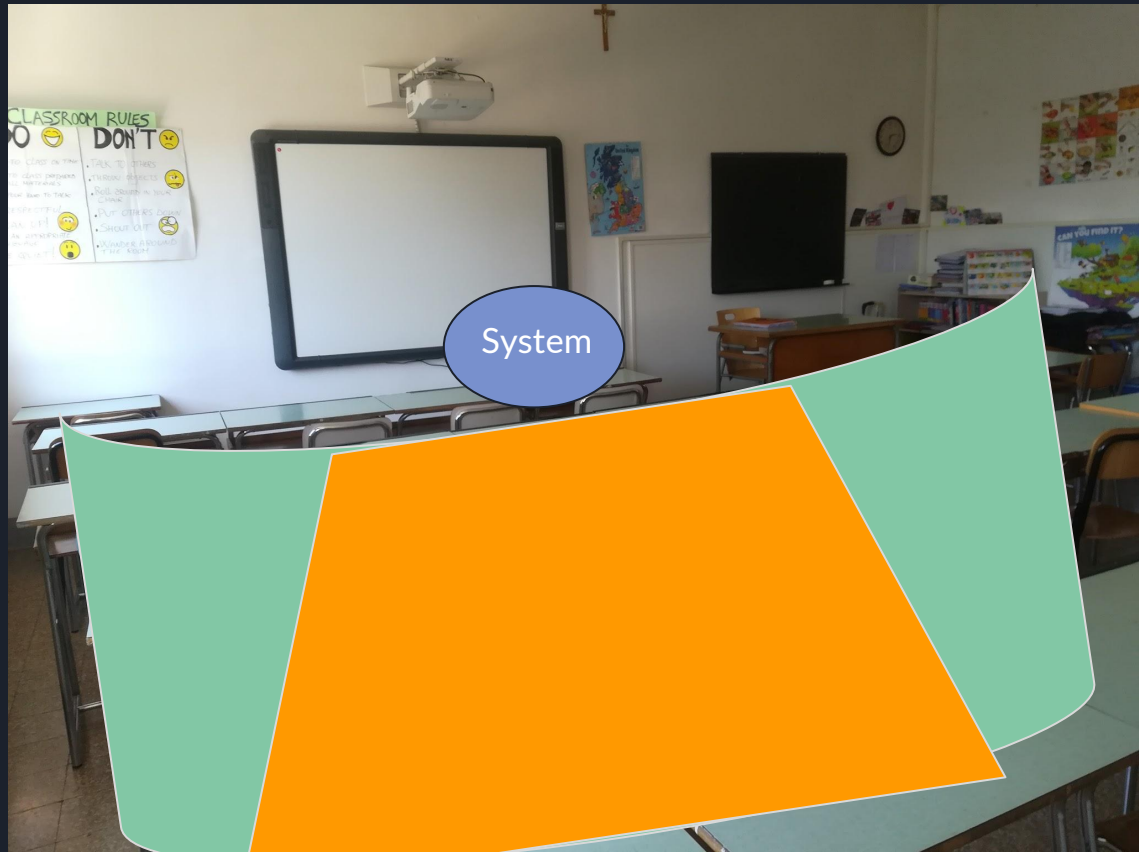
Noise Cancellation

Jess and Zach

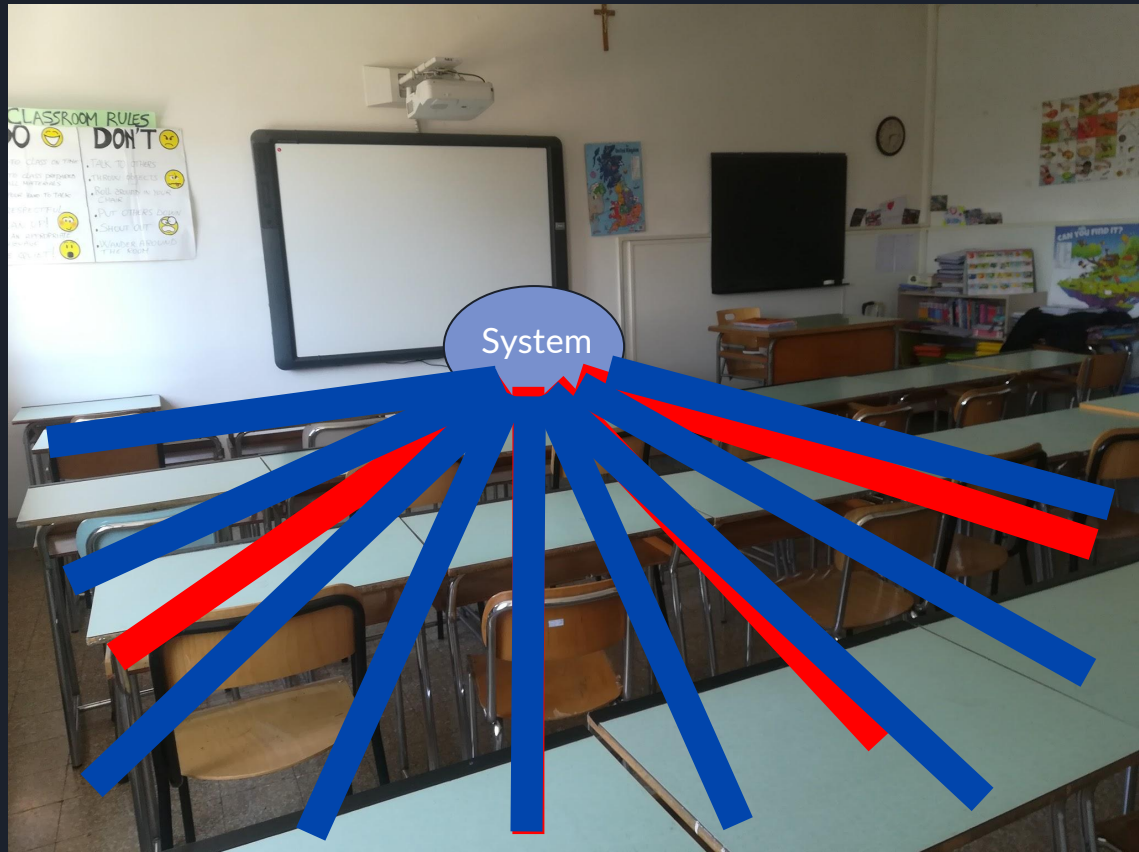
Goal

Our goal is to be able to record a sound wave, manipulate it, and then play out a shifted sound wave that interacts destructively with the sound in the room



Beginning Phase

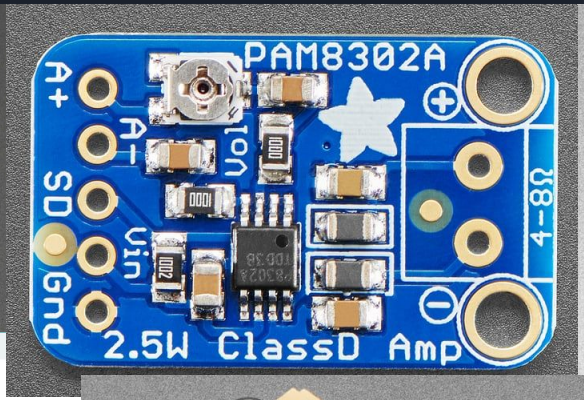
Use two speakers to determine how the sound interacts in different locations and different frequencies

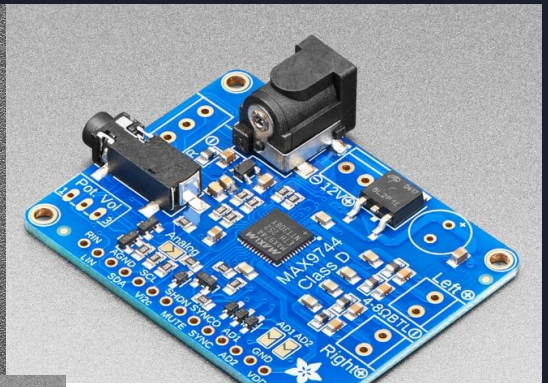


Overview of What was Done This Semester

- Tried out multiple amplifiers
  - Found out issues with many of them
- Wrote multiple codes that did various things to understand the hardware
- Wrote code to take measurements from the microphone and write them to an SD card
- Created a new set-up
- Took many measurements to find where two speakers will cancel sound

