REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

ACOUSTIC SOUND SOURCE LOCALIZATION USING RASPBERRY PI

17011087 — Mucahit KARABULUT
17011039 — Ömer Aras KAPLAN

SENIOR PROJECT

Advisor
Assist. Prof. Dr. Ali Can Karaca

June, 2022

# ACKNOWLEDGEMENTS

We would like to thank our families, friends, professors for supporting us and giving this opportunity. We especially want to thank our dearest advisor Ali Can Karaca for aiding us in this project whenever we needed help. We also would like to appreciate and show gratitude towards his guide to us throughout this process. We can not stress his help to the project enough. It is clear that without his efforts this project would not be anywhere near this level.

Mucahit KARABULUT
Ömer Aras KAPLAN

TABLE OF CONTENTS

LIST OF SYMBOLS

Δt	Delayed Time Distance

LIST OF ABBREVIATIONS

OS	Operating System
MEMS	Micro Electromechanics
GPIO	General Purpose Input/Output
RAM	Random Access Memory
GB	Gigabyte
GHz	Gigahertz
MB	Megabyte
SD	Secure Digital
IDE	Integrated Development Environment
I2C	Inter Integrated Circuit
I2S	Inter IC Sound
MHz	Megahertz
WS	Word Select
SD	Serial Data
SCK	Signal Clock
NC	Normally Closed
NO	Normally Open

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Acoustic Sound Source Localization using Raspberry Pi

Mucahit KARABULUT
Ömer Aras KAPLAN

Department of Computer Engineering
Senior Project

Advisor: Assist. Prof. Dr. Ali Can Karaca

In today's standards sound localization is used in fields of rescue operations and military facilitations. Ranging from detecting a sniper's position to find people stuck under the rubbles after an earthquake. With the rise on deep learning models, the way of process can differ from using conventional angle calculations to machine learning. Most examples of the subjects are performed by utilizing two microphones or iterating over and over.

In this study, it is targeted that the localization is done on one cycle with four microphones. The goal is to give the user precise approximations on coordinates and angles of the sound source on three dimensions in real time restrictions. In this paper, a way of projecting a sound source using three different angles is presented. In addition, a different approach on using more than two I2S MEMS microphones on general purpose boards are presented. The content in the paper can be applied to relevant situations with minimal modification with real time working conditions.

The choice of electronic board is a Raspberry Pi 400, due to its versatility, price, speed and portability. The application can be utilized in a more specialized System-On-Chip devices to increase performance, reduce costs and provide more portability.

Keywords: Time Difference of Arrival, Binaural Sound Localization, Raspberry Pi, I2S, Location Estimation, MEMS microphone, INMP441

ÖZET

Raspberry Pi Kullanarak Ses Kaynağı Konum Tahmini

Mucahit KARABULUT
Ömer Aras KAPLAN

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Dr. Ögr. Üyesi Ali Can Karaca

Günümüz standartları içerisinde ses kaynağının yerinin tahmini, birbirinden farklı birçok alanda kullanılmaktadır. Askeri alanlarda keskin nişancıların konum tahmini için veya sivil alanlarda doğal afetler sonrasında, özellikle depremlerden sonra, mahsur kalan kazazadelerin yerini tespit etmek gibi farklı alanlarda geniş yelpazelerde kullanılabilmektedir. Derin öğrenme modellerinin yükselişiyle beraber kaynak bulma işlemi farklı şekillerde yapılmaya başlanmıştır. Günümüzde açılar üzerinden giderek daha geleneksel yaklaşımlar kullanılmakla beraber, makine öğrenmesi kullanılarak daha yeni yaklaşımlar da kullanılmaktadır. Şimdiye kadar geleneksel yöntemlerle yapılmış olan araştırmalarda genellikle iki mikrofon kullanıldığı ve iki boyuttaki açı üzerinden gidildiği, ya da dört mikrofon kullanılarak birden fazla iterasyon yapılarak sonuca ulaşıldığı görülmüştür.

Bu çalışmada, kaynak bulma işleminin tek iterasyonla ve dört mikrofon kullanılarak yapılması amaçlanmıştır. Projenin beklentisi, kullanıcıya yüksek keskinlikte koordinat ve açı tahminlerini gerçek zamanlı sistem kısıtları içerisinde sunmaktır. Bu makalede, üç farklı boyuttaki açılar kullanılarak konum tahmininin nasıl yapılabileceği açıklanmıştır. Ek olarak genel amaçlı elektronik araçlarda I2S bağlantı sayısı yetmediğinde, kullanılabilecek bir çözüm de üretilmiştir. Makaledeki içerik benzer durumlarda minimal değişiklikler yaprak gerçek zamanlı olarak kullanılabilmektedir.

Projedeki elektronik araç seçimimiz Raspberry Pi 400 modeli olmuştur. Bu modeli seçme nedenimiz çok yönlülüğü, ücreti, hızı ve taşınabilir olmasıdır. Proje içeriği System-On-Chip araçlarına uygulanarak performansı artırılabilir, ücreti azaltılabilir ve

daha fazla taşınabilirlik sağlanabilir.

Anahtar Kelimeler: Ses Kaynağı Konum Tahmini, Çift Mikrofon Konum Tahmini, Raspberry Pi, I2S, Kaynak Tahmini, MEMS Mikrofon, INMP441

1
Introduction

To reveal the idea behind this project, we have to take a look at a similar system that is being used by mammals for millennias. Put simply, mammals use the difference between two signals' intensity and phase.

Similar to representing points in cartesian systems using polar notation, two values are needed. Normalization angle and distance to midpoint. To obtain them, we use binaural sound localization.

Due to sound waves being transmitted in a circular wave form, sound arrives at receivers on different times. While doing so, the wave keeps its characteristic properties. In regards to this instance, by checking the margin of arrival times on microphones, the angle of arrival can be deduced. In order to obtain a distance metric, a geometrical model is prepared.

Nowadays sound localization has a wide range of applications. Some examples would be voice conferences, localization of sources in robotics, determining ground zero for explosions, sniper location finding and rescuing people from rubbles in earthquake zones.

A note to keep in mind during this project is that using more microphones might yield better results. While doing so the distance between microphones must be measured and known.

In later pages, wiring of the system, phase correlation calculation, coordinate estimation are explained in a more detailed manner.

2
Review

Before starting the project, it is obvious that a literature search is due in order to achieve greater results. We gather articles that it is similar to what we have in mind.

2.1 A Probabilistic Model for Binaural Sound Localization

The first of these articles "A Probabilistic Model for Binaural Sound Localization" [1]. The article mentions how human sound localization system works. When we understand how humans localize sounds, we can make a model which represents human localization system. Also it mentions how many microphones are needed to determine where the sound is coming from. We also gained information involving formalization of normalization angle.

2.2 Advanced Binaural Sound Localization in 3-D for Humanoid Robots

In this article the contributors used conventional methods with deep learning methods. They found the angle by repeatedly getting inputs and rotating the system to the calculated angle. When the last two angles are the same they stopped iterating and compared the results.

For the distance part they have trained a deep learning model. Which has an error rate of %10 on distances less than 10 meters and up to %36 on distances to 20 meters.

