# Design Overview

## Introduction

In this project, we plan to create a circuit design where we use 2 microphones to identify the source of the location. We will label the source as an angle to the microphones and then use the display in our kit to output the value. Along with the microphones, our solution must incorporate both analog hardware and a microcontroller (Arduino). Audio Direction finding is a relevant skill in today’s world and can be used to find the direction of an incoming cell phone call.

The purpose of this project is to be able to work with a partner to create an audio direction finder which requires the use of both hardware and software. This project will demonstrate our understanding of implementing filters and will test our problem-solving skills. The project will involve creating algorithms, creating equations that relate to our previous physics and math classes, and the understanding of filters.

## Overview

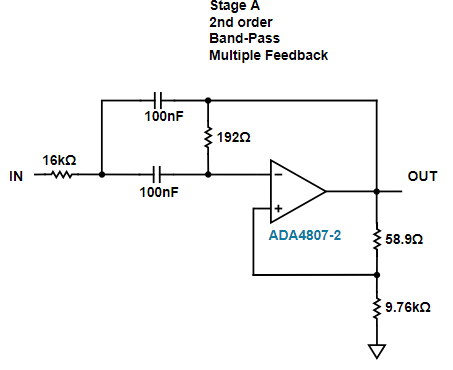
Our outlined plan will involve steps that allow us to debug and test each component of the project. First, we must read the datasheet for the electret microphone and the Arduino so we can connect the microphone to the Arduino. After that, it will be suitable to create a program that reads the signal from the microphones after passing it through a band pass filter to eliminate the noise so we can get a clean signal and only allow the 900Hz signal.

To calculate the angle, we can perform some operations on the 2 signals that we receive to figure out the difference in the time between the 2 signals. Since the distance between the 2 microphones will be fixed, and the speed of sound is known, we can easily use an algorithm to calculate the angle of the sound.

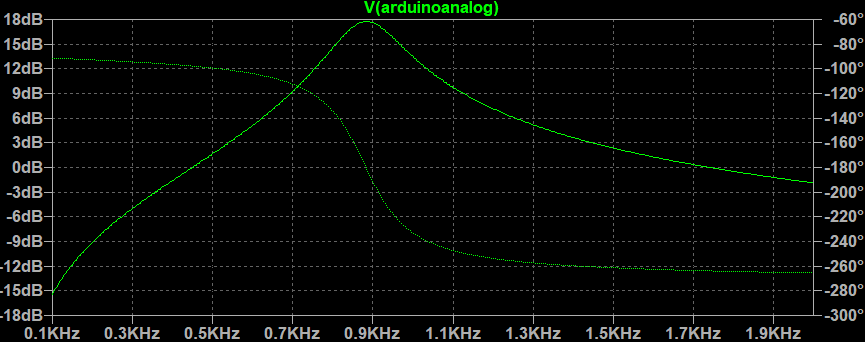
After the angle is found, we need to output the value to the display in out kit. The display in our kit uses the I2C protocol to communicate. So, we will need to use that to connect it to the Arduino.

The steps to produce the resultant angle are as follows. First, we get the signals from the microphones into the opamp to amplify the signal and create a band-pass filter. This signal is then sent to the Arduino’s analog ports. The Arduino will convert the analog signal into a digital one with it’s built in ADC. Once we have a clean version of the 2 signals, we can perform calculate the phase shift in the 2 signals by comparing their phase. When we get the phase shift, using the known speed of sound and the known distance between the 2 microphones, we can calculate the angle between the normal (the line that bisects the 2 microphones) and the source of the sound. We then use the I2C interface to send the result to the display where the user can then see the value.

## Block Diagram:



Using the filter wizard and estimating some of the values to create the filter, these are the values that we have come to for the amplifier. I created an LTspice simulation of the filter, and this is the analysis of that circuit:



This is obviously not going to reflect how this will perform in real life since the resistor values and op-amp are not ideal so there will be variance in this output simulation. However, it should still be good enough to amplify the signal and create a bandpass so we can do the calculations.

# Future Milestones

1. Band-pass Active amplifier with Op-amp
   1. Wire up the op-amp and Arduino
   2. Test filter
   3. Test amplification
   4. Make a simple program that tests the microphone and reads the data into the microphone.
2. calculate the phase shift in the 2 signals by comparing their amplitude.
   1. Create equation that allows us to calculate the phase shift in the 2 signals.
   2. Create a program that does this.
   3. Compare Measured results to calculated results.
3. calculate the angle between the normal (the line that bisects the 2 microphones)
   1. Create equation that allows us to calculate the angle between the normal
   2. Create a program that does this.
   3. Test the program by putting the speakers at different angles.
   4. Compare Measured results to calculated results.

To test the hardware in out kit, we can test each component individually to be able to identify exactly where the issue is, should we find an issue. Since the only hardware we have is the microphone and the display, it should be simple to test them.

To test the mic, we can use the speaker that is also provided in our other kit and check if the microphone picks up the speaker and reads the correct frequency. If there is a mismatch, we can use the other microphone to check if the problem is the speaker or the mic.

To test the display, we can find that out by just using it and checking visually. If we see that when we display the angle when we complete the circuit, and after we can verify that the code is working properly, if something is wrong, then we can conclude the display is not working correctly.