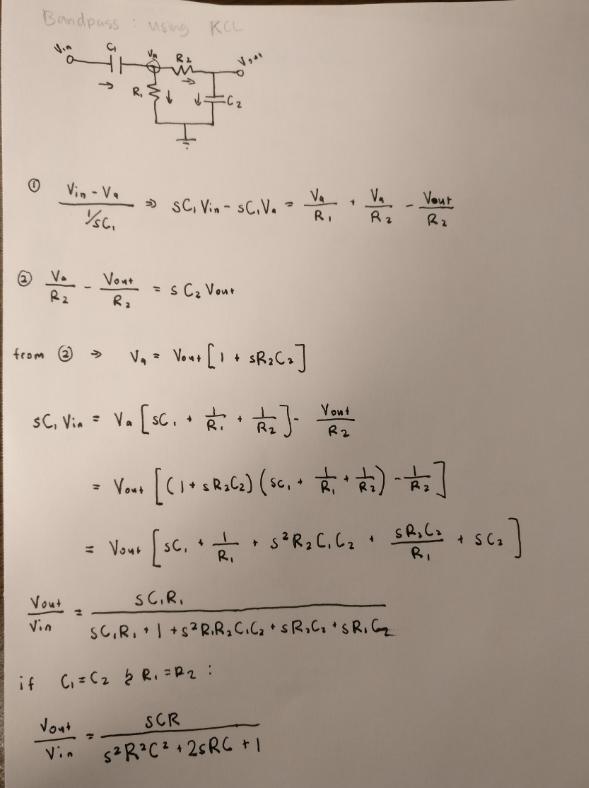
Method:

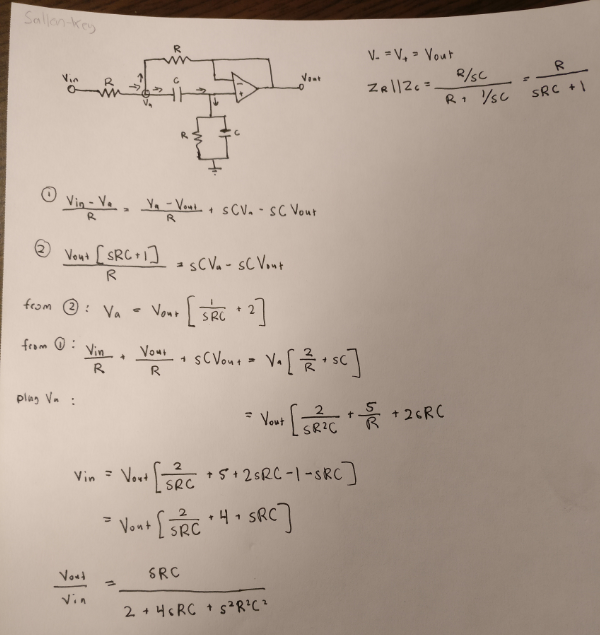
Now that there is code for interfacing the flex and vibration sensor, we can combine the two. In code, I start the speaker with no sound and constantly check for a vibration detection. If a vibration is detected, I will read the desired tone, set and turn on the speaker for a moment. There is no change in hardware and part 4 is all software.

**Part5 – Simple Analog Filtering**









**Part6 – Experimental Validation of Analog Filtering**

Method:

Part

**Conclusion**

T