Sound Localization System

Design Lab Project

By:

Devashish Taneja
170102026

Under the guidance

of

Dr. Ribhu

Department of Electronics & Electrical Engineering
Indian Institute of Technology Guwahati

Aim of the Project:

- The aim of the project is to build a system which can detect the direction of the source of sound.
- Sound Localization system has multiple applications

    - Navigation : Collision avoiding

    - Robotics : Auditory response

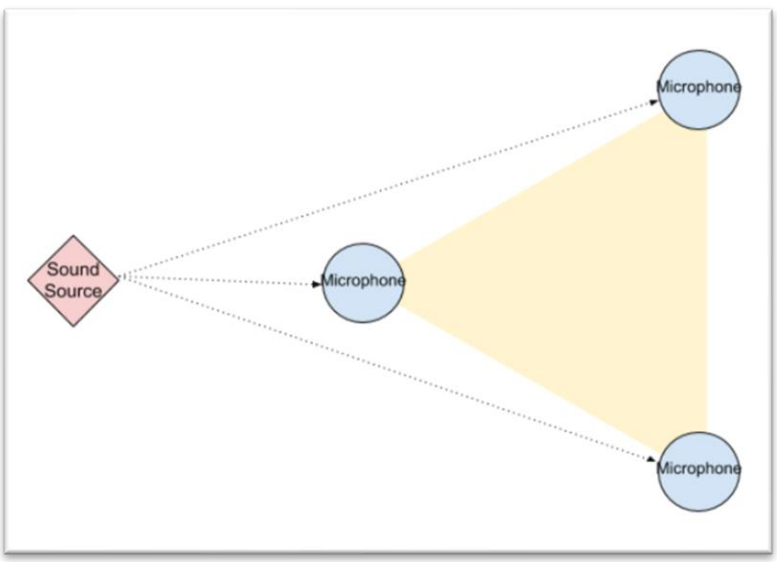
Components Required:

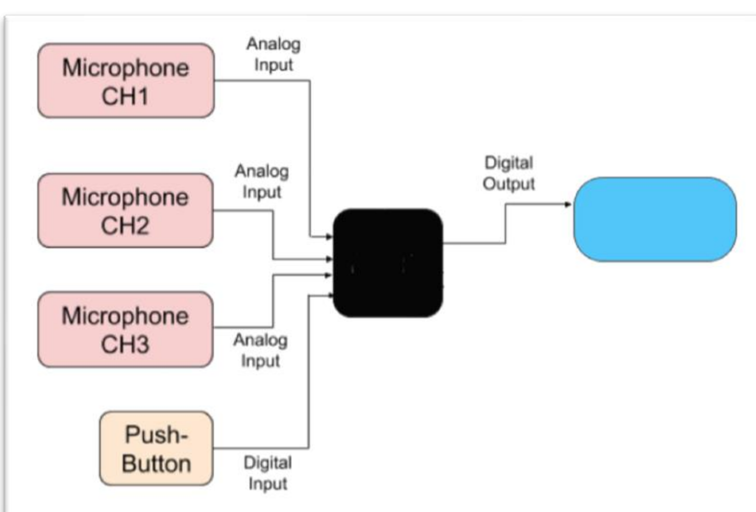
- Arduino Microcontroller
- Breadboard
- 3 x Microphones
- 2 x 180 Servo Motors
- Connecting Wires
- Battery
- Acrylic Sheets

Workplan:

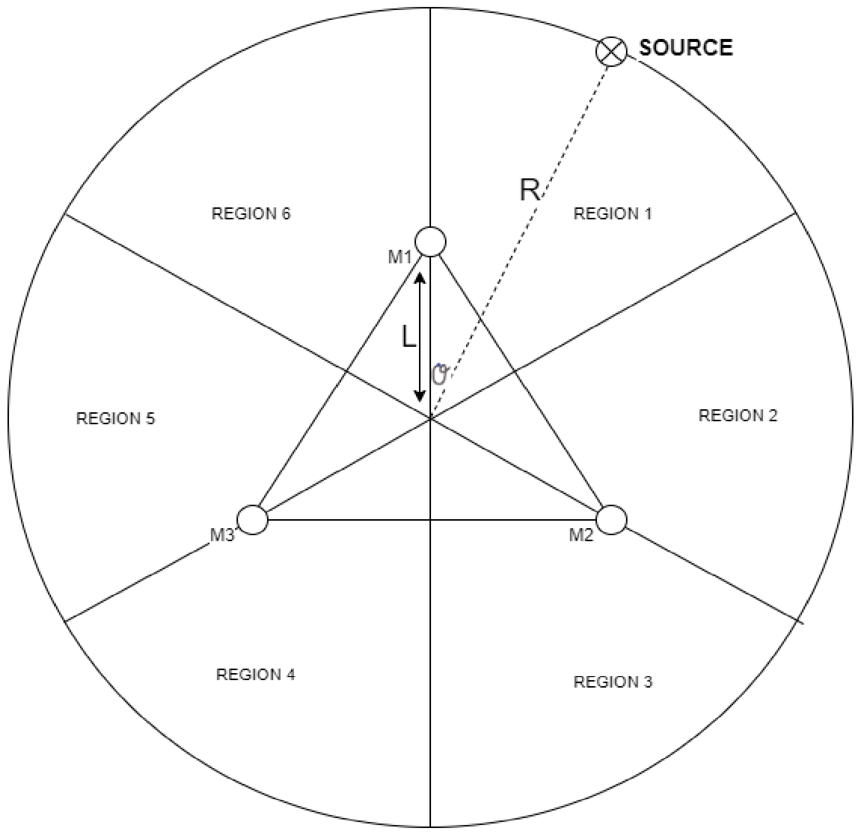
- 3 Microphones are arranged in a triangular formation to determine the location of source.
- We calculate the angle of the source using the delay between the microphones.
- To point to the source, we use 2x180 servo with one mounted on of the other one to simulate like a 360 servo.

Illustration Diagram:





Theory:



Here, SM_i & t_i is the distance and time of arrival for the i th microphone.

Using Cosine Rule, we get

$$SM_1 = \sqrt{R^2 + L^2 - 2RL\cos(\theta_1)}\ ,\ \theta_1 = \theta$$
$$SM_2 = \sqrt{R^2 + L^2 - 2RL\cos(\theta_2)}\ ,\ \theta_2 = 120 + \theta$$
$$SM_3 = \sqrt{R^2 + L^2 - 2RL\cos(\theta_3)}\ ,\ \theta_3 = 120 - \theta$$

$$t_1 = SM_1 \ / \ v$$
$$= \sqrt{(R/v)^2 + (L/v)^2 - 2RL\cos(\theta_1)/v^2}$$
$$t_2 = SM_2 \ / \ v$$
$$= \sqrt{(R/v)^2 + (L/v)^2 - 2RL\cos(\theta_2)/v^2}$$
$$t_3 = SM_3 \ / \ v$$
$$= \sqrt{(R/v)^2 + (L/v)^2 - 2RL\cos(\theta_3)/v^2}$$

$Here,\ d_{ij}\ represents\ the\ difference\ in\ time\ of\ arrival\ for\ i\ th\ and\ j\ th\ microphone.$
$Putting\ a\ =\ (R/v)^2 + (L/v)^2\ and\ b\ =\ 2RL/\ v^2,$
$we\ get$

$$t_1 = \sqrt{a - b\cos(\theta_1)}$$
$$t_2 = \sqrt{a - b\cos(\theta_2)}$$
$$t_3 = \sqrt{a - b\cos(\theta_3)}$$

$$d_{12} = \sqrt{a - b\cos(\theta)} - \sqrt{a - b\cos(120 + \theta)}$$
$$d_{13} = \sqrt{a - b\cos(\theta)} - \sqrt{a - b\cos(120 - \theta)}$$
$$d_{23} = \sqrt{a - b\cos(120 + \theta)} - \sqrt{a - b\cos(120 - \theta)}$$

<u>Since these equations are non-linear and difficult to solve we divided the detection region in six parts and used the delays between microphones to decide the region.</u>