Their project is more accurate when the sound is more skewed to the axes. [2]. An aim of our project is rotate system to where sound is coming from, The article mention same subjects with the other one. Additionally, the article consist of the subject rotation.

2.3 Full-Sphere Binaural Sound Source Localization Using Multi-task Neural Network

In this article, the contributors developed a fully spherical SSL system. They used two parallel convolutional neural networks where the output is interaural and monaural localization cues. The SSL is composed of two sections. One is specialized in predicting the azimuth value and the other is specialized in predicting elevation.

Said system is trained at different environments involving different noise, temperature and other variables.

They concluded that the system is very accurate, reaching %96 and more accuracy on their tests.[3]

2.4 End-To-End Binaural Sound Localisation From The Raw Waveform

In this article, instead of explicit feature extractions, contributors proposed a convolutional neural network with a cascading array of layers to extract cues directly from the raw waveform.

Their project is able to accurately predict the azimuth levels in a non-echo environment. But the results tend to fall in places that is able to echo sounds.

They generated three different models called WaveLoc. Each of them excelling at different environments.[4]

2.5 Methods For Robust Binaural Sound Localization

In the article it is suggested that machine based methods utilise head related transfer function(HRTF) datasets, which are not really applicable to real life situations. In the article it is stated that an HRTF based interaural, full spherical system with a direction of arrival augmentation would be robust to noise, different sound types and echoes.[5]

2.6 A Time Difference of Arrival Estimation with Integration of Generalized Cross Correlation Samples for Continuous Wave Signals on Passive Positioning Systems

In this paper it is stated that any miniscule propagation or delay in TDOA systems can cause a deviation from the true location for hundreds of meters. Later on they have stated that this delay can be minimized by using generalized cross correlation and maximum likelihood approaches. After that the contributors compared different GCC algorithms and stated that GCC-PHAT algorithm, which is used in this paper, will yield a lower accuracy when signal to noise ratio(SNR) is low.[6]

2.7 Localization of an Acoustic Emission Source Based on Time Difference of Arrival

In this article, the contributors designed a project very similar to the design we have used in our project. In their paper, they designed an environment where all microphones are scattered along an area. In their design the sound source is in the middle of the microphones. The difference between their and our project is that in our design the sound source is outside of the area between the microphones. In the paper, the contributors used two different algorithms, namely Gradient Descent Algorithm and Firefly Algorithm. Their distance results were extremely efficient.[7]

3
Feasibility

3.1 Technical Feasibility

3.1.1 Software Feasibility

While python is good at reading sensor values from GPIO pins, MATLAB excels at processing signal operations. In the experimenting phase we use python to read sensor values and MATLAB for processing signals.

Algorithms and basic concepts of sound source localization were developed in MATLAB environment and real time version was implemented in Python.

While developing the project we use Raspberry Pi's standard Python IDE, Thonny, which is a lightweight python environment developed specifically to run python code on embedded systems.

3.1.2 Hardware Feasibility

Since the project is intended to work as a real time system, there would be certain criteria to meet. Aside from software quality, the system is recommended to run in a system with at least 4 GB RAM and a multi thread capable processor with at least 1.5 GHz clock speed. At least 512MB of memory space is needed to avoid crash.

Aside from an electronic board the system uses four INMP441 MEMS microphones and a 4 channel 5V relay.

3.2 Legal Feasibility

The project does not use personal data, neither shares it with a third party, nor keeps data more than 5 seconds. Therefore there is no legal binding or conflict.

3.3 Economical Feasibility

The project is being made by 2 developers which has a salary of 5000 Turkish Liras spanning over 3 months. The equipment list and their prices are as follows:

- Raspberry Pi 400 4 GB:2300 Turkish Liras
- Arduino Uno: 175 Turkish Liras
- Samsung 100 GB SD Card: 100 Turkish Liras
- 4 X SPW2430 MEMS Microphone: 400 Turkish Liras
- 8 X INMP441 MEMS Microphone: 800 Turkish Liras
- 120 X M-M Jumper Cable: 120 Turkish Liras
- 120 X F-F Jumper Cable: 120 Turkish Liras
- Soldering Iron: 60 Turkish Liras
- 2 X 5 Meter Soldering Cable :30 Turkish Liras
- 40 Meter Copper Wire: 120 Turkish Liras
- 3 X 3.3V-5V Level Shifter : 15 Turkish Liras
- 4 Channel Relay Card : 45 Turkish Liras
- Protractor Set: 12 Turkish Liras

For a total of 4297 Turkish Liras

3.4 Time and Labor Management

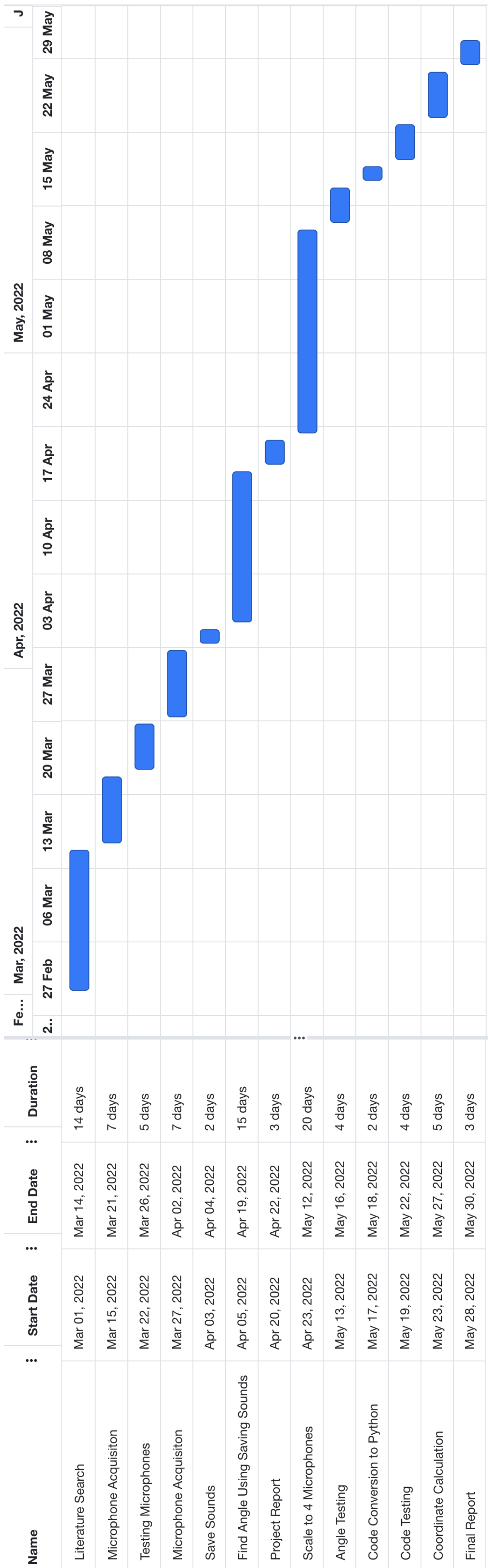

Figure 3.1 Gantt Diagram

4
System Analysis

4.1 Microphones

Since sound quality and quick sampling speeds are two of the most important factors of this project due to sensitivity of the signals to noise and the fact that these operations have to be handled in a matter of miliseconds, we targeted a microphone with I2S connections, high resolution and high sampling. The optimal microphone to use in our project is INMP441.

