* Project: sound localization
* Student name: Hamze Hammami
* Registration number: 1906232
* Supervisor: Morteza Varasteh
* Second assessor: Amit Singh

Computer Science and Electronic Engineering, University Of Essex, 2022

CE301 Capstone project Report

0.1-Table of Contents

[0.2-Acknowledgment. 3](#_Toc102080881)

[0.3-Abstract. 3](#_Toc102080882)

[0.4-Keywords. 3](#_Toc102080883)

[0.5-figures, video 3](#_Toc102080884)

[0.6-symbols 4](#_Toc102080885)

[0.7-sumary of sections 4](#_Toc102080886)

[1-Introduction 5](#_Toc102080887)

[1.1 brief 5](#_Toc102080888)

[1.2 why sound? 5](#_Toc102080889)

[1.3 goal of the project 6](#_Toc102080890)

[2-background work 6](#_Toc102080891)

[2.1 sound localisation 6](#_Toc102080892)

[2.1.1 ITD: 6](#_Toc102080893)

[2.1.2 ILD / IID: 7](#_Toc102080894)

[2.1.3 HRTF: 7](#_Toc102080895)

[2.2 projects 7](#_Toc102080896)

[2.3 ethical and safety issues 8](#_Toc102080897)

[3- testing robot 8](#_Toc102080898)

[3.1 the robot 8](#_Toc102080899)

[3.2 modification 9](#_Toc102080900)

[3.3 key components to program 9](#_Toc102080901)

[3.3.1 sound sensors 9](#_Toc102080902)

[3.3.2 Ultrasonic 10](#_Toc102080903)

[3.3.3 LCD screen 11](#_Toc102080904)

[3.3.4 servo motor 11](#_Toc102080905)

[3.4 section conclusion 11](#_Toc102080906)

[4- Woodworth’s ITD formula 12](#_Toc102080907)

[4.1 ITD 12](#_Toc102080908)

[4.2 bearing formula and testing 12](#_Toc102080909)

[4.2.1 formula 12](#_Toc102080910)

[4.2.2 testing formula 13](#_Toc102080911)

[4.4 ITD code implementation 13](#_Toc102080912)

[4.3.1 explanation 13](#_Toc102080913)

[4.3.2 ITD pseudo code 13](#_Toc102080914)

[4.4 formula error 13](#_Toc102080915)

[4.5 finding general direction 13](#_Toc102080916)

[4.6 finding the azimuth 14](#_Toc102080917)

[4.7 improved algorithm 15](#_Toc102080918)

[4.8 section conclusion 15](#_Toc102080919)

[5- navigation 16](#_Toc102080920)

[5.1 nav methods 16](#_Toc102080921)

[5.2 program method 17](#_Toc102080922)

[5.3 section conclusion 18](#_Toc102080923)

[6- UI and program 18](#_Toc102080924)

[6.1 UI 18](#_Toc102080925)

[6.2 main program 19](#_Toc102080926)

[6.2.1 sound Loc program 19](#_Toc102080927)

[6.2.2 pseudo code 20](#_Toc102080928)

[6.3 functions 21](#_Toc102080929)

[6.4 program demo 22](#_Toc102080930)

[6.5 section conclusion 23](#_Toc102080931)

[7- testing 23](#_Toc102080932)

[7.1 testing robot turning movement 23](#_Toc102080933)

[7.2 testing ultrasonic obstacle detection 24](#_Toc102080934)

[7.3 testing microphone sound detection 24](#_Toc102080935)

[7.4 testing sound localisation algorithm 24](#_Toc102080936)

[7.5 section conclusion 25](#_Toc102080937)

[8- project changes 25](#_Toc102080938)

[8.1 language and machine 25](#_Toc102080939)

[8.2 algorithm 25](#_Toc102080940)

[9- future development 26](#_Toc102080941)

[10- project management 26](#_Toc102080942)

[10.1 Jira 26](#_Toc102080943)

[10.2 GitLab 28](#_Toc102080944)

[11-project conclusion 28](#_Toc102080945)

[12 code GitLab 28](#_Toc102080946)

[13-citation 28](#_Toc102080947)

[13.1 content citation 28](#_Toc102080948)

[13.2 figures citation 29](#_Toc102080949)

# 0.2-Acknowledgment.

I want to give my many thanks to my supervisor Morteza Varasteh, who supported me through the project with many ideas, Mr mark marney and Vishwanathan Mohan who both helped to put the project in check and with the suggestions towards the robot which the project is to be tested, I also want to give thanks to many lecturers and GLAs who approached me while using the robot to give me advice on different aspects of the project from their own experience and research, it has really helped a lot with practical work on research and potential improvement to the algorithm.

I also want to thank my friends and family who helped me push through when something seemed impossible; who motivated me to do the work and find solutions when a solution seemed impossible, special thanks to my friend Abdelrahman Mansour for also helping me choose the project.

I would like to honour this project to my father who passed away 2 months before the deadline of the project; he was cheering me on throughout the project and never got to see the final result; all the hard work is thanks to his support.

# 0.3-Abstract.

Robots are machines that come in many applications and many ways for mobilizing them are discovered, sound being a useful concept to detect a person in distress or danger as they are out of sight, the concept of localising sound to be as cheap and efficient as possible with no special equipment whether it is a humanoid an RC or drone, it is to find a method to work with any machine. With that, any robot with sound sensors can direct an angle of where a sound comes from which could be used in many other applications; restaurant waiters, nurses, searching parties, police. Etc. robots they do not speak our language, robots interact with us using various coding languages and programs, one of those codes is called Arduino and the robot that uses the language is called EMoRo robot an RC car that uses a circuit board to move around and do other tasks, this machine will be used to test the speed localisation algorithm to help it find the azimuth, then with that we use ultrasonic sensor so when the first obstacle it finds it assumes as the source of the sound although for a more efficient sound localisation using sound decibels to stop when sound presser has reached high point where the target is close which it will use the sound presser and stop at predicted location of sound.

# 0.4-Keywords.

Robot, ITD, EMoRo robot, Arduino, EMoRo 2560 board, C/C++, sound localisation, azimuth / bearing

# 0.5-figures, video

Figure 1 ITD and IID, page 8

Figure 2 HRTF, page 8

Figure 3 EMoRo robot, page 9

Figure 4 ky-038 sensor, page 10

Figure 5 ultrasonic sensor, page 11

Figure 6 glam pro, page 11

Figure 7 sound wave graph, page 12

Figure 8 ITD formula, page 13

Figure 9 bearing formula, page 13

Figure 10 general direction, page 14

Figure 11 efficient algorithm sketch, page 15

Figure 12 intensity distance, page 17

Figure 13 Intensity Navigation method 2 page 18

Figure 14 Main menu, page 19

Figure 15 debug menu, page 19

Figure 16 system report, page 19

Figure 17 listening start, page 20

Figure 18 listening end, page 20

Figure 19 locating middle stage, page 20

Figure 20 locating end, page 20

Figure 21 moving end, page 21

Video 1 program demonstration video, page 23

Figure 22 QISDK simulation, page 26

Figure 23 Pepper robot, page 26

Figure 24 Kanban board, page 27

Figure 25 timeline, page 27

Figure 26 GitLab timeline, page 29

# 0.6-symbols

ITD = interaural time difference

θ = bearing of a far field source or also known as azimuth

RBM = radius between microphones

SS = speed of sound, 343 m /s

USS = ultrasonic sound sensor

ILD/IID = interaural level/intensity difference

HRTF = head-related transfer function

NN = neural network

# 0.7-sumary of sections

Section 0: includes the abstract, acknowledgment, and other factors of the report, it will also include a summary of each section in the report. (1265 words)

Section 1: brief of the project, its importance, and the end goal of the project. (445 words)

Section2: This section is the lit review of the project; this shows background work and previous research used to achieve the result of the project, looking at similar projects and talking about the concept of other mobile robots and different sound localisation research other than robots

Section 3: this section will talk about the equipment and robot used to test the algorithm, how they work, changes made to the robot-like removing or adding components and will also talk about the other machines that were potentially being used.

Section 4: talk about the ITD and formula how it works, the errors that come with it and how they are fixed in our algorithm to provide an efficient sound localisation method.

