Jeff Kedda
Saksham Goyal

Milestone 2 Report

Integrated Design Project (Audio Direction Finding)

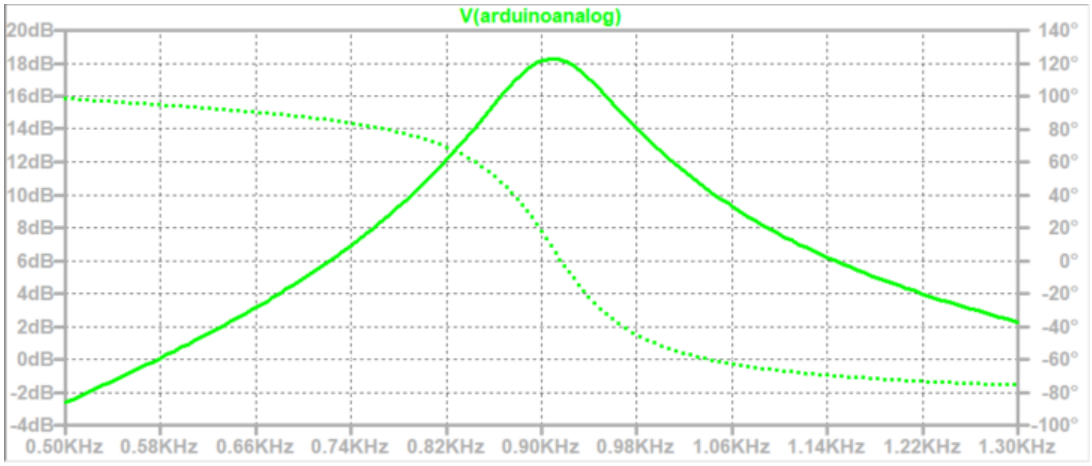
Jeff Kedda
Saksham Goyal

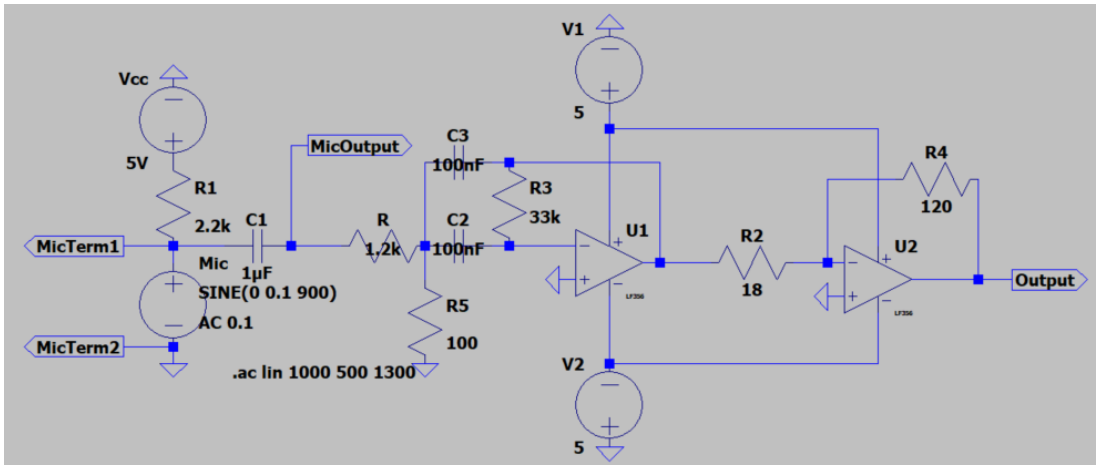
Objective

Our objective this week was to work on our project so that we can build and test the circuit that we built. In the project proposal, we defined our milestone 2 to be the point where we complete the circuit for the microphones, test the filter to make sure it works and is readable by the arduino, and then create the program so that the data can be read and displayed on the display. So, the first part of the milestone that we wanted to tackle was the circuit which included the 900 Hz Band Pass Filter and the amplifier so the signal is high enough for the arduino to read. The other part of the milestone that we wanted to tackle was the functionality of the LED display.