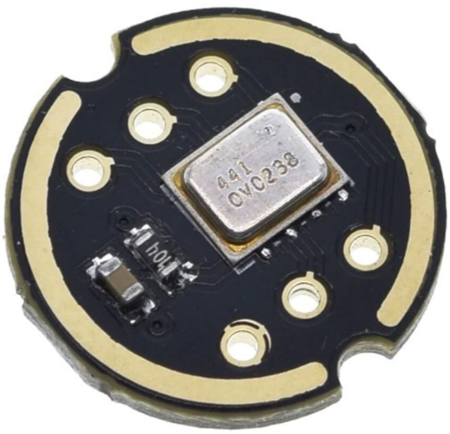

Figure 4.1 INMP441

One of the microphones are set on the center and the other microphones are set perpendicular to the center microphone with a distance of 53 centimeters.

4.2 Input

In this project the system is always running and standing by for values higher than the threshold. When a sound higher than the threshold arrives, the system will start finding angles from each running microphone.

4.3 System Design

System's latest design in real life environment.

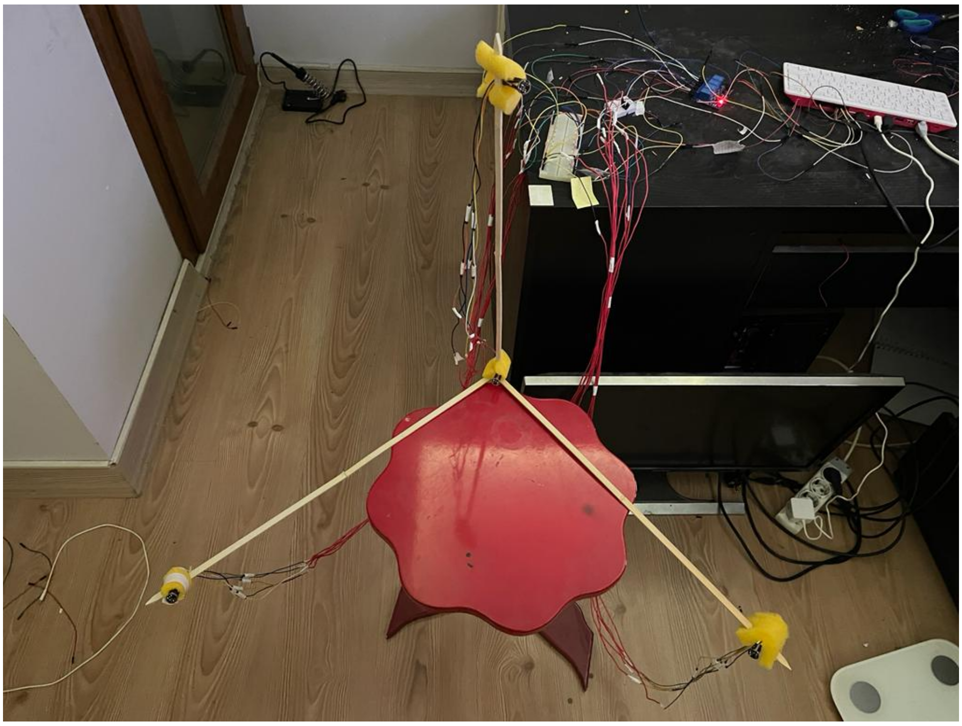

Figure 4.2 Platform

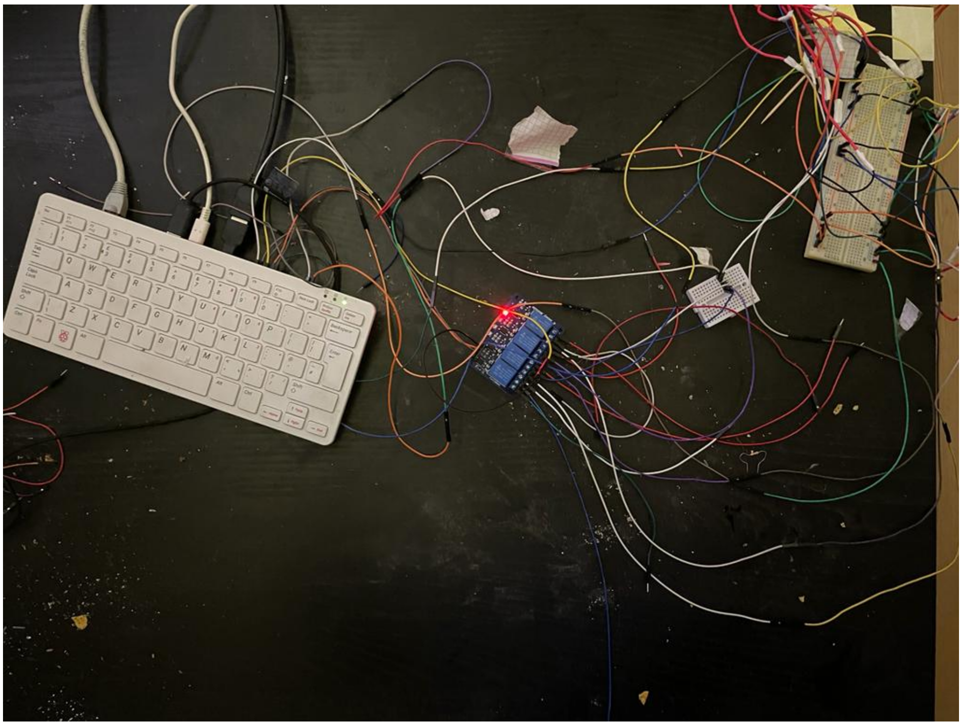

Figure 4.3 Raspberry to Breadboard Wiring

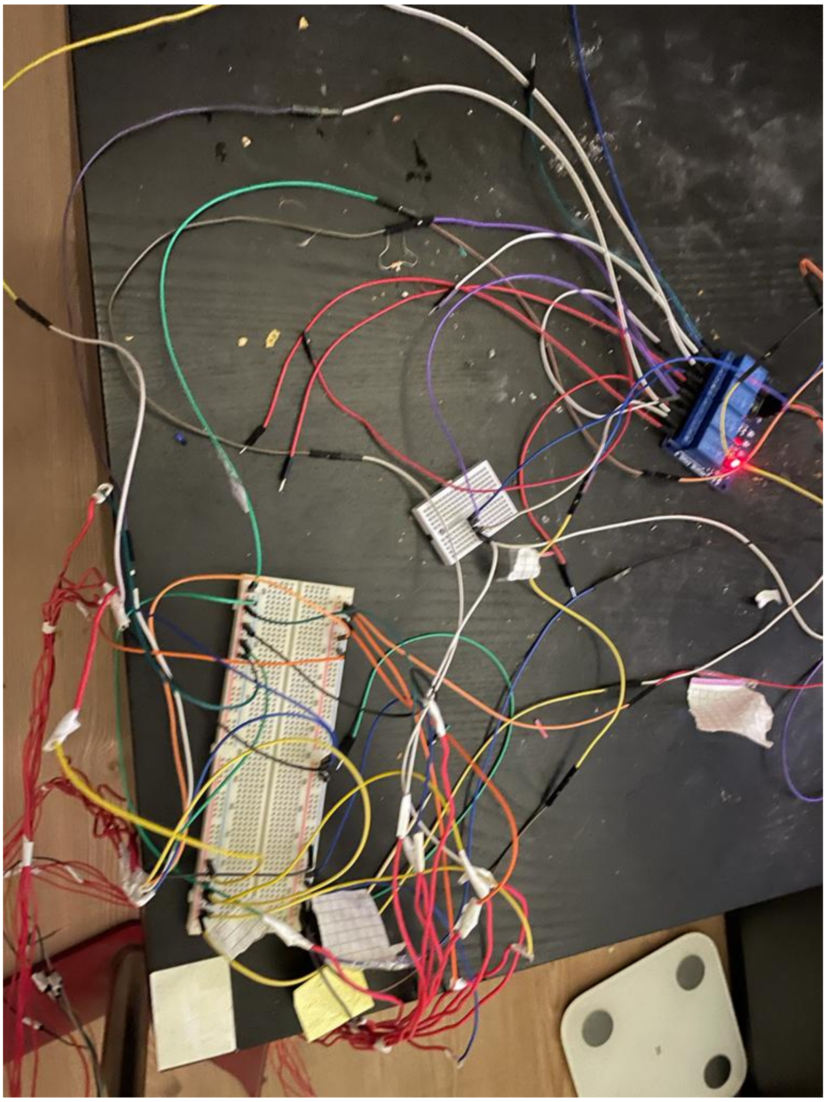

Figure 4.4 Relay Wiring

4.4 Angle Calculation

When the input data is given, system will begin to perform a 2 layer FFT on the signals and conjugate them. After this operation the distance between peak frequency samples are found. These processes are called "Generalized Cross Correlation- Phase Transform".

After cross correlating the signals, the product value is multiplied with the speed of sound to determine the extra distance of arrival. After using arcsine function on the division result of these two values, the angle is found.

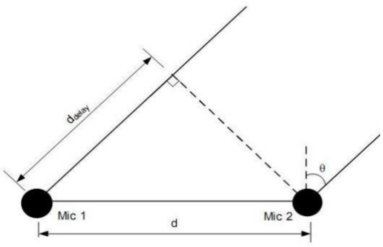

Figure 4.5 Time Difference Deduction Between Microphones

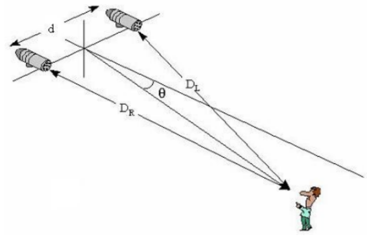

Figure 4.6 Angle Deduction

4.4.1 DFT Formula

The equation below is the formula to obtain the DFT of signal. Since they produce the same result, DFT formula can be examined for further insight below.

$$k = 0, ...N-1$$

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N}$$

After performing 2 layer FFT on the signal, the next step is to conjugate the signals.

4.4.2 Equations for Angle Calculation

Audio signals' equation from the same source can be written as below.

$$r_1(t) = s(t - \Delta t) + n_1(t)$$