Section 5: talks about the navigation method we are using for this testing phase of the robot how it can be altered to be more efficient and other machine possible navigation methods.

Section 6: talk about the code implementation of the algorithm that was decided on in sections 4 and 5, we talk about the program itself like the graphical interface on the LCD screen and other parts of the program.

Section 7: this section is for the testing and practical work, like training for the program, algorithm testing, movement testing, testing sound sensors …. etc.

Section 8: this section will talk about major changes in the initial planning of project compared to how it is now, the reason for those changes and how it improved or helped with the progress.

Section 9: this section talk about the future of the project and how current problems can be improved.

Section 10: this section talks about project management, Jira tasks, personal management, goals set for each timeline of the project duration.

Section 11: concludes the results of the project.

Section 12: this section provides a link to the code of the program provided on the GitLab software.

Section 13: referencing various sources used throughout the project, mainly for algorithm.

# 1-Introduction

## 1.1 brief

The report will give a summary of what is to come, which will describe my journey for this project and details about the car-like robot EMoRo and how the algorithm was achieved and tested step by step. With the many difficulties faced in thought the project and the different changes in the project which we will talk about, the important matter of the project is traversing around an unknown and dark environment by finding the general direction of the sound, to create an algorithm which works on any type of robot and program, using math and sound sensors, which would help in situations like rescues by automated bots or even remote-controlled machines to pinpoint a bearing for sound source the report will also discuss other factors of the project background research, changes in the project, improvements in algorithm, future development…..Etc.

## 1.2 why sound?

sound is wave which we can’t see, it is a wave which we can hear with our ears, and without sound so many factors would not be possible, speaking with others, finding objects or even people, when a person loses their phone for example they ring the phone in order to produce a sound so that they can find the object, without the sound it would take longer for the object to be found, that is exactly why this concept is important, sound localisation doesn’t need to be a main factor of finding an object or a person, it would function as a helping method, sound is also a great method when other methods are obsolete, malfunctioning camera or environment unsuitable for finding the object, sound localisation would play a role in that when possible.

## 1.3 goal of the project

Sound localisation can be achieved in many ways, this can be using 3d microphone, where capturing the sound is much easier, however these microphones tend to be expensive and harder to install, where as a normal microphone would be much cheaper, another method would be adding different microphone for every angel and the first one to receive it is an angel but that is also expensive, the goal of this project is to find a method that would work on any robot using 2 microphones, while saying a cheaper method would be with one microphone it would be impossible to achieve sound localisation, which we would discuss in a later section, even for a robot with no microphones it would not be hard to install 2 small microphones and mount them adjacent to each other once that condition is achieved with the algorithm used in the project this would allow robot to find sound in a 2 dimensional environment.

# 2-background work

What are mobile robots? any machinery that can move is, mobile robots can vary from several types, wheeled robots, legged robots, flying robots, and mobile methods is what controls the mobility of the robot, this could depend on the sensors used for the robot, laser sensors, RGB sensors, lidar sensors... etc, one method is sound localization, the ability to pinpoint the source of the sound and then the robot would move to that source, this project uses the biology of human hearing and sound localization to simulate that into the robot and then implementing that into robotics, these functions are shown in the sub sections for 2.1, after that will take a look at similar project that use sound localization.

## 2.1 sound localisation

The sound localization concept is not limited to that of a machine or robot, sound localization is implemented in a lot of our daily life, our bodies included, as humans were able to tell where a sound is coming from using our ears, one experiment involves placing different sound sources and moving them around as a person tries to locate the sound sources, this concept of ‘spital hearing’ simulates different sound sources coming from different locations where the person is able to ‘subconsciously localize sound sources’ this is an important step as if we know how it is done we can simulate in our technologies, some experiments involved neural networks and sound data, to pick up / estimate anomalous sounds, and where they come from, using a ‘Convolutional Recurrent Neural Network’ this combines the 2d convolution NN and then followed by a “normalization,ELU activation,max-pooling,and dropout with a dropout rate of not more than 50%” this method has been used to predict the ‘distance class’ most of these experiments fall into the category of sound localisation specifically bearing localization, bearing is the angel on which a far field source is located, we locate the bearing angel by going clockwise to the zero point and at the point where the sound source is estimated is our bearing, localizing the bearing usually depends on interaural delays, these delays is theorised to how humans listen to sound, there are 2 types of delays that help us find the sound bearing, ITD and (IID or ILD), there another important cue to sound laicization known as HRTF, discussing these cues will help us understand briefly how sound localization cues helps us find the sound location between left and right.

2.1.1 ITD: Is the difference between the times each ear has received the sound, usually, if the distance is zero the sound distance is the same while if it is coming from another location there is a difference, there are many formulas for, the Woodworth formula being the most popular formula used for studies on human sound localization, however, this formula has its limitations as some formulas were updated to match the radius of a sound that is less than that of the listener.

2.1.2 ILD / IID: This cue focuses on the difference of sound intensity/level, it is used for high-frequency sound waves as these sounds can easily pass through the head and would provide a lower intensity on the other ear, creating a shadow where the sound has less presser to that of where the sound is coming from, depending on which side of the ear has a higher, the sound shadow depends on the frequency of the sound as for lower frequency sounds a sound shadow could not exist therefore making this cue not efficient for sound localization.

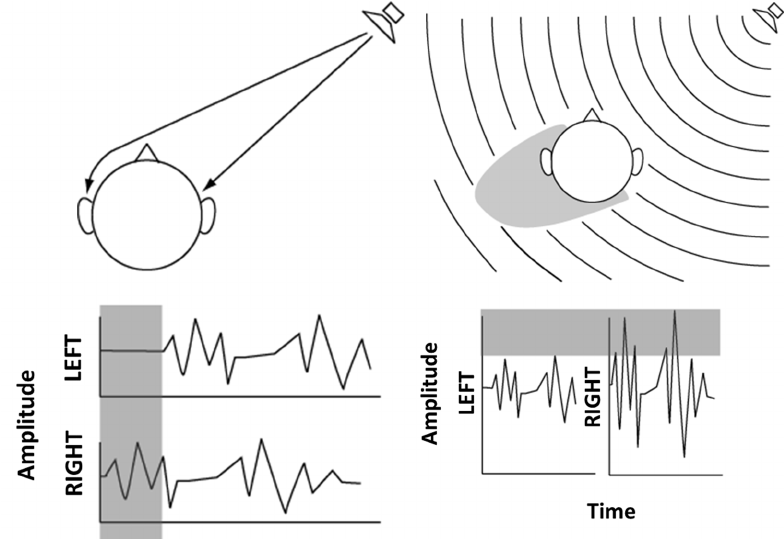


Figure 1 ITD and IID [25]

Figure above we can see the uses of sound localization using ITD (on the left) and IID (on the right).

2.1.3 HRTF: This cue focuses on studying the difference in frequency from the sound source to that of when it reaches the listener. This is different from the other cues as information about the location of sound comes from a spectrum of frequencies.

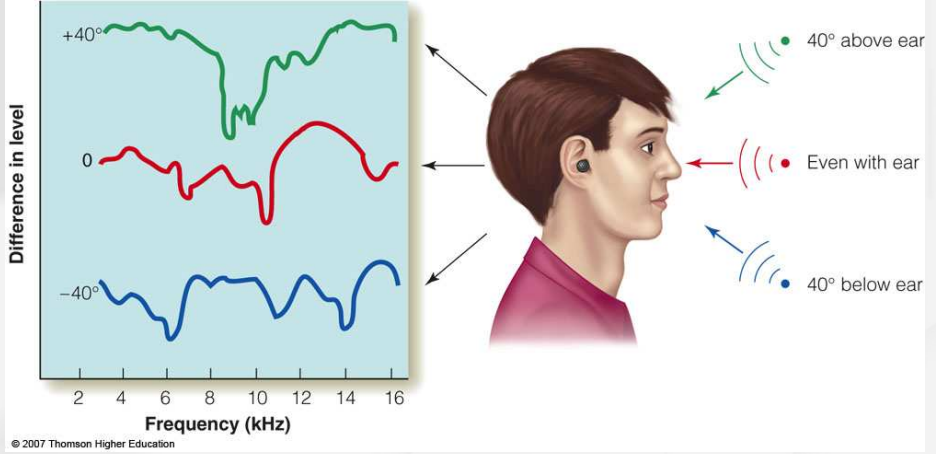


Figure 2 HRTF [5]

## 2.2 projects

We talked about sound localization and many studies related to it, how is that applied to mobile robots? We will look at 2 similar projects implementing sound localization and the research like it.