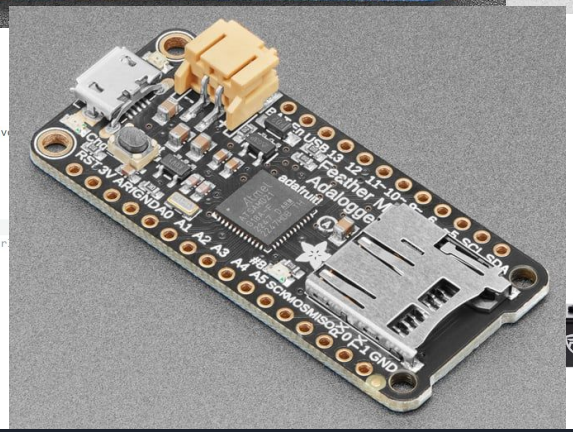
Things that didn't work

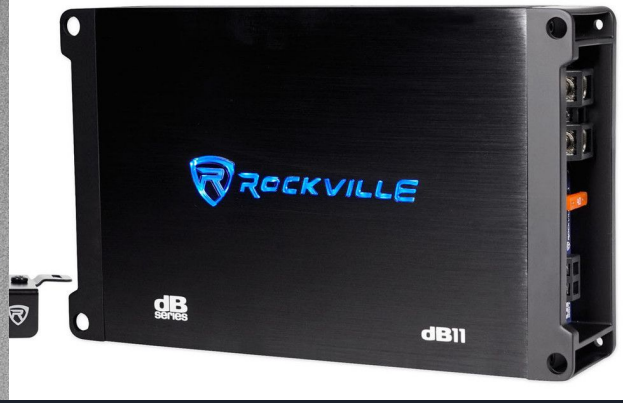




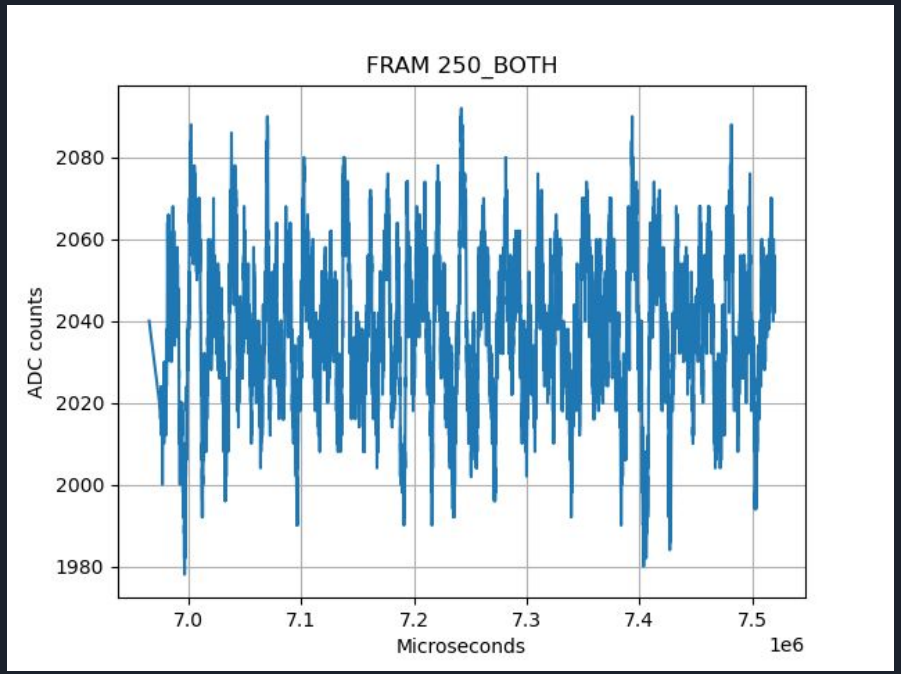
RingBuffer_code.ino

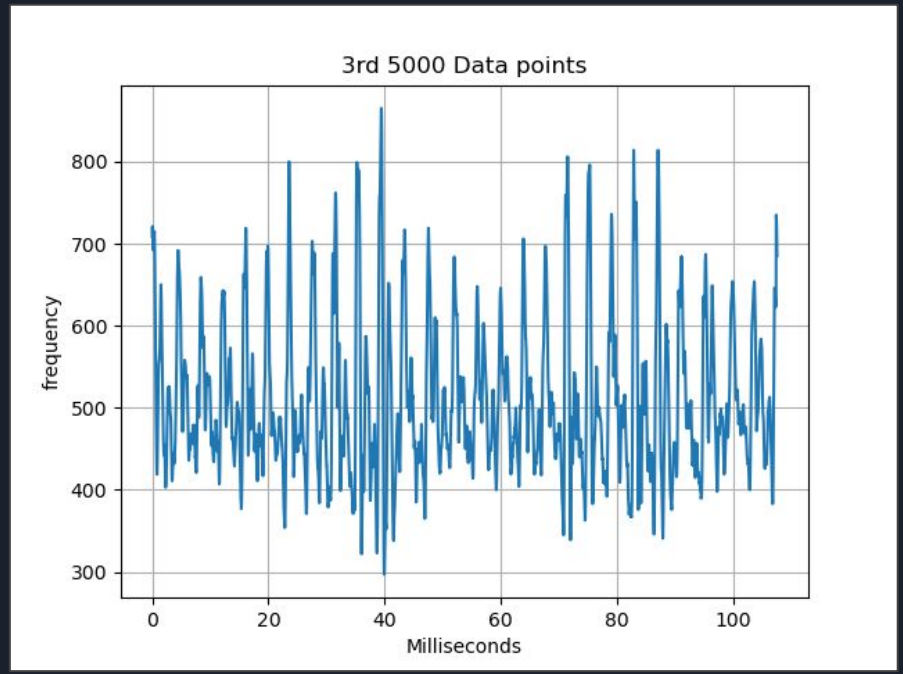
```
27  void loop() {
28    // put your main code here, to run repeatedly:
29    //Serial.println(microindex % bufferSize);
30    //Fixes the offset by converting to voltage, subtracting the offset, then converting back to adc.
31    //Multiplys by ten to make the signal larger
32    uint potValue = analogRead(potPin);
33    multiplier = potValue / vin * 10;
34    uint microphoneValue = analogRead(sensorPin);
35    uint microoffsetfix = (((((microphoneValue * vin) * inversebit) - halfvin + amplifieroffset) * bit) * inv
36
37    //Serial.println(microphoneValue);
38    //Serial.println(microofffsetfix);
39    //Serial.println("Math done");
40    //Put the microphone output value into the appropriate buffer index
41    buffer[microindex % bufferSize] = microoffsetfix;
42    //If the microphone index is larger than the offset play the microphone output value stored an offset ear
43    if (microindex >= offset) {
44      // Serial.print(microphoneValue);
45      // Serial.print(", ");
46      // Serial.print(microoffsetfix);
47      // Serial.print(", ");
48      // Serial.println(multiplier * buffer[(microindex - offset) % bufferSize]);
49      analogWrite(speakerPin, (uint32_t)(multiplier * buffer[(microindex - offset) % bufferSize]));
50      //Serial.println("Sent to speaker");
51    }
52    microindex++;
53  }
```