$$r_2(t) = s(t) + n_2(t)$$

Now that time of delay between arrivals are known, the angle can be computed.

$$\theta = arcsin(d_{delay}/d) = arcsin(v.\Delta t/d)$$

4.5 X-Z Dimension Coordinate Deduction

Given the angles and the starting points of these lines, it is possible to calculate their intersection. Giving us the ability to determine the coordinates of the sound source projecting in the X-Z dimension.

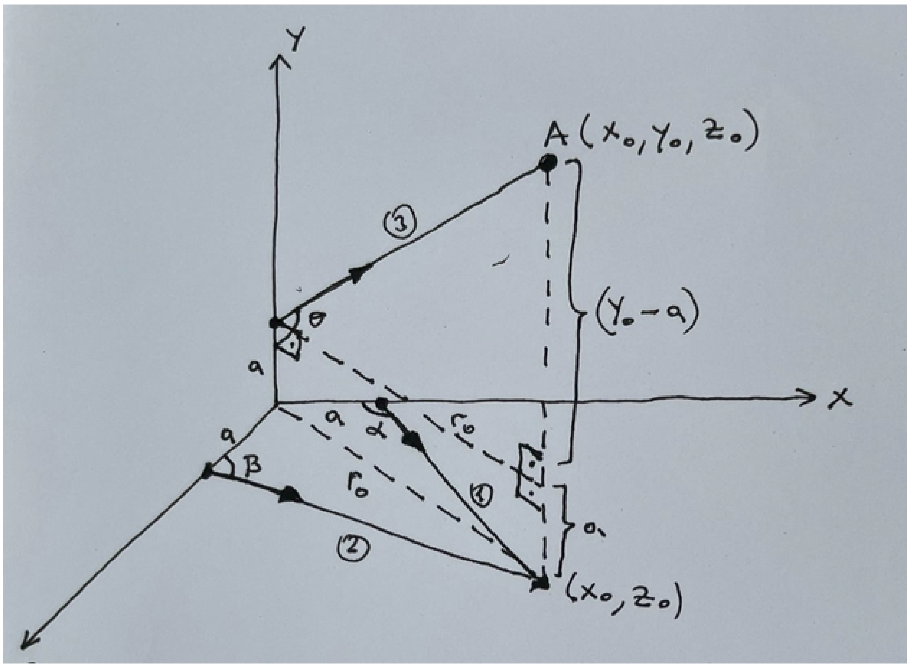

Figure 4.7 Model of Calculation

4.5.1 Baseline For Equations

The equations that can be found in the later parts of this article are derived from the equations below. All the equations used in this project have the geometrical origin of the model in 4.7.

$X_0 = a + tan(\alpha - 90) * Z_0$

$Z_0 = a + tan(\beta - 90) * X$

$Z_0 = (a + tan(\beta - 90)) * [a + *tan(\alpha - 90) * Z_0]$

$Z_0 * [(1 - tan(\beta - 90) * tan(\alpha - 90)] = a[1 + tan(\beta - 90)]$

4.5.2 Z Axis Microphone Angle

The formula to obtain Z axis level of the sound source is as follows.

$Z_0 = a * [1 + tan(\beta)]/(1 - tan(\beta - 90) * tan(\alpha))$

4.5.3 X Axis Microphone Angle

The formula to obtain X axis level of the sound source is as follows.

$X_0 = a * [1 + tan(\alpha - 90)]/(1 - tan(\beta - 90) * tan(\alpha - 90))$

4.6 Elevation Deduction

Given the solution coordinates in the X-Z dimension is known, it is possible to generate a line formula to find the elevation Y level.

The formula to obtain Y axis level of the sound source is as follows.

$r_0 = \sqrt{x^2 + y^2}$

$Y_0 = a + r_0 * tan(\theta - 90)$

4.7 Output

As the last step of the system process, the system will print the angle and distance to target and show a 3D vector.

5
System Design

5.1 Development Phase

While developing this project time is a luxury we could not afford. So in the process it is clear that some modulation and scaling tasks were needed.