NAOqi, [10]: NAOiq is the program used to control some of the SoftBank robotics bots, in this section were going to talk about the NAO robot and Pepper sound localization, like the ITD cue it uses ‘time difference of arrival but using 4 four microphones and can get the azimuth and localization id activated when the microphone receives the sound.

Waiter Bot, [1]: This article talks about a waiter robot, using ITD the method of finding the sound source and using cross-correlation, it is to find a similarity between 2 vectors with the values of signals, which by getting the maximum length/ correlation gets the time delay, these are combined to create an algorithm for sound localization, this project uses a robot called peoplebot which can be installed with sound sensors and other components to achieve the goal of the project.

## 2.3 ethical and safety issues

Many issues with the project are to be looked at it as with the project comes a lot of issues and solves some, one of the main issues the comes with applying sound localisation for robot mobility is how safe is it, the robot could be attracted to any sound not just the object that it needs to be head to therefore leading the robot to head to completely different location, another thing is the capability of that machine to perform the task at hand, rescuing a person for example the robot needs to be able to lead a person to safety, maybe carry an unconscious person or direct rescue teams to current location for extra support.  
Another issue would be some of legal issues to face, like consent, consent can be hard to get as we can’t ask a person as consent to be rescued however, with other issues like the ‘waiter bot’ or other projects requiring sound localisation, any project we can use sound localisation will require consent for specific tasks.  
And finally, ethical issues we must consider the projects affect on the human labor that the project could possibly replace; for example, for rescue teams the robot would be more like a tool to rescue the person so they would not risk their life mapping the environment, and to direct them to rescue target safely. but for water this robot could be taking a role that could be taken by a human, and because of the project that human never got a job.

# 3- testing robot

## 3.1 the robot

The robot that was chosen to implement the sound localization algorithm is known as the EMoRo robot controlled using a circuit board, the board can be coded using Arduino, the robot itself can move around scanning the terrain below it and stopping at different obstacles that lie ahead of the robot, while it is not capable of detecting sound waves (not counting ultrasonic waves) it can be modified as we please due to the many unused input and output ports on the board, these ports include analog inputs, digital outputs, servo outputs, and other inputs, detailed look below (note that the LCD screen is missing from the figure below).

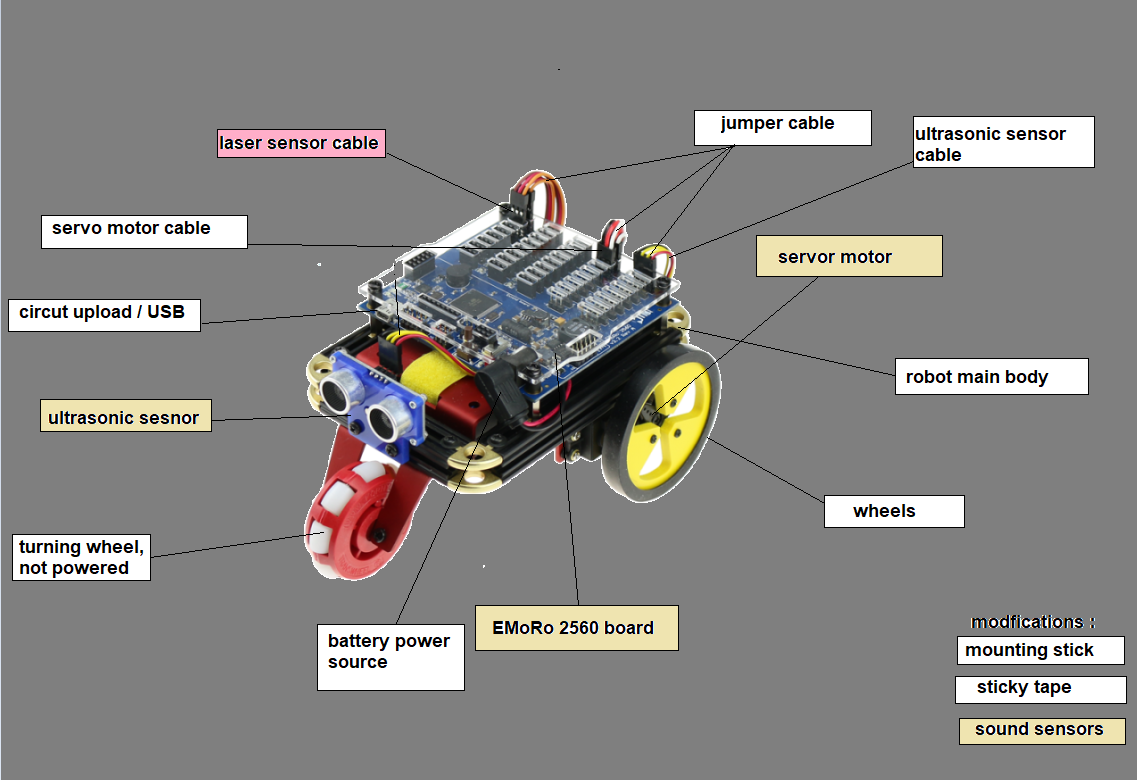


Figure 3 EMoRo robot [20]

The robot we use also provides an LCD screen which consists of 4 buttons that control the circuit and can be used as a graphical interface to ask the robot to perform the various tasks mentioned before or any tasks that we modify the robot to be capable of doing, this robot is efficient and easy to use and as shown in the graph below we can see a set of modifications which were added to our EMoRo robot to perform the sound localisation task.

## 3.2 modification

As we can see above there are different coloured tags shown yellow are key components and red are removed components, the importance of modifying the robot is to enable the robot to find sound signals, the modifications include a stick made of material that prevents the circuit on the sound sensor to be shorted, double-sided sticky tape to mount the stick and mount the sound sensors on the mounting stick, and finally, the sound sensors which is a KY-038 sound sensor, we also remove the laser sensor (not shown in the figure above as they are not visible, only the cabling ) as it is not needed to have fewer cables connected and save power consumption if needed, Although not a modification the LCD screen is missing from the figure above so it is an already existing component and will be a key component mentioned in the next section.

## 3.3 key components to program

### 3.3.1 sound sensors

The ears of the robot, KY-038 sensors are the most important components in this robot which were modified into our Arduino board robot this sensor will use its analogue signal to provide our program with data, they mainly detect vibrations and can be changed how sensitive they are using a screwdriver for noisy or less noisy environments, figure below for analysis.

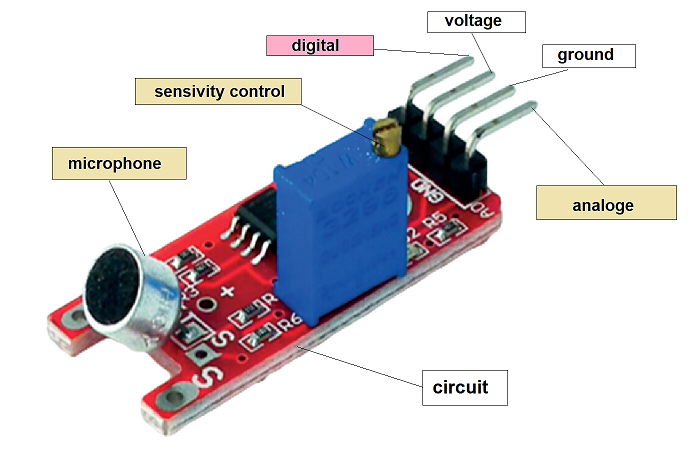


Figure 4 ky-038 sensor

In the figure above we can see the sensor in detail, the yellow screw on the blue box is a small screw used to control the sensitivity of the microphone (in the circuit), we can also see the microphone which can be modified into a different one if needed, and finally, the analogue signal output, which should send a signal when vibrations are detected indicating a sound wave, we do not use digital signal, therefore, it was marked in red in the figure above.

### 3.3.2 Ultrasonic

The eyes of the robot, this component is used to provide obstacle detection for the robot and as we are testing the sound localisation in an empty environment, we can use it to stop the robot’s movement as it reaches an obstacle as the obstacle would indicate it has reached the sound source as the only possible obstacle in an empty environment would be the sound source, in this case, the figure below for component.