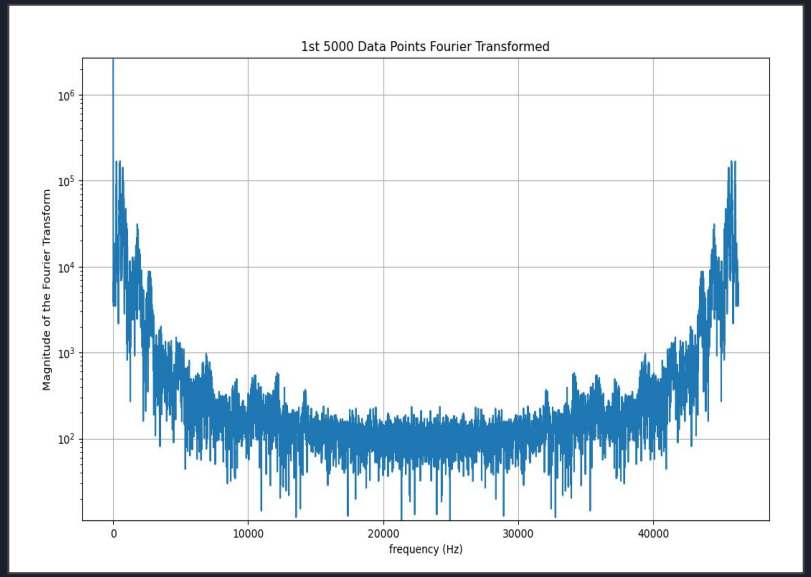


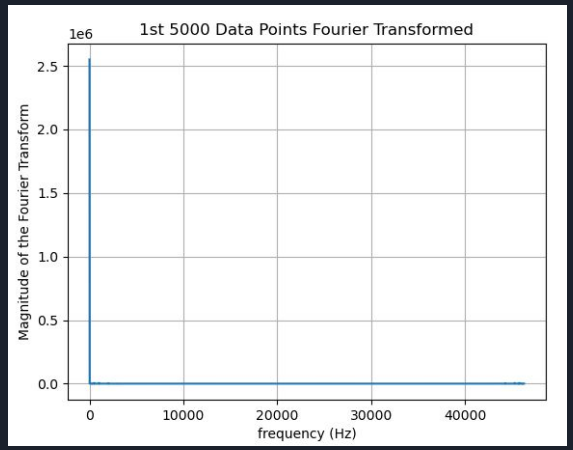
Old Signal Graphs

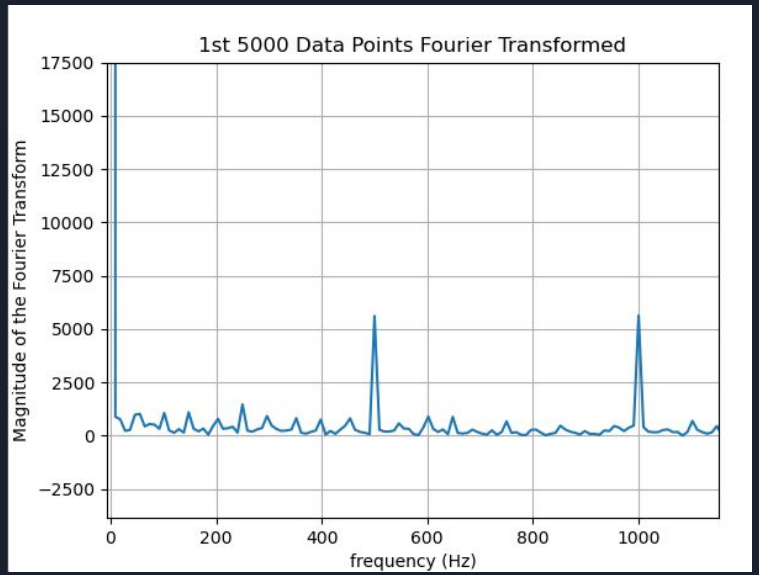




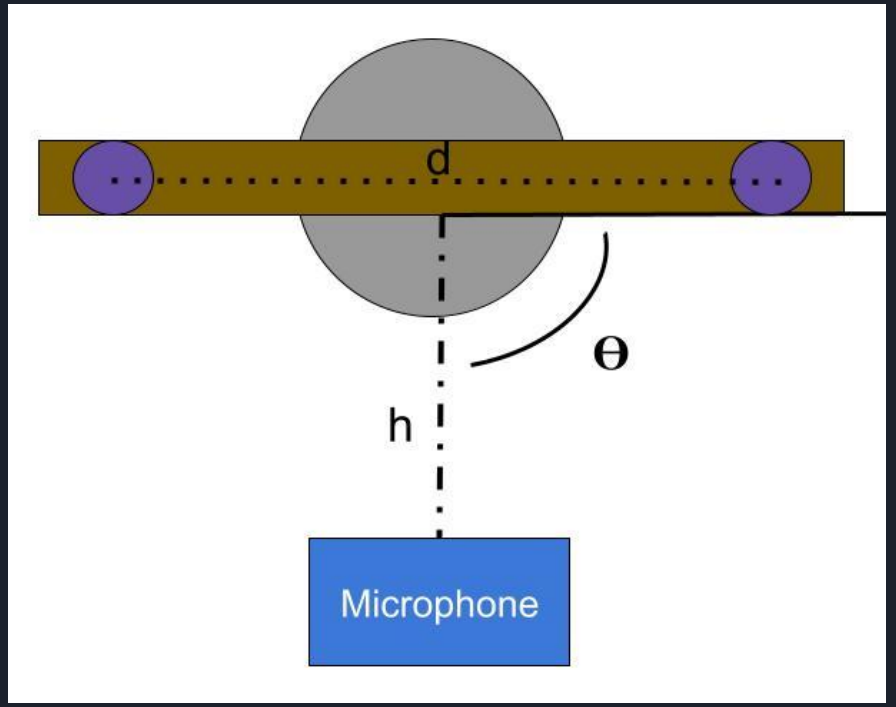
Fourier Transforms

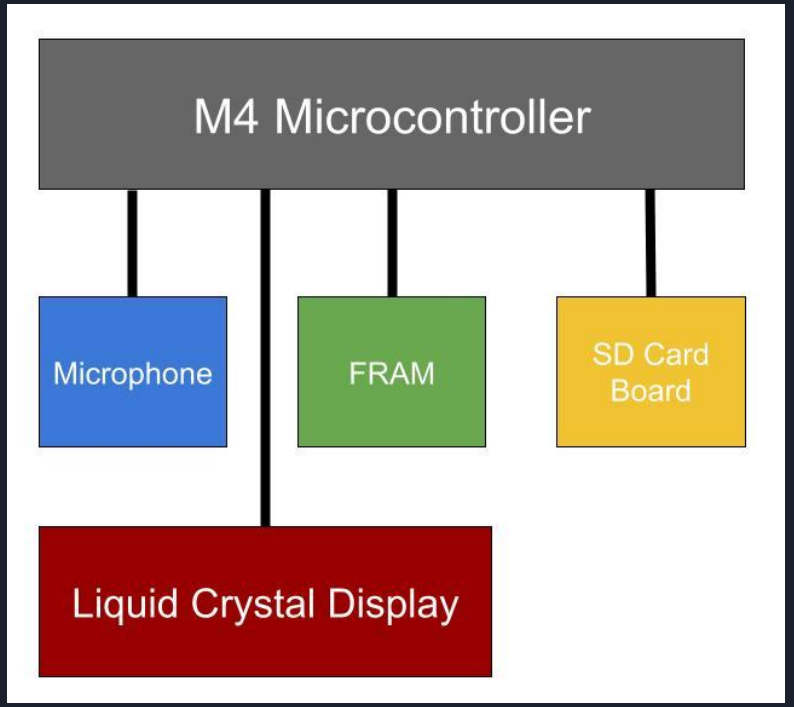




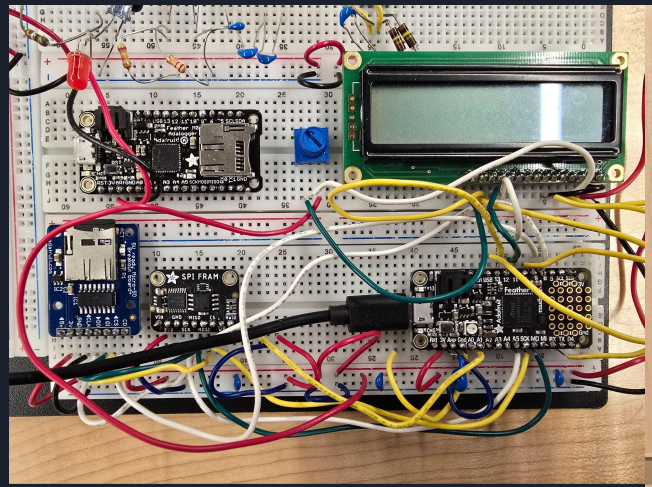


New Setup