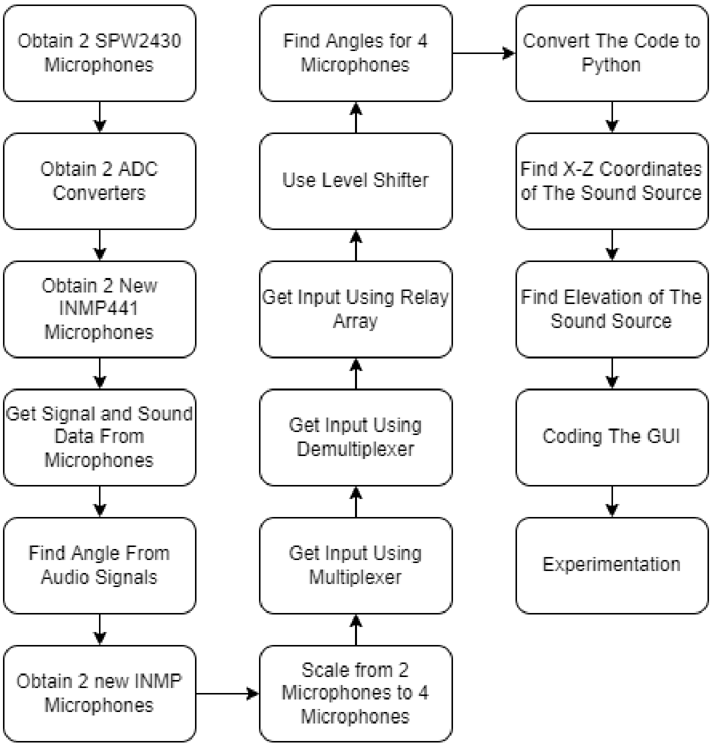

Figure 5.1 Development Phase

The development of this project was a path full of errors. Nearly every step taken to achieve results yielded more errors to look into. Though, in the end a product was made.

The Development started with obtaining two MEMS microphones. These two microphones needed analog to digital converters to run properly. Sadly, there was not any available converters to satisfy the microphones sample rate and resolution for nyquist rate requests.

The microphones were switched with new INMP441 microphones. They needed no extra material to get the signals. Wiring was done and the signal was clear. The initial code written for angle deduction was faulty. It could not detect certain angles. When the code is fixed into a better state, development of the project moved to the next step, scaling the system to four microphones.

Scaling the system to four microphones took more time than expected given that the microphones use I2S protocols and not the I2C protocol. After elaborate research I2S protocol can only use two devices for one bus. Since Raspberry Pi has only 1 I2S bus pin, a new design should be made to use four microphones.

The project does not necessarily need four microphones to run simultaneously. It only needs the source to be stable while getting audio signals. In light of this knowledge, 4x1 multiplexers were ordered.

When multiplexers arrived, the SD lines of corner microphones were connected to the multiplexer. Which made it possible to switch microphones from raspberry. The signals read from multiplexer were distorted. Due to the propagation delay of the multiplexer. Which made faulty estimations.

To delay each line the same, all lines were connected to multiplexers. The result was even worse, no sensible output was gathered.

After that Vcc cables were connected to a demultiplexer to cut off voltage supplies. This method proved futile as well because the back current from other cables made the microphone send random data from SD line, making it impossible to read.

The last solution to read from microphone data was to use a relay array to change microphones. The relay switch happens in an instant, making it perfect for switching dynamically.

After finding a way to use four microphones at once, the code is converted to python code. There was some adjustments to run it in a real time manner.

Lastly, the code to calculate angles and distance was written.

5.2 System Usage

System will wait for a sound signal higher than the threshold. If such higher signal is granted, system will perform 2 layer FFT and conjugation on the signals. Then the angle is found using the aforementioned formula. After finding all the angles, the system will continue to find the distance of the sound source from the center microphone.

After these steps, the angles and distance is shown on the user's screen.

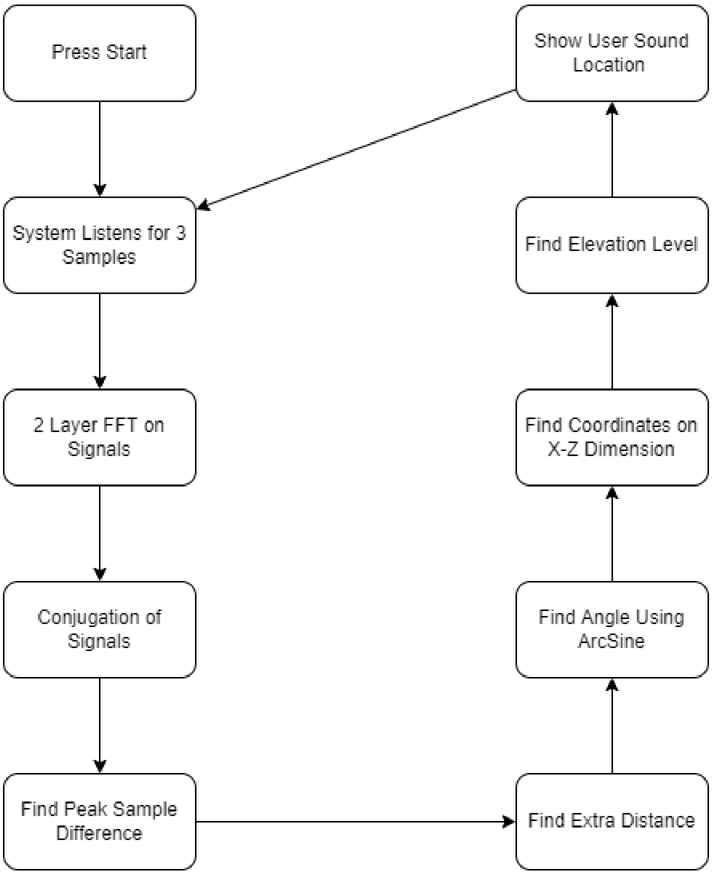

Figure 5.2 System Usage

6
Application

6.1 Circuit Design

6.1.1 Pinout

The microphones used in the project are supplied with 3.3V from Raspberry Pi's first pin. The 5V to supply the relays are connected to sixth pin of the Raspberry Pi. For I2S communication, 4 pins were used. The twelfth pin is configured as the clock generator. The thirty fifth pin is for word select. Lastly, the thirty eighth pin was used to gather serial data. These pins are specialized for PCM and PWM modulations.To work with inmp441 there are a few library needed to be installed [8]

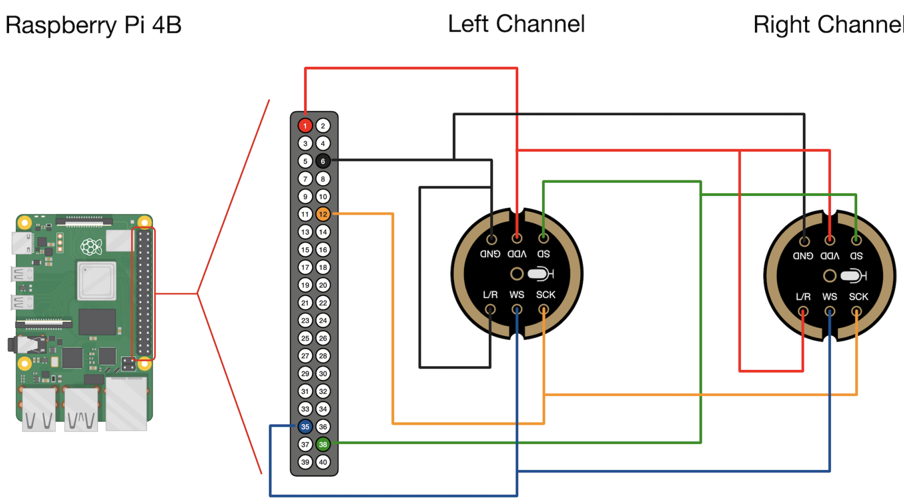

Figure 6.1 Microphone Wiring [9]

6.1.2 I2S

The INMP441 microphones run on a protocol known as I2S also known as Inter IC Sound. This protocol is specialized in communicating high quality audio data. I2S is able to transmit single or dual channels of audio with clock speeds such as 12 MHz for 192 kHz sampling rate. Data word generally used in this protocol is 32 bits. I2S is not very distant from SPI but the most important difference is that I2S is specialized in streaming data.