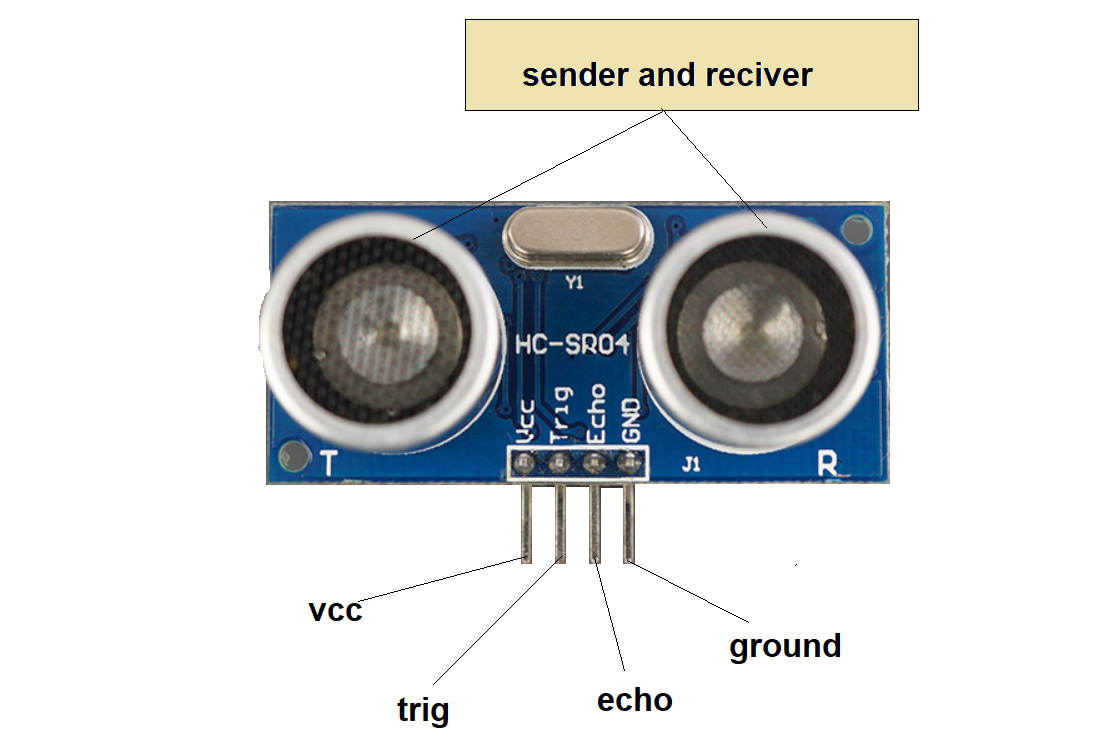


Figure 5 ultrasonic sensor [22]

The sensor uses the time of flight to find the distance between the robot and object where one sensor sends a sound wave that cannot be heard by the normal person as it is an ultrasonic soundwave, then waits for it to bounce back and receive the sound, then depending on the time it took to send and receive divided by 2 and then calculate the distance using the speed of sound and that time.

### 3.3.3 LCD screen

The brain of the robot, the glam pro, is the LCD screen provided with our EMoRo robot; this screen is where the robot will be controlled using this screen; this will also act as the user graphical interface, which will show on the screen the tasks the robot is able to perform. Check the figure below for the component.

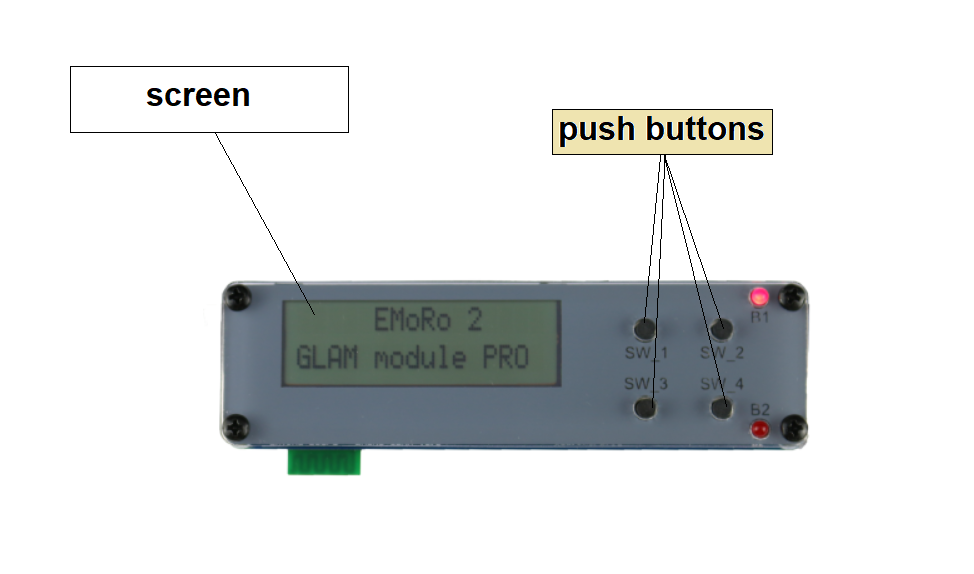


Figure 6 Glam pro [21]

The pushbuttons shown provide the user with many options; for example, let us say we want the robot to start listening for sound, the screen will provide the user with the details necessary on which button /switch should be pressed in order to perform this task. This LCD screen is important as it can also allow us to test the various stages the robot goes through in order to achieve sound localization and lets us turn on and off the various tasks when necessary.

### 3.3.4 servo motor

The legs of the robot, the servo motors are connected to a pair of wheels, one wheel per motor these motors spin around in order to perform various navigation tasks, most importantly will be used to turn around as it locates the angel of the sound and will also move the robot towards the sound source, the movement depends on each separate motor, and they are controlled individually so depending on the necessary task at hand we need to provide each motor with its turning speed and direction in order to achieve the speed of turning / moving.

## 3.4 section conclusion

In this section we were introduced to the EMoRo robot, the robot that will go through the testing phase of the algorithm; we looked at its various components and what they provide for the robot to achieve various mobile robot tasks.

# 

# 4- Woodworth’s ITD formula

## 4.1 ITD

Woodworth’s ITD formula is the most thought out and tested formula, ITD is the time difference between when the sound was received on both ends of the microphones, which gives us the time delay, let’s say I am speaking to the robot and as the sound wave is traveling it reaches the right ear microphone/ear first and then the second receives the sound, when we get the time of when the sound is received and minus them by each other we get a number, that number is the time delay.

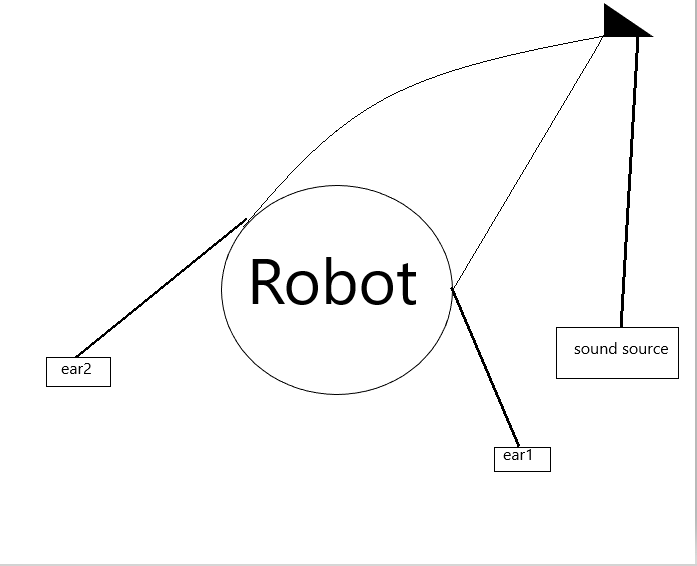


Figure 7 sound wave graph

As shown in figure one, we can see the source of the sound sending sound waves; that wave will hit the right ear and then the left ear. The ITD formula is as follows ‘ITD = (RBM (θ + sin (θ)))/SS’ the explanation for the symbols can be seen in 3.1 or 0.6.