REGION	***SIGN***		***PROPERTY***
	D1(d12)	***D2(d13)***	
1	**-**	**-**	***|D1| < |D2|***
2	**+**	**-**	***|D1| < |D2|***
3	**+**	**+**	***|D1| < |D2|***
4	**+**	**+**	***|D1| > |D2|***
5	**-**	**+**	***|D1| > |D2|***
6	**-**	**-**	***|D1| > |D2|***

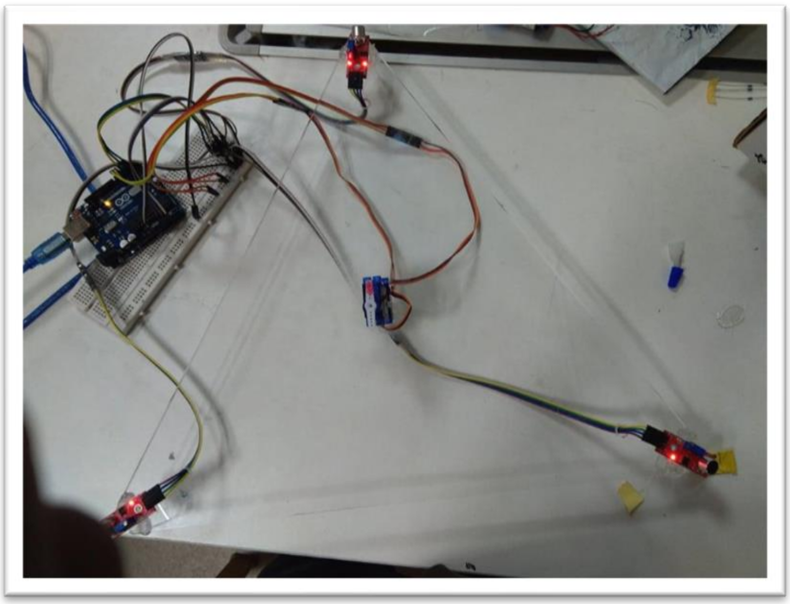
Code:

```
#include <Servo.h>
Servo myservo1, myservo2;
int a0 = A0;    // select the input pin for the potentiometer
int a1 = A1;    // select the input pin for the potentiometer
int a2 = A2;    // select the input pin for the potentiometer
int angle;
int d0=4;
int d1=3;
int d2=2;
void setup()
{
  // declare the ledPin as an OUTPUT:
  pinMode(d0, INPUT);
  pinMode(d1, INPUT);
  pinMode(d2, INPUT);
  //Serial.begin(9600);
  myservo1.attach(5);
  myservo2.attach(6);
  Serial.begin(9600);
  myservo1.write(0);
  myservo2.write(0);
}

void loop()
{
 int data0= digitalRead(d0);
 int data1= digitalRead(d1);
 int data2= digitalRead(d2);
 Serial.println();
 Serial.println(data0);
 Serial.println(data1);
 Serial.println(data2);
 Serial.println();

 if(data0==1 & data1==0 & data2==0)
 {
  angle=0;
 }
 if(data0==0 & data1==1 & data2==0)
 {
  angle=240;
 }
```

```
 if(data0==0 & data1==0 & data2==1)
 {
  angle=120;
 }
 else
 angle=0;
 if(angle<=180)
{
  myservo2.write(0);
  myservo1.write((160*angle)/180);
}
else
{
  myservo1.write(160);
  myservo2.write((190*(angle-180))/180);
 // delay(600);
}
//int v0 = analogRead(a0);
//int v1 = analogRead(a1);
//int v2 = analogRead(a2);
//Serial.println(v0);
//Serial.println(v1);
//Serial.println(v2);
//Serial.println();
Serial.println(angle);
delay(300);
}
```

Current Status:

- Currently, we were able to detect the direction of sound using a clap sound.
- We planned to add correlation function to detect the delay between the microphones for continuous sound.



Conclusions:

- We started with the aim of detecting the sound for all possible directions but could finally detect sound for only 6 possible regions.
- For most of the cases the detection was correct but there was some inconsistency in the detections.