A typical I2S design is composed of three elements, a serial clock signal(SCK), a word select(WS) and a serial data signal(SD). All of the signals are synchronous to the clock signal's falling edge. I2S is a single master system with only two channels for a single bus.

In the project there is a need of four different microphones. Due to the fact that the main board of use in the project is a Raspberry Pi 400 only two microphones can be used simultaneously.

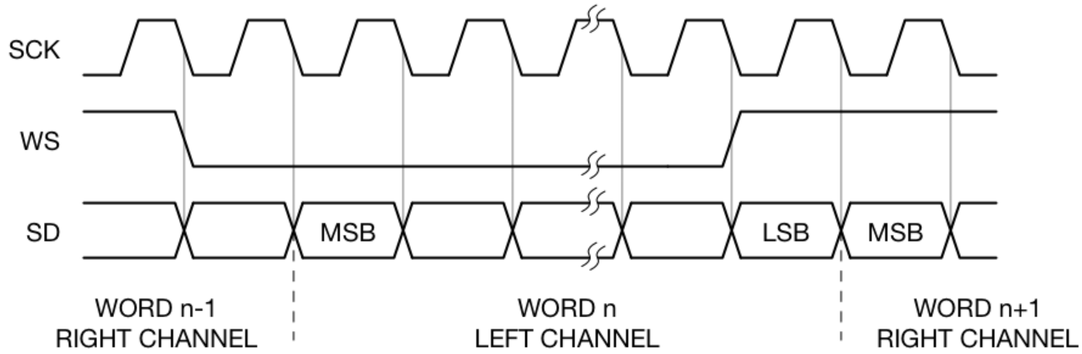

Figure 6.2 I2S Timing Diagram [10]

6.1.3 Wiring

As mentioned before only two microphones can be used simultaneously on a Raspberry environment. Two use four microphones an array of trials were made. Our first trial was using a 4x1 multiplexer to quickly switch between microphones. The SD cables of the microphones are connected to the multiplexer and the output was directed by the selection bits.

Initial trials were not clear and they were very noisy due to propagation delay made by the several gates inside the multiplexer. This situation is caused because the I2S protocol is extremely fast on clock strikes. The minimal propagation inside the multiplexer gates was simply too delaying to synchronize with the WS and SCK lines.

To fix the situation, the WS and SCK lines were connected into the multiplexers as well to delay all signals. Unlike what was projected, this combination was proved futile too. This time the sound signals were completely beyond recognition. The output was composed solely by noise.

In another attempt to fix the noise generation, a demultiplexer was used. First trials were made by cutting of the microphones' voltage supply. But the voltage from the other lines, although they were voltage low values, provided enough voltage to the microphone to emit noise to the serial data line. In essence, any electronic component with a propagation delay interfered with the efforts to gain a clear sound.

The last but the successful attempt were made by using an array of relays. Generally relays are used to control higher voltage circuits with lower amounts of voltages. In our case, the use of relays was the often forgotten property of theirs, the instant switching ability caused by their electromagnetic capabilities.

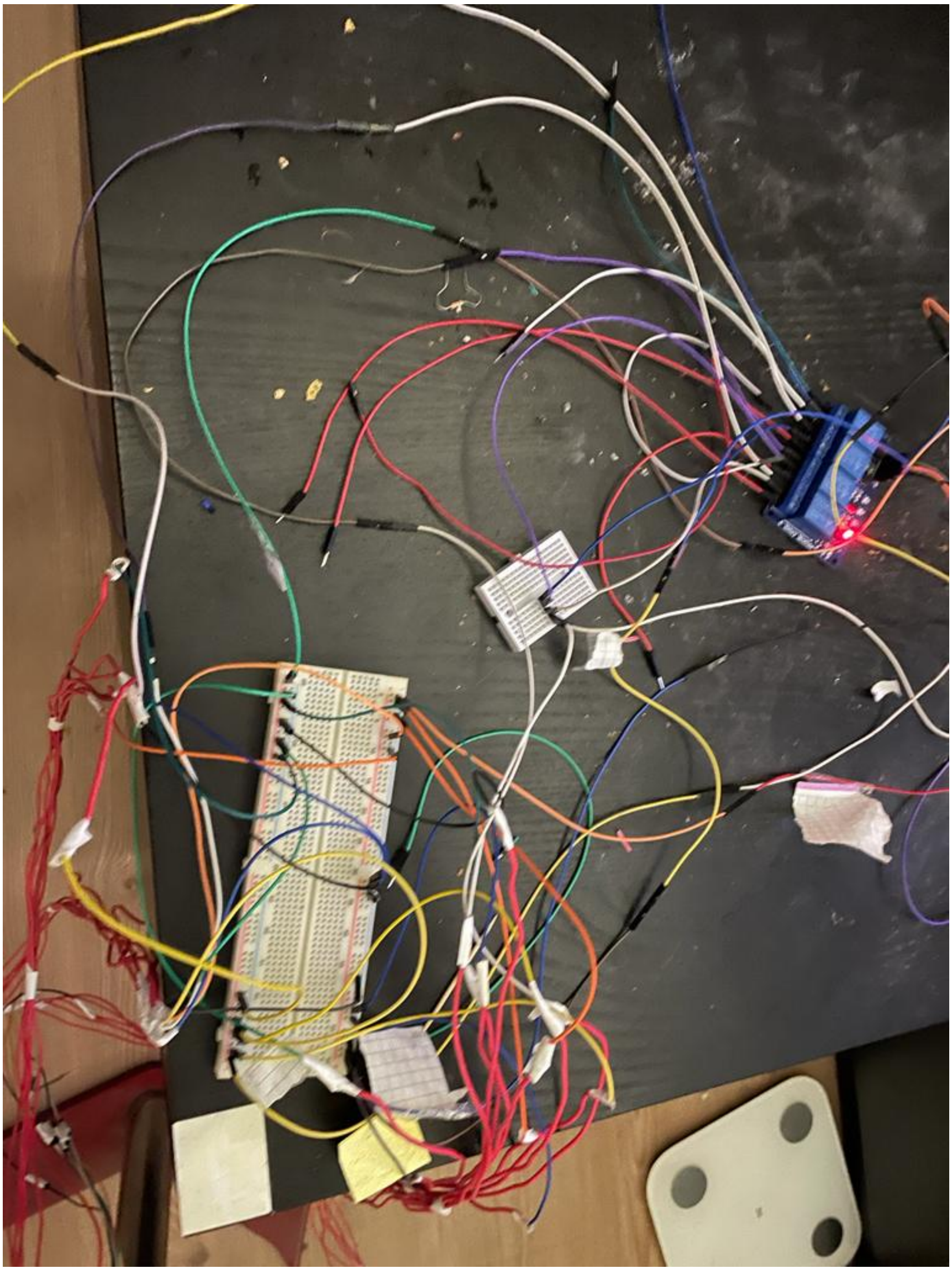

Figure 6.3 Relay Wiring on Project

The relay in use is a 5V 4 channel relay. To switch between normally closed and normally open legs, a level shifter was used but to no avail. The system did not act as planned and could not make the switch. So a different design was made. The serial data line connected to the other legs of the relays. This approach made it possible to control the outputs by switching the GPIO pins from high to low.