Figure 8 ITD formula

## The formula gives us the delay between when the first microphone received the sound compared to that of the second microphone, assuming we have the other symbols. However, finding the ITD is not the goal; we need to find the θ in order to find it we need to modify the formula.

## 4.2 bearing formula and testing

### 4.2.1 formula

Now in order to change this algorithm we need to figure out which variable or set of variables we need to find in that case we need to find (θ + sin(θ)), first we move the /SS to the other side and turn it into multiplication which gives us the following RBM \* (θ + sin(θ)) = ITD \* SS, then we take the RBM to the other side and switch it to divided, ‘θ + sin(θ) = (ITD \* SS) / RBM’.



Figure 9 bearing formula

Why do we need to find the bearing? What does bearing even mean? “In mathematics, a bearing is the angle in degrees measured clockwise from north” (Bearings - Using bearings in trigonometry - National 5 Maths Revision - BBC Bitesize, 2021) which is simply described as the angle from which the sound is coming from, and the direction the robot should point in order to reach the target.

### 4.2.2 testing formula

In order to test we need to make sure that the reverse of switching ITD and theta is correct, first thing we transform our SS value from M to mm which gives us 34300, the assumed robot radius is 88.9 mm, then we use the normal formula in this test we assume the angle is 90 degrees and we need to find the ITD, ‘(88.9(90+sin90))/343000’, 8089.9/343000 = 0.02358571429, which would be our ITD, to check if the bearing formula it would be 0.02358571429 \*343000/88.9 = 91, θ + sin(θ) = 91, θ + 1 = 91, this means the formula is correct.

We can estimate our bearing by taking the value and subtracting the sin of the value to get a close estimate to the value, which would prove the formula to be efficient.

## 4.4 ITD code implementation

### 4.3.1 explanation

How the code will work in terms of the algorithm it is by listening to the sound waves on both ends of the microphone, when a microphone finds a sound wave it returns the time since we started the program and then when the second microphone gets a sound wave it does the same and then find the difference between these values to get the ITD, which would be the time it took the sound wave to travel from the first microphone to the second microphone which then we implement into the bearing formula after calculating the radius between the microphones as well which can differ depending on the robot we are using.

### 4.3.2 ITD pseudo code

1 If M(N) > frequency + 1{

2 Value 1 = micros

3 If M(N+/-1) > frequency + 1{

4 Value 2 = micros

5 ITD = value 2 - value 1}}

## 4.4 formula error

While ITD find the angel of that of where the sound is coming from using only 2 microphones it has a specific error, this formula is only efficient at doing that within 90 degrees of space as an ITD of 45 degrees bearing could be 45 bearing, 135 bearing, 225 bearing or 315 as the ITD would be the same when sound is received from those angels, same thing for if the sound is coming from the front or back ITD would be zero in both cases, and so on.

## 4.5 finding general direction

With the current formula the robot is only able to know the angel of the sound using 90 degrees of bearing from it zero point, which even if the sound is coming from the left the robot will assume it coming from the right, in order to fix that we can use positive, negative delays where in certain conditions the positive and negative value will tell us if the sound came from the right, the left, or if the value is zero (or close to zero) then the sound is perpendicular to the robot (front or back), assuming the following conditions: we find the ITD value by subtracting the left value from the right value where right – left = ITD, then the results would vary depending on the sound source, if the sound is coming from the left that means right > left and right – left = positive, that means the sound is coming from the left since the left value is smaller, if right < left and right – left = negative, then the sound would be on the right since the left value was bigger, the smaller value always wins since it reached the microphone first and took less time to reach, when right – left = 0 zero this could lead to 2 outcomes either bearing is zero or 180, since the sound reached at the same time there was no delay and the sound is perpendicular to the sound, see figure below.

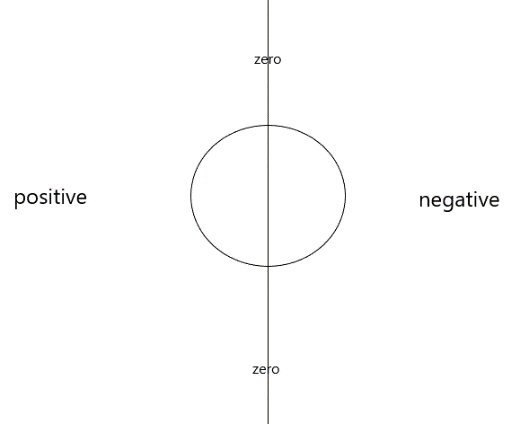


Figure 10 general direction

Using the negative and positive value we can now cover 180 degrees bearing of the robot starting from its from zero to 90 and also from its zero to counter clockwise bearing of 90 , however there is still a problem where the robot could receive a sound from the back and mistake it for front value, not just with zero delay but can also happen with left and right value where if a sound is coming from 135 degrees the robot would mistake the bearing for 45 degrees instead, since both produce a negative value because they are both on the right of the robot while facing the front the robot will assume the angel to be at 45 degrees to its current zero bearing, which means this method is only efficient to find the general direction of the sound.

## 4.6 finding the azimuth

Finally to improve the algorithm we use double values where the algorithm will work 2 different times changing its bearing depending on the result of the first wave where it has 3 outcomes, 2 of those out comes will lead the robot to move 90 degrees to the right, which is negative and zero, if the outcome is positive it goes 90 degrees to the left from there the robot will need a second sound wave to be sent to determine the final location of the sound, from there we have 3 possible outcomes for each initial outcome for the positive and negative values if a zero delay is detected in the second outcome then the sound source is confirmed to be to the front and of its current position and it would start moving to sound source, if we get a duplicate value to that of the first value then we find the angel using ITD and moves clockwise at the bearing that we found which indicates the sound source was to the back right / left of its initial position, if the value is the reverse value of the initial value ( example, first value positive and second is negative ) then we also find the angel and it moves anti clockwise and indicates the target was to the right / left front of its initial position, however when we get a negative delay in the second process we need to change the value back to positive before we initialise it in the ITD formula, since a negative delay is not physically possible and in the equation delay should only be positive and negative and positive values only indicate where the general direction of the sound to make the algorithm more efficient, finally for the zero initial outcome, if a duplicate is detected, it assumes an error or multiple sounds since the object that made the first sound can’t have a zero delay a second time without changing positions or if a second similar sound source sent the second sound wave, since the robot turned to the right we use the same second value (which wasn’t mentioned as both right and left were described together) to indicate front or back, so if negative then sound is coming from the back and the robot will turn 90 degrees clockwise, else it is positive and the robot will go 90 degrees anti clockwise, in all initial outcomes if the doesn’t receive a second sound wave then it will time out ang go back to listening for a first sound wave, graph describes algorithm efficiently however values are flipped to that of the report since conditions are different where ITD is found by subtracting right from left.

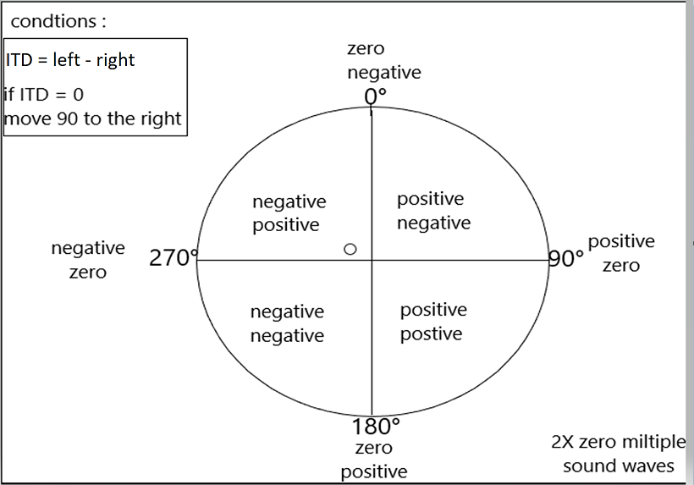


Figure 11 efficient algorithm sketch

Using double values our algorithm is now efficient to find the angel within 360 degrees, however, it does take 2 stages to do that, and the minimum and maximum turning bearing is 11 - 90 degrees and will not require turning more than 90 and the front perpendicular range will be between -10 to 10.