Results and Analysis

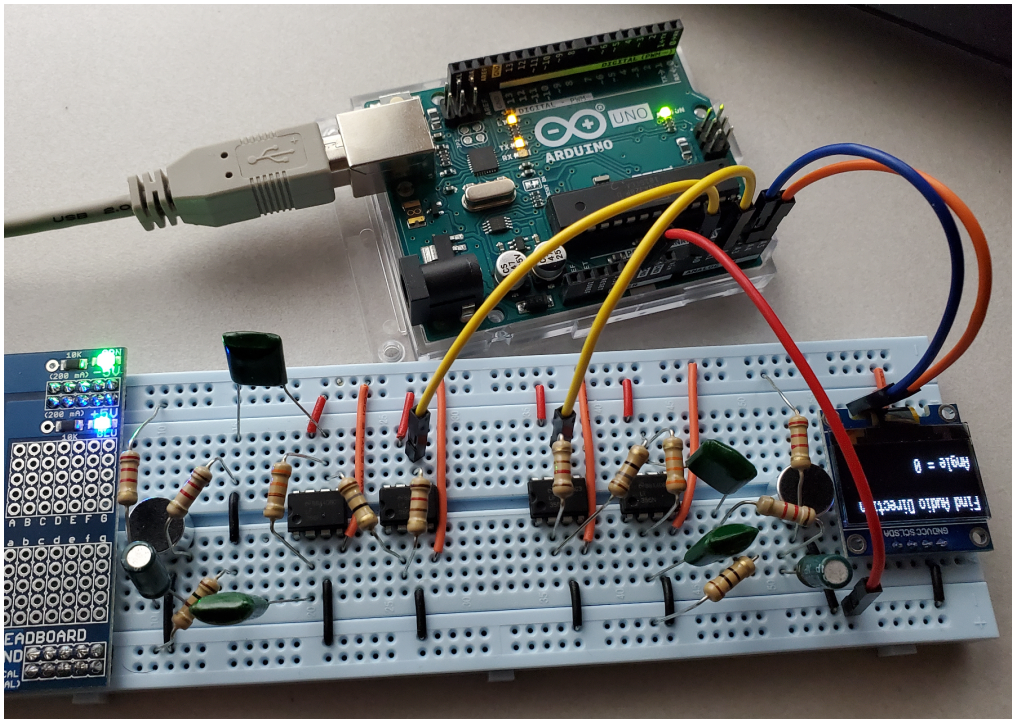
Amplifier and Band Pass filter

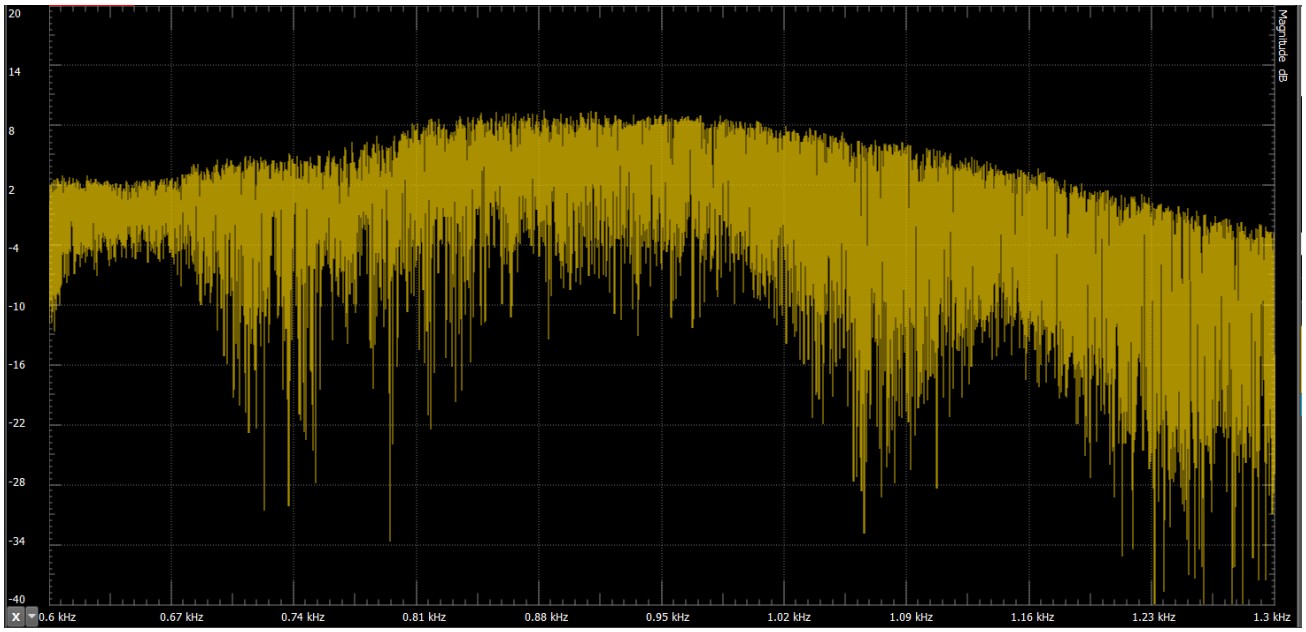




From the Project proposal, the LTspice simulation was modified slightly so that we could be more precise with the circuit. The older model used an ideal op-amp. Obviously since our op-amps are nowhere near ideal, I found a LTspice version of the LF365N op-amp to include so that the simulation is better.

Jeff Kedda
Saksham Goyal





After building the circuit. I wanted to test the actual bode plot output to check how close I got compared to the simulation. There seems to be some issues with an abnormal amount of noise in this graph which I have yet to diagnose, however a general trend can be seen which seems to emulate the LTspice simulation to a large degree. I tested the circuit using my phone and a frequency generator app. Currently, there is no digital filter implemented in code and the filter works to a reasonable degree. The arduino picks up frequencies between 700-1200 Hz. Using a digital filter to limit this to 800-1000 should not be too difficult since it is already so close.

Jeff Kedda
Saksham Goyal

LED Display

Initial Thoughts

One goal for this week was to test and understand the led display so that we can display input values on the display later in the project. It was also one of the few components that we were given. As a result, we should understand this component in the early stages of the project rather than later.

Process

First, we researched different websites and videos to find out how to connect and code the specific led display given. We found that because there already exists a library for the display in arduino, the code was very simple to implement after installing 2 libraries to handle the adafruit display.