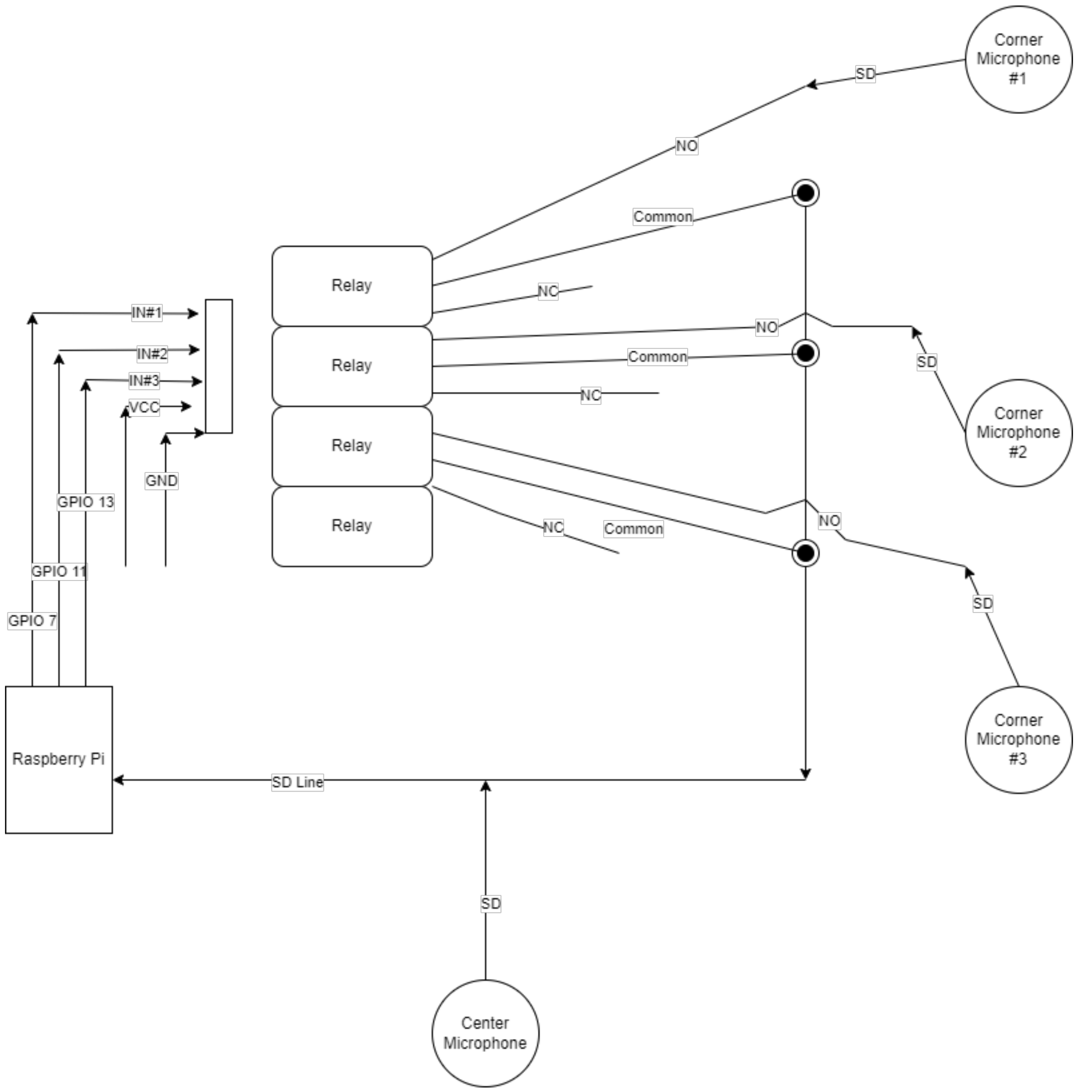

Figure 6.4 Relay Wiring Schematic

Finally the microphone could be chosen dynamically from the raspberry.

6.2 Program

6.2.1 Microphone Inputs

When the system starts running, it listens to the signals received from the first and the center microphone. At the moment when the sound signal is received by both of the microphones, a windowing operation is done on each of the signals. After windowing operations, a two layered Fast Fourier Transform is performed on the signals.

6.2.2 Time Difference of Arrival Calculation

Sequentially, two signals will be conjugated. Peak of the signal is found by a max() function. Instead of searching the whole array for the peak point, the program only searches for the peak in its theoretically probable places. To elaborate, the peak sample can only be found in a certain interval determined by speed of sound and the distance between microphones.

When both the peak values are found, the indices between are divided by the sampling frequency to determine the time between arrival. After gathering the gap between arrivals, the gap is multiplied by the speed of sound to obtain a distance regarding sound waves' extra movement to reach the further microphone.

After obtaining the extra distance it is possible deduce the angle using arcsine functions where the counter vertice is the stable distance between microphones and the hypotenuse is the extra distance made for the outer microphone.

These steps are iterated for all three corner microphones to obtain three different angles in 2 different dimensions.

6.2.3 Generating X-Z Dimensional Lines

Afterwards, two different lines are created using the previously obtained angles and their point of intersection in each of their axes. These points are the virtual representation of the microphones on X-Z dimension. Succeedingly, the point of intersection of the lines are calculated. This point is the representation of the sound source on X-Z dimension. Meaning that the distance between sound source and the center microphone as the crow flies is now available to use.

6.2.4 Calculating Elevation Levels

Lastly, using the aforementioned point it is now possible to find the elevation level of the sound source. A three dimensional line is created by the Y-angle and center. By utilizing the point of projection in the solution set.

At last, all of the properties of the sound source's location on X-Y-Z dimensions are gathered.

6.2.5 Graphical User Interface

For the user to see the results, a GUI is available on the program. User is able to see the 3D projection of sound source on the screen.