## 4.7 improved algorithm

With our current algorithm we a require a 2-stage process for the robot to do full sound localization, with only one stage it would be faster and more efficient for the robot to locate the sound source, to do that we need an indicator, a way for the robot to know if the sound is coming from the front or the back. a possible way to do that is by using an already existing algorithm that is used for sound localisation which is called IID or interaural intensity difference, this algorithm uses the intensity of the sound instead of the time delay where it find a general location based on where the sound is more intense , if it is more intense on the right then the sound is to the right if it more intense to the left the sound is coming from the left, we can develop a similar method to indicate front or back, conditions : we need to have one microphone that gets the same sound intensity if it is front or back, then have another microphone that is blocked by a small obstacle to make the sound less intense if it is coming from the back , what will happen then if the sound is coming from the back the intensity reading would be different, the microphones also need to have similar intensity reading when it is coming from the front to indicate that sound is coming from the front, we can do that by either adding a third microphone to the front so that when it gets a reading from the front the sound intensity is high so it is to the front , if intensity is low then it is from the back, other option is to add an obstacle to the back of one of our existing microphones to lower the intensity if the sound, not block it if the sound is blocked the whole algorithm fails, this step would provide the robot sound localisation using one stage, as it uses the same algorithm from 4.5 with this added step to indicate if the sound is for example 35 degrees of 0 bearing or 125 of 0 bearing.

## 4.8 section conclusion

in this section we looked at ITD how we improved it to work into our algorithm, and how we can improve it with IID, although the algorithm in 4.7 is more efficient it will not be used in the testing phase of this robot and the testing will be concluded using the algorithm in 4.6, it is also good to mention that the algorithm in 4.7 might not work and could also cause other parts of the algorithm to not function efficiently this information is known during possible testing phases in the future.

# 5- navigation

## 5.1 nav methods

find the distance based on receiving sound only, we can however improvise with other methods based on sound or other methods to navigate, how efficient our method depends on its environment and the type of robot we are using, a method to use is using a camera and machine learning when the robot sees itself to close proximity of the sound source which would be trained to know the possible sound source, like a human in distress or variety of objects the robot would proceed to move towards the object, avoiding obstacles and once the source is visible through the camera to be close the robot it would declare reaching the sound source, a method like this would be better used In a less dark environment as a camera would struggle to find different objects in a dark environment. Method two sound decibels, using sound intensity depending on how intense the sound received, it would decide to stop or keep moving; the problem with that is, how far the sound is, does not depend on its distance; the intensity could be the same from 2 different sound sources. But one is farther; the other is nearer.

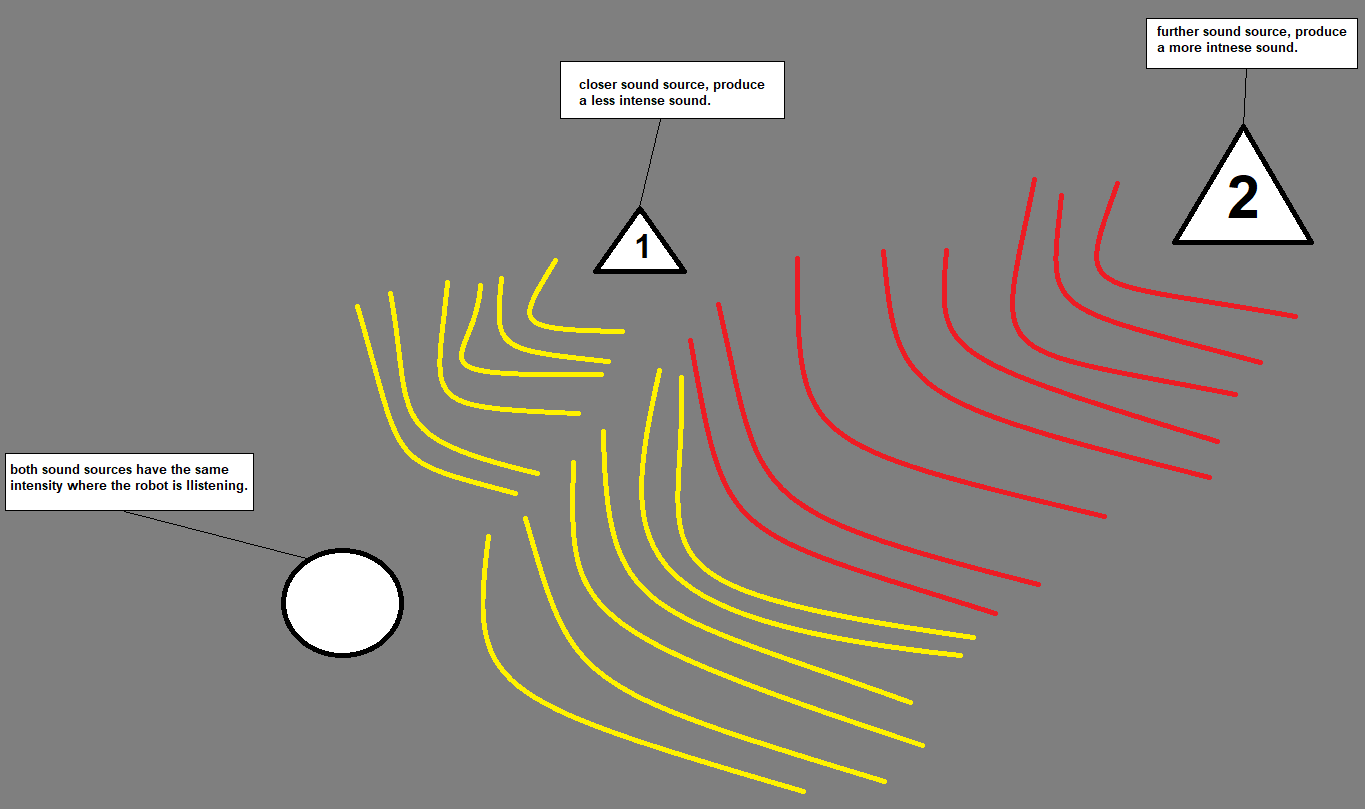


Figure 12 intensity distance

As we can see in the figure above there are two sound sources numbered 1 and 2, number two provides a louder noise, but the sound intensity is recoded to be the same as number one which ides a less intense noise, this could cause confusion for the robot as it might pass the sound or in other cases with more intense sounds stop before reaching it, there are 2 solutions to adapt to this method, one we could set a stopping intensity / decibel for the robot to stop at and then use that sound that the robot would recognize and would know to stop at that specific frequency, another solution that requires more time for the robot to reach the sound but would work with various intensities, however, this method will cause the robot to pass its source and then go back, as it gets closer to the source the robot is keeping track of its intensity readings and the highest intensity recorded, at the same time keeping a record of its path by either mapping around its environment or recording its trajectory, once the robot gets a lower intensity recording compared to a highest intensity it would record the trajectory where the highest intensity point was and back track to that position using it previous path and the mapped out environment, for this method a robot however requires multiple internal sensors for calculating trajectory and lidar sensor for mapping the environment, it would also require a constant ongoing sound, the more often sound waves are sent, the more accurate the trajectory of the robot would be, this could also be added with obstacle detection, however the robot could sometimes go back through a dead end and could also cause confusion to the robot, this method is efficient for most environment however the section solution is not as efficient in maze like environments due to the robot having to backtrack in certain paths, even if a robot has knowledge of the map before and the environment could change sometimes.