In the code, we first had to figure out the I2C address since it is different for each display. In order to find out the address, we had to use the Arduino I2C Address locator program which we found online. Then we had to include 2 libraries (Adafruit_GFX, Adafruit_SSD1306). From there, we tested a simple hello world onto the screen of the device.

We then later implemented the code to display a message for the angle. In addition, the wiring diagram for the I2C circuit is shown. The code below is unfinished since we have not implemented the digital filter to limit the frequency to a hard limit of 800-1000, nor was the code to calculate the phase angle, nor the code for calculating the angle of the sound. Those parts will be implemented in the following milestones.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 4
#define SCREEN_ADDRESS 0x3C
Adafruit_SSD1306 display(OLED_RESET);

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  display.println("Find Audio Direction");
  display.display();
```

```
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 20);
  display.println("Angle = ");
  display.display();
}

void loop() {
  display.setCursor(50, 20);
  display.setTextColor(WHITE, BLACK);
  // Change this to print the actual angle after we figure it out.
  // I just put this to test the display capabilities.
  display.println(analogRead(A0));
  display.display();
}
```

Plan for Next Week

- Create code that extracts the information from the microphones and calculate the correct sampling rate
- Create and test code that finds the phase shift between the microphone signals
- Compare measured results from expected/calculated results

Jeff Kedda
Saksham Goyal