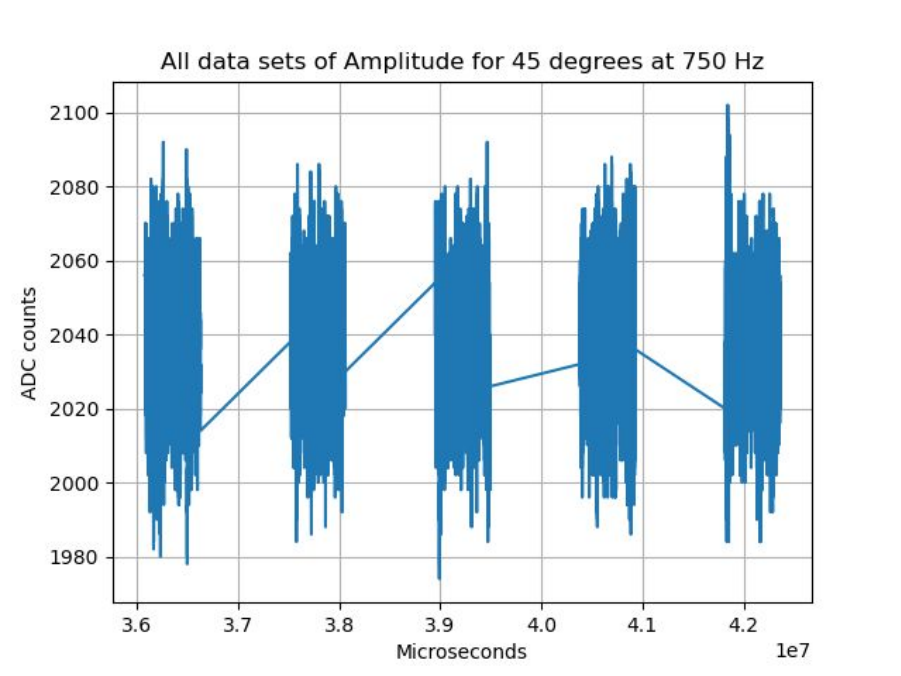

d
θ
h
Microphone

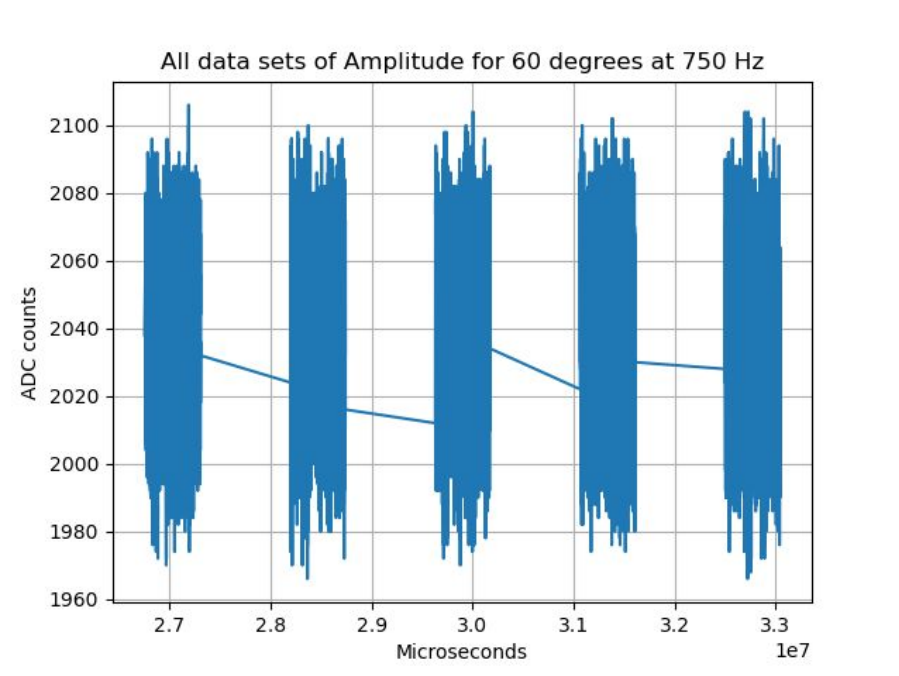

M4 Microcontroller
Microphone
FRAM
SD Card Board
Liquid Crystal Display

New Set Up Continued

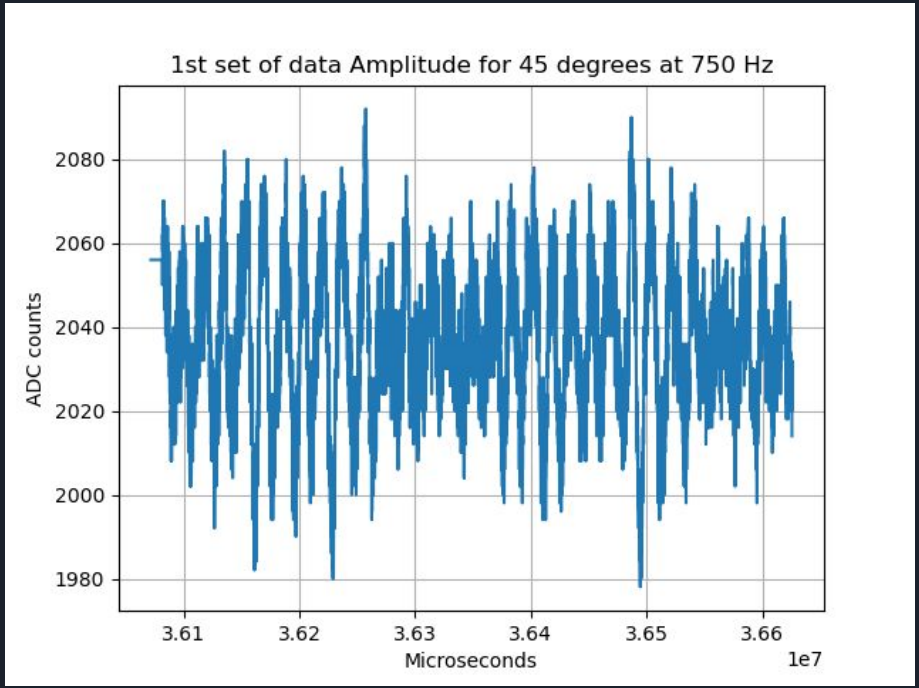


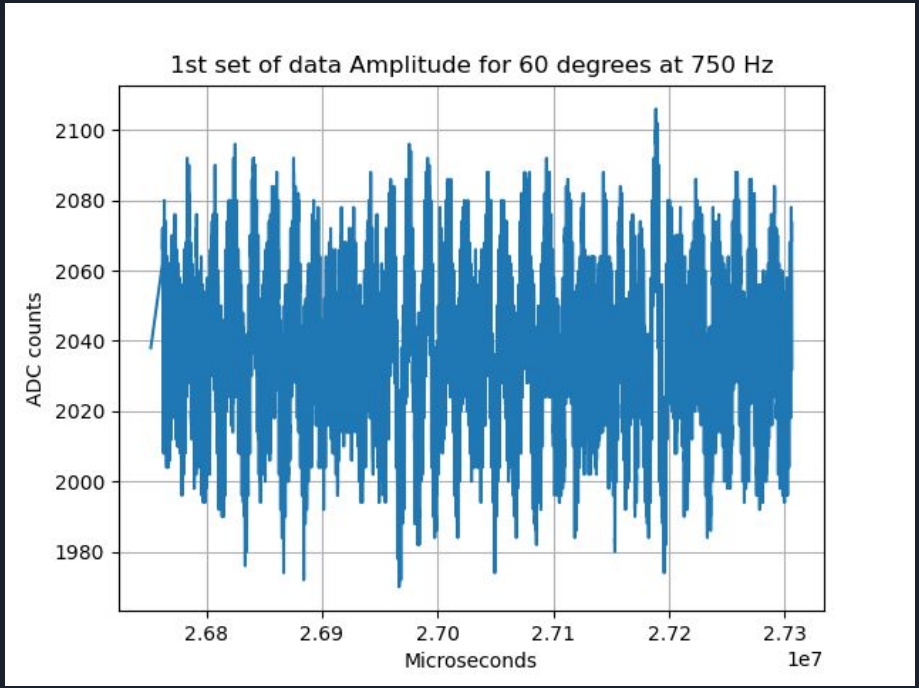
New Signal Graphs