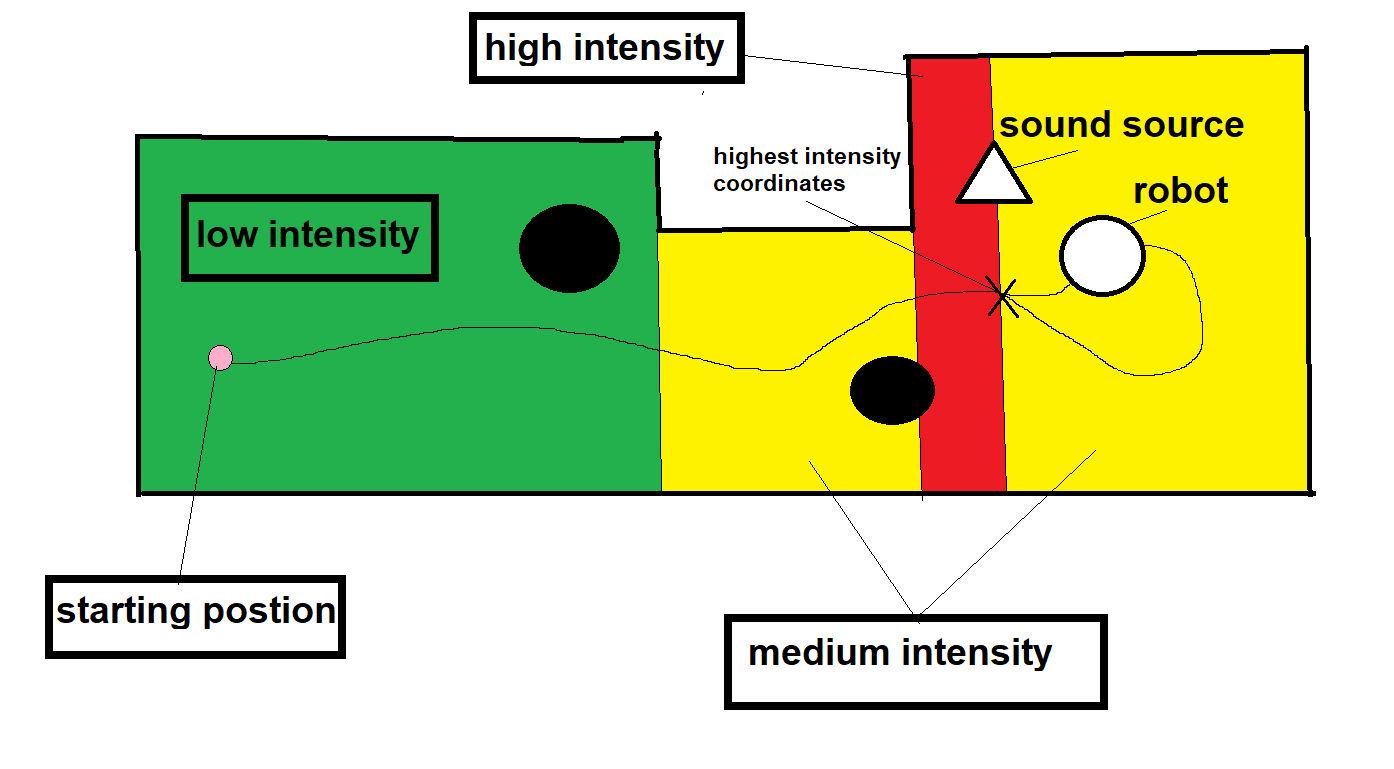


Figure 13 Intensity Navigation method 2

Method three remote operator is simply remote-controlled by a human, in a rescue situation human operators are more efficient, but why would we need to know the angel of the sound the source if the robot is being controlled by a human anyways; it is not easy for a person to find the source through a camera and 2 microphones; the robot being used won't provide spital audio with 2 microphones, using our azimuth algorithm as the robot is navigating its keeping track of azimuth of the sound and indicating it to the machine operator with each sound received then the operator would navigate the robot with that information to the sound source.

Lastly, method four object detecting, the simplest method to use for robot navigation robot will simply stop moving at the closest target to its proximity sensor; this would only work in an empty environment.

## 5.2 program method

For this project, I decided to use the last method since our testing environment is assumed to be an empty no obstacle environment, and this method requires the least amount of equipment that was not available to our robot, as previously mentioned intensity reading did not work out with the sound sensors used on our robot, for all those reasons the least efficient method was used although it is most convenient for us as our environment is empty and doesn’t have and obstacles for the robot to avoid, and lastly, the machine learning method is also not possible due to equipment for a camera not being available for our robot, in a later section it is mentioned that pepper the humanoid robot by SoftBank robotics was the robot initially planned for this project, pepper provides a lot of the missing equipment and would've used a different method if the change from pepper to EMoRo was not made, this method relies on the sound sensor implemented on the robot to detect and obstacle which would be assumed to be the sound source, even in empty environment we can't always assume an obstacle is our target as it could pass the target and hit a wall.

## 5.3 section conclusion

In this section, we looked at many methods used for navigating around environments to reach the sound source after finding the azimuth of the sound, the different methods can be used in different environments, we then talk about the method we use in our program and why we did not use other more efficient methods, in general, this section extends on our algorithm, putting the final steps in check as mentioned in previous section the algorithms and methods might not work this information will be known when they are tested.

# 6- UI and program

## 6.1 UI

The program has many, tasks and functions, we can control these functions normally, however for a user needs to run these tasks, the interface is written using Lcd screen with lcd print, using lcd prints the Lcd screen tell the user which switches do which tasks, when the program first start we get 3 options for switch 1-3, switch 1 runs the sound lionisation code which will be explained later, second option is report which comes up with 2 other options, sound report, which give us a reading on the analogue signals in the sound sensor, this will help us calibrate how sensitive the sensors are to adapt to different environments and to test the ultrasonic sensor and see how all these readings work, second option for report is a system and theta, it shows the time since the program have started in micro seconds and if we run the main program before looking at the report it will give us the last bearing the program found this bearing will show the assumed general location of the sound source since the last test, top right, bottom left ...etc if the program didn’t run it will show ‘nan’ , and lastly debug, this debug would normally not be given to a user, we get 2 other options angel turning and modes, the angel turning is tested how accurate the turning speed to the delay in order to perform an accurate turn, and then debug modes lets us debug each stage of the main program to make sure they work properly, in every sub section of our options we use the 4th switch to go back to the main menu.



Figure 14 Main menu figure 15 debug menu figure 16 system report

## 6.2 main program

### 6.2.1 sound Loc program

Main program’s function uses the algorithms in 3.5, 3.6 and 4.2, there are 3 stages for the program, listening, locating and moving. when starting the program, we are shown different prompts to changed calibrated numbers for reading where the program will run based on these numbers

#### 6.2.1.1 Listening

This mode relies on 4.5 where we use the negative and positive value, in order to find general direction, where ITD = m2-m1, when this mode runs, the program will show a stage number indicating it is waiting to get sound wave from both microphones, if the stage Is at 1 no sound was found,  and if stage is 2 then one sound is received, once that disappears, the lcd screen shows a prompt indicating the sound location, and will then to move to the sound, if the ITD is positive, then it goes to the right, else if it is negative then sound is to the left, and then it will move to the next stage.

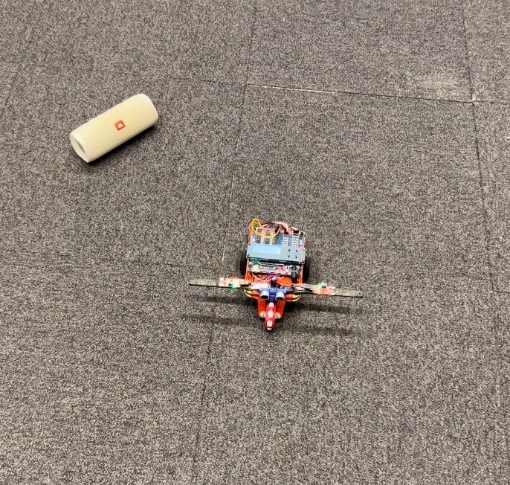
 

Figure 17 listening start Figure 18 listening end

Above we can see the robot starting position at listening and then where it ended up 90 degrees to the right and closer to the speaker.

#### 6.2.1.2 Locating

Based on 4.6 this mode is like the listening method as it uses the positive and negative but the delay of its movement is less in each stage this process uses ITD not efficiently as first planned , this stage is meant to use the ITD formula to find the angel but the formula failed in the testing phase of the program, same as listening stage number is shown to indicate how many microphones got the sound or which microphone is waiting to get the sound, if the ITD formula worked the program would instead of showing just the sound location it will show the bearing and the time delay on the LCD screen, then the robot will move on the angel and the direction of the robot will be based on the negative or positive value, due to ITD not working efficiently the vary depending on the ITD but doesn’t get the exact angel, the robot keeps repeating the process till the obstacle In front of it is found, the robot keeps trying to fix its position to the sound source and once it does it moves to the next stage, it finds the object using ultra-sonic sensor and time of flight method.

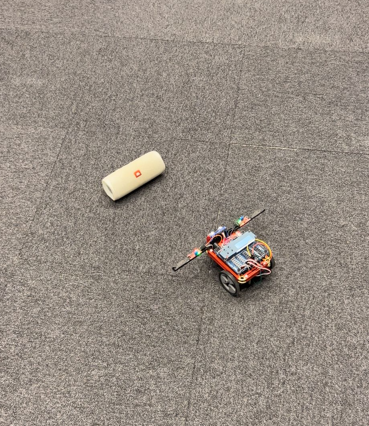
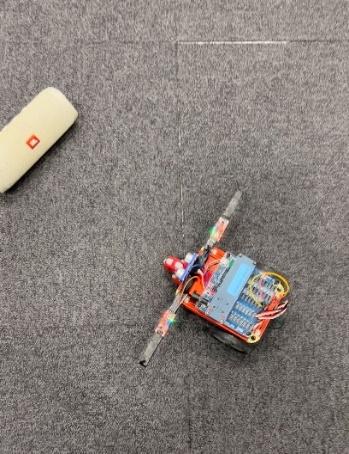
 

Figure 19 locating middle stage Figure 20 locating end

First figure is the robot in one of its stages as it tries to get a direct position at the speaker.

#### 6.2.1.3 moving

Using algorithm in 5.2, after moving from the previous stage the robot has certainly found the bearing for the sound, the robot will start moving forward, since we are testing on an empty environment only obstacle it will face is the sound source, using the ultra-sonic sensor readings the robot will detect obstacles in front of it, once it has reached that location it will stop moving in front of the obstacle and will show a prompt indicating target is reached, the robot will stop when reading goes below 20 digital signal but can be changed before running the program.

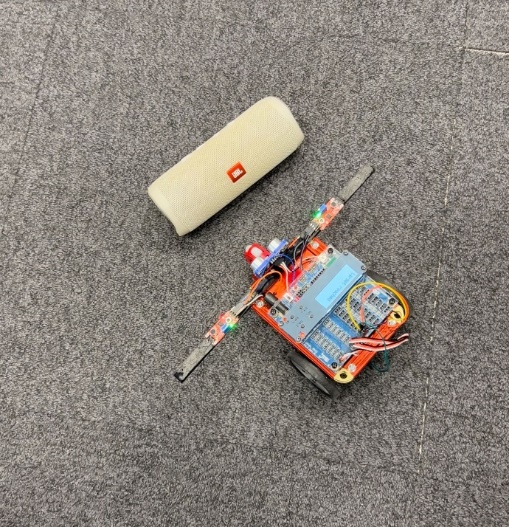


Figure 21 moving end

This is where the robot stop at after moving at a fair distance to the speaker.

### 6.2.2 pseudo code

#### 6.2.2.1 Pseudo Listen

1 If m1 or m2 is above calibrated number

2 m1 or m2 value = micros

3 If other microphone is above calibrated number

4 Other microphone value = micros

5 Itd = m2 value - m1 value

6 If itd > 1

7 Show prompt: sound location right

8 Turn 90 degrees right

9 If itd < -1

10 Turn 90 degrees left

11 Show prompt: sound location left

12 Else = error

13 end

#### 6.2.2.2 Pseudo locate

1 If m1 or m2 is above calibrated number

2 m1 or m2 value = micros

3 If other microphone is above calibrated number

4 Other microphone value = micros

5 Itd = m2 value - m1 value

6 If itd > 1

7 Turn N degrees right , N is based on ITD

8 If itd < -1

9 Turn N degrees left , N is based on ITD

10 repeat once

11 if ultrasonic reading > assumed distance

12 repeat again

13 if ultrasonic reading <= assumed distance

13 end

#### 6.2.2.3 Pseudo move

1 Move forward

2 If ultrasonic reading < calibrated number

3 Stop moving

4 Show prompt: target reached

5 end

## 6.3 functions

The code contains many functions these functions help us to the tasks in the program without cramping the main code; these functions include the following:

1 Void setup (): this our setup function where all the readings from the circuit are activated.

2 Void th\_report (): this function shows us the previous theta and time since program started.

3 Void sound\_report (): this function gives us reading on microphones and ultrasonic sound.

4 int itimed (int calibration1 , int calibration2): this function finds the ITD and is used in function 5 and 7.

5 int listen\_hc (int lmd): this function performs the mode ‘listening’ however it is hardcoded using ITD data that the user feeds to the program (not in main program).

6 voide locate\_hc (int imd) : this function performs “locating” with hardcoded data.

7 int Listen (int calibration1 , int calibration2): this function performs the mode ‘listening’ however it is not hardcoded.

8 int locate (): this function performs the mode “locating” without hardcoded data.

9 void move\_uls (): this function performs the mode “moving” no version of this node was used with hardcoded data.

10 void autocalibrate(): this function automatically calibrates the “calibrated” number to know the needed numbers for the program to work

11void calibrate(): this is a calibration function that is calibrated manually using button to change the value of calibrated number and to change which microphone needs to be calibrated.

Ps: the manually calibrate function is less error worth as the automatic can calibrate to a higher numbers if the microphones were touched during the process, however if not disturbed automatic is faster than manually.

10 void loop (): this is the main function that runs all program-related matters like the Uie, and it calls the function above except function 1 are used on it.

## 

## 6.4 program demo

[video demo capstone project](https://youtu.be/Z77NfRfk1zg)

[](https://youtu.be/Z77NfRfk1zg)

Video 1 program demonstration video

This video shows 3 demos, one for the first draft with sound localization going in all directions, one where the sound localization fails due to microphones not being calibrated properly and lastly a demo of the final program with obstacle detection on the ‘locating’ stage.

## 6.5 section conclusion

In general, the program provides an efficient 3 stage sound localisation with simple UI on the lcd screen for the user to use, that lets the user manage different tasks in the robot program, providing with checking the robot’s readings and data based on sound localisation method, which provides efficiently for the task at hand, we also looked at the different function and their role in the program.

# 7- testing

In general, the program provides an efficient 3 stage sound localisation with simple UI on the lcd screen for the user to use, that lets the user manage different tasks in the robot program, providing with checking the robot’s readings and data based on sound localisation method, which provides efficiently for the task at hand, we also looked at the different function and their role in the program.

## 7.1 testing robot turning movement

One of the aspects of the robot is moving around and turning, it is important for the robot to be able to turn around so it faces the sound source, the robot wheels are each controlled separately, the base value for the servo motors would be 1500, when given this value the robot does not move anything above that value will cause the robot to go forward and anything below will cause it to go backward, the higher the value the faster the motor will spin therefore accelerating the speed of the EMoRo robot, so if we need the robot to go forward we need to change the value, according to which wheel we control, the forward turning will different from one wheel to other so to go forward the value will be different, example (wheel 1 = 500, wheel 2 = 2500) to turn the values should be equal but above 1500 to turn to the other direction it would be below and to go backwards it would be the opposite of the example shown for forward.

The wheel that goes backward is the direction the robot is heading. The turning testing program written to test the turning helps us by finding the value written on the ‘delay ();’ function to turn at the angel we need, this testing phase was mostly trial and error and changing values, before figuring out the program the thought of turning more than 90 degrees as possible the delay was approximately 2000, and we divide that number by 360 and multiply it by the angel, however after writing the program the delay was changed to 550 since we only need to turn around 90 degrees, 45 degrees 275 and if an extra stage is needed would be 45 and 275 divided by 2, when the robot starts turning it will stop its movement after the delay by resetting the motor value to 1500.

## 7.2 testing ultrasonic obstacle detection

During the moving stage, the robot needs to detect obstacles to stop its movement, stopping the movement process to assume the robot reached the sound source, will require obstacle detection to stop the motors, the ultrasonic testing program for this testing phase doesn’t only look at obstacle detection, it also looks at obstacle avoidance, a concept that would be useful in a different environment and when the moving algorithm used is different (section 5.1), as explained in section 3.2 about the time of flight and ultrasonic sound sensor, the value that the sensor sends back indicates how far an object is if the value is 400 there is no object detected or the object is too far for the sensor to detect, the closer the object is the less the value that the sensor returns, the program consists of loop where the robot is moving forward, while the value of Ultrasonic sensor is below than 50 the robot will make a turn, