Saksham Goyal
Jeff Kedda

Milestone 4 Report

Integrated Design Project (Audio Direction Finding)

Saksham Goyal
Jeff Kedda

Objective

Our objective this week was to refine our code and accurately calculate the angle from the speaker to the 2 microphones. We have completed building the circuit however, the code to perform the calculations to display the angle was not done yet which was the objective of this milestone. We had to calculate the phase difference and use that to find the angle of the sound.

Results and Analysis

Test Code Part 1

Code that Shows the output below from the two microphones with the speaker playing at 900 hz frequency (Jeff Kedda)

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
#define SCREEN_ADDRESS 0x3C

Adafruit_SSD1306 display(OLED_RESET);
volatile unsigned long v1_left_time, v1_right_time, v2_right_time, v1, v2;
float last_v1, last_v2;
int timerStart, timerStop;
int frequency;
double periodMS;

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 0);
  display.println("Find Audio Direction");
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 20);
  display.println("Angle = ");
  display.display();

  Serial.begin(9600);
}

void loop() {
  last_v1 = v1;
  last_v2 = v2;
  v1 = analogRead(A0);
  v2 = analogRead(A3);
  Serial.print("left:"); Serial.print(map_float(v1, 0, 1023, 0, 100)); Serial.print(" ");
  Serial.print("right:"); Serial.print(map_float(v2, 0, 1023, 0, 100)); Serial.println();

  if (v1 < 0.1 && !(last_v1 < 0.1)) {
    v1_left_time = micros();
  }
  if (last_v1 > 0.1 && v1 < 0.1) {
    v1_right_time = micros();
    periodMS = (v1_right_time - v1_left_time);
    frequency = 1 / periodMS;

  }
  //Serial.println(periodMS);
  if (v2 < 0.1 && !(last_v2 < 0.1)) {
    v2_right_time = micros();
  }

}

float map_float(float value, float from_low, float from_high, float to_low, float to_high) {
  return (value - from_low) * (to_high - to_low) / (from_high - from_low) + to_low;
}

int calculatePhaseAngle(int timer) {//converts time difference into phase angle (degrees)
//this also rejects the out of range frequencies with a 50hz tolerance
  if (frequency < 750 || frequency > 1050) {
    frequency = 0;
    return 0;
  }
  return (360 * frequency * timer);
}
```

The current problem I have with the code above is that I am getting inconsistent data from the microphones. Therefore, I plan to test more of the code on different frequencies and figure out the flaws of the code.

Figure 1

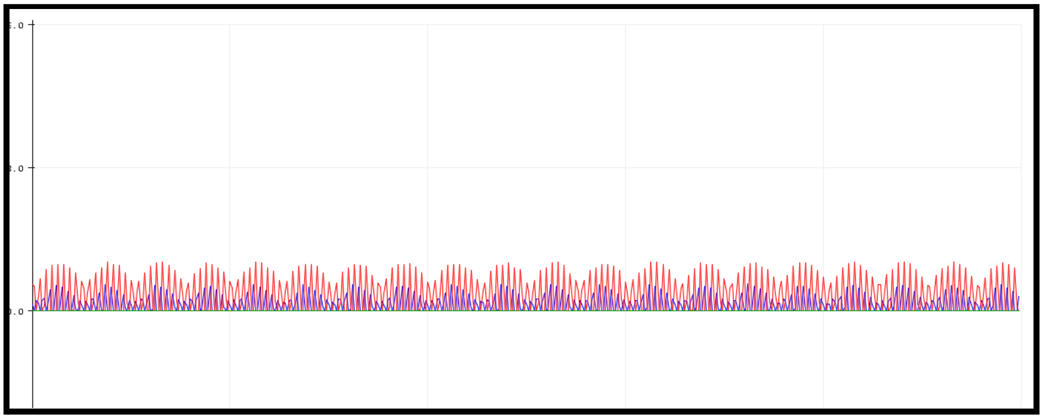

Figure 1: Test trial 1: Output from the code above (Speaker playing at 900 hertz frequency)

Figure 2

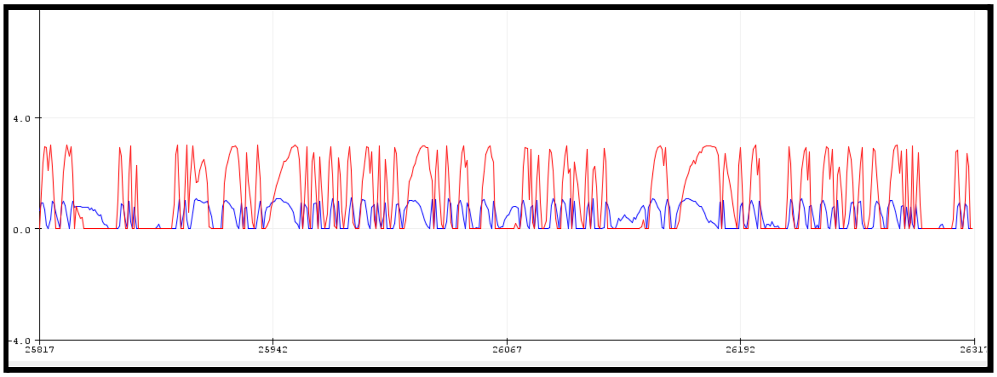

Figure 1: Test trial 2: Output from the code above (Speaker playing at 900 hertz frequency)
Example of inconsist data from the microphone

Figure 2

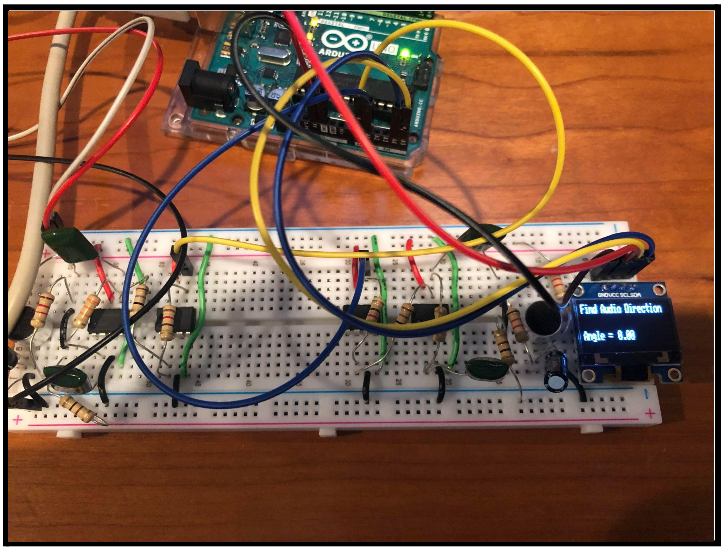

Figure 2: We have been able to read data from the microphones and be able to display data on the OLED. This picture also shows our filter and amplifier of the two mics

Plans For Next Week

The plan for next week is to finish up the project and do more testing to refine the system. The code is incomplete so finishing this is the top priority. Without the code, the angle cannot be calculated so this is essential for this project to be completed.

Author Recognition
Jeff Kedda - **Test Code Part 1**

Saksham Goyal - Milestone writeup and testing code.