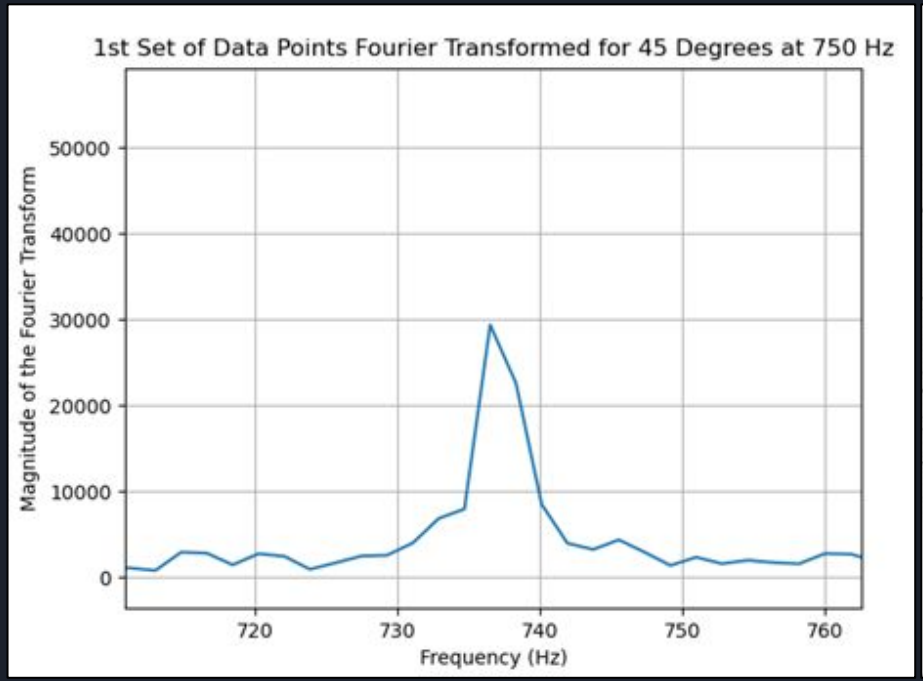


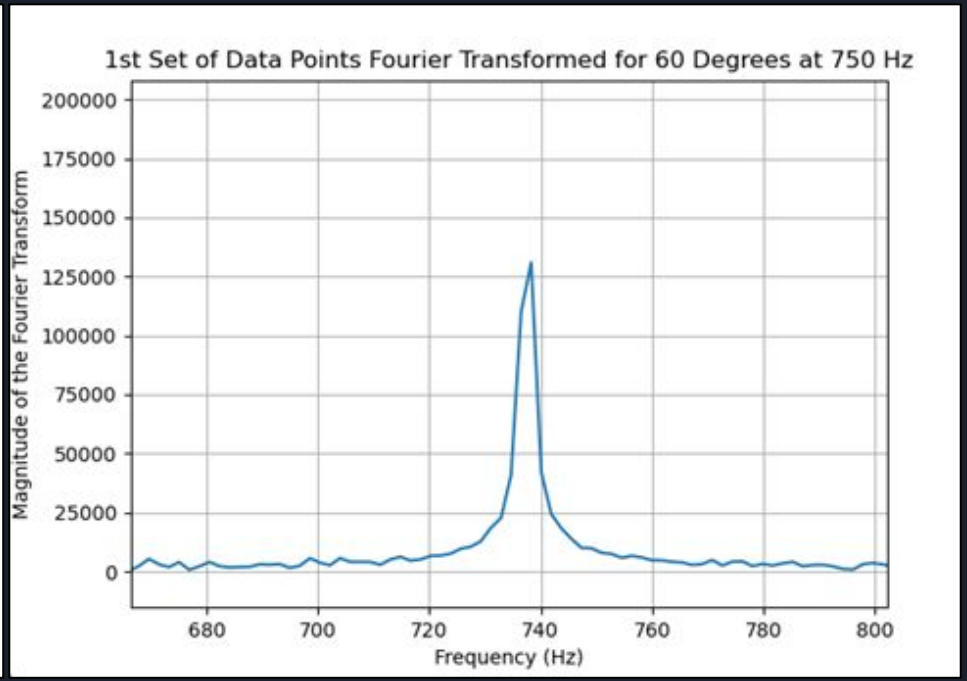
New Signal Graphs Zoomed in





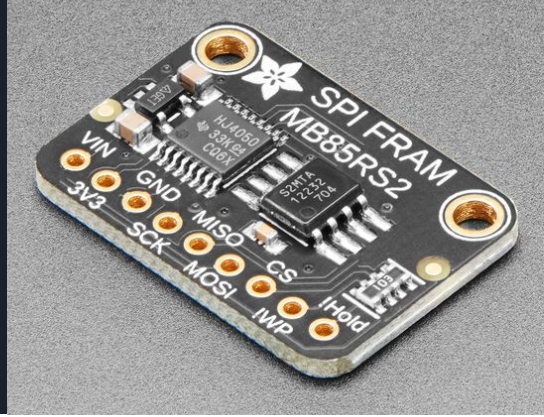
New Fourier Transforms

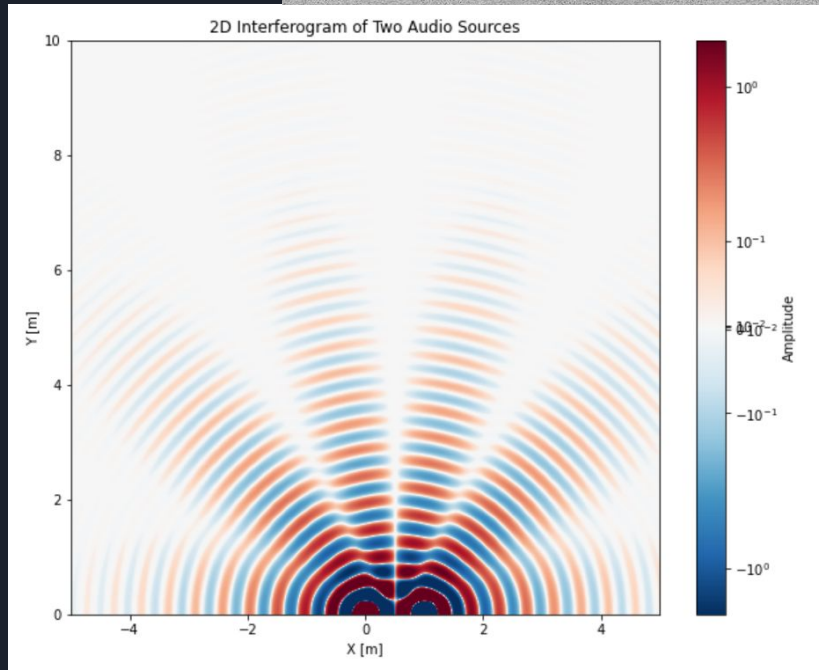




What we Learned

- We need a linear amplifier
- For the distance measurements to be consistent, multiple apparatuses were created
- We learned how to correctly calculate where the the waves should cancel
- Taking the measurements can be difficult due to the small margin of error in close range
- How to write the code needed for a fourier transform and how to write code that talks to all the hardware we use
- The memory on the M0 Adalogger wasn't enough so we moved onto the M4 with a FRAM memory





Plan for Next Semester

- Create a single speaker prototype that will cancel sound in a room
- Map out where the system does not cancel the noise
- Potentially add more speakers to cancel sound in more areas

Questions?