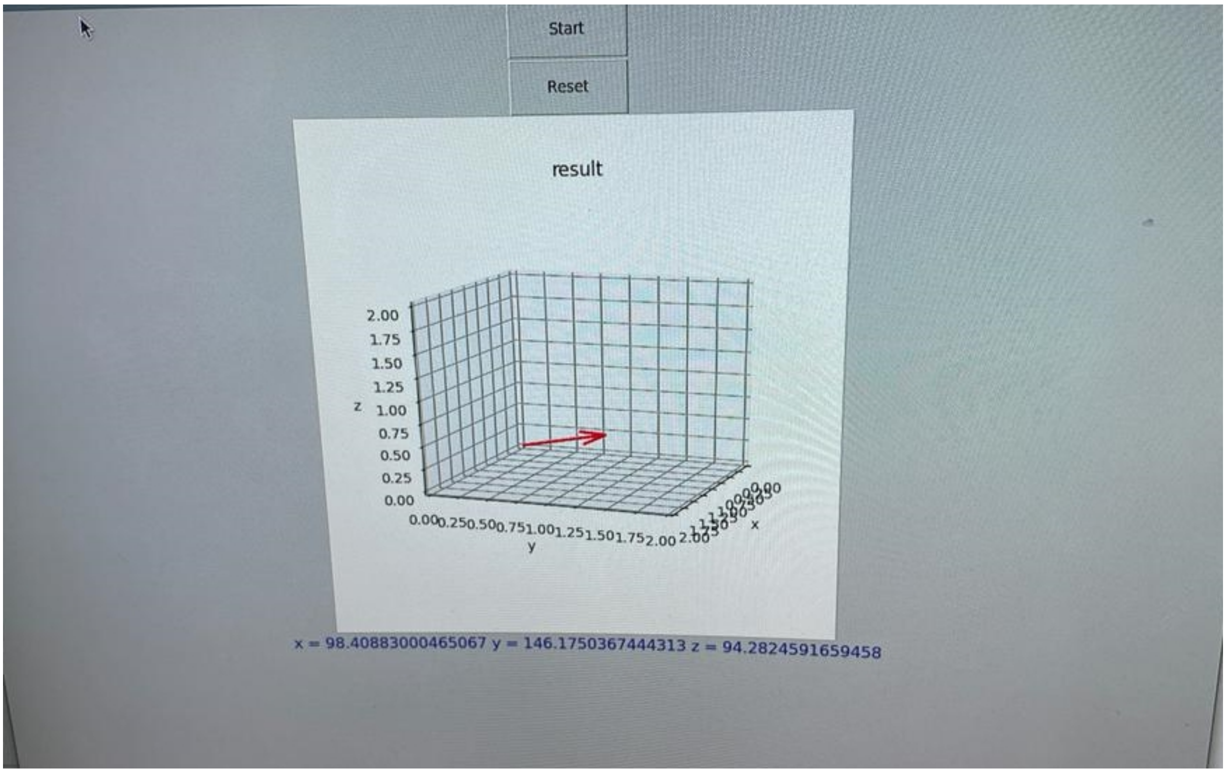

Figure 6.5 I2S Timing Diagram [10]

7 Experimental Results

Table 7.1 Angle Prediction Experiment Results

Alpha	Theta	Beta	Alpha Pred	Theta Pred	Beta Pred
134	96	126	135	94	131
119	130	120	110	139	102
134	115	137	126	132	120
121	92	141	117	95	139
101	104	155	102	114	145
115	93	140	121	93	156
165	100	105	160	103	93
160	82	112	156	92	100
112	91	150	105	93	145

As it can be seen on the table of angle prediction results, predicted and the real values differ from each other at an interval like -10 degrees and +10 degrees.

Table 7.2 Distance Prediction Experiment Results

Distance	Predicted Distance
374	271
114	102
213	170
300	195
232	114
435	236
364	87
240	159
445	124

As it can be seen on the table above, predictions are not very accurate. The cause of this situation lies on the error when predicting angles.

8
Performance Analysis

8.1 Error Margins

In the first table, the error margins between real angles and predicted angles are %5, %6.2, %10.7 for x, y and z respectively.

In the second table, the error margins between real distances and predicted distances are %46.

By examining the two tables above we can make two assumptions. In the first table, the degree of error is very minimal. But on the second table it is possible to state that the error margin is too high.

The cause of this situation is generally relevant to error at angle predictions. The error at calculating angles can have many sources. The speed of sound differs with temperature and height levels. The sound can echo from an obstacle and create a false positive. But the most probable reason we acquired is the error prone nature of making predictions on high degrees.

When the sum of A and B angles exceeds 270 degrees the lines will not intersect. This situation makes it harder and more prone to errors while calculating angles at higher degrees. Even as little as 3 or 4 degrees can alter the distance prediction by 10 or 15 meters.

A more secure way to obtain distances is to minimize error on angle predictions.

9
Conclusion

In this project, it is presented that it is possible to locate sound sources using MEMS microphones and Raspberry Pi. Additionally, a way of using multiple I2S channels on a general purpose electronic board with only one I2s bus, is presented.

After examining the experiment results it can be stated that angle calculations are performed with miniscule error. Alas, the miniscule error margins have proven to cause much bigger error margins on locating the sound source on a three dimensional space.

Although the localization phase is faulty depending on the angle inputs, the algorithm involving coordinate estimation is concise. More microphones or better algorithms to obtain less error prone angles, will increase the accuracy of the output.

References

[1] V. Willert, J. Eggert, J. Adamy, R. Stahl, and E. Korner, "A probabilistic model for binaural sound localization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 982–994, 2006.

[2] F. Keyrouz, "Advanced binaural sound localization in 3-d for humanoid robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 9, pp. 2098–2107, 2014.

[3] Y. Yang, J. Xi, W. Zhang, and L. Zhang, "Full-sphere binaural sound source localization using multi-task neural network," pp. 432–436, 2020.

[4] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform," *CoRR*, vol. abs/1904.01916, 2019. arXiv: 1904.01916. [Online]. Available: http://arxiv.org/abs/1904.01916.

[5] B. R. HAMMOND, "Methods for robust binaural sound source localisation," eng, 2021.

[6] H. B. Önen and M. Kartal, "A time difference of arrival estimation with integration of generalized cross correlation samples for continuous wave signals on passive positioning systems," pp. 293–297, 2021. DOI: 10.23919/ELECO54474.2021.9677772.

[7] M. Khyzhniak and M. Malanowski, "Localization of an acoustic emission source based on time difference of arrival," pp. 117–121, 2021. DOI: 10.1109/SPSympo51155.2020.9593909.

[8] J. Hrisko. "Recording stereo audio on a raspberry pi." (2020), (visited on 06/02/2022).

[9] "Inmp441 wiring." (), (visited on 06/01/2022).

[10] N. B.V. "Um11732 i2s bus specification." (2022), (visited on 06/01/2022).

Curriculum Vitae

FIRST MEMBER

Name-Surname: Mucahit KARABULUT
Birthdate and Place of Birth: 26.06.1998, İstanbul
E-mail: mucahit.karabulut@std.yildiz.edu.tr
Phone: 0538 460 26 06
Practical Training: Cezeri Robotik Artificial Intelligence Department

SECOND MEMBER

Name-Surname: Ömer Aras KAPLAN
Birthdate and Place of Birth: 01.04.1998, İstanbul
E-mail: aras.kaplan@std.yildiz.edu.tr
Phone: 0541 534 31 88
Practical Training: Turkcell Digital Tellco and Communications Department

Project System Informations

System and Software: Raspberry Pi OS,Python
Required RAM: 4GB
Required Disk: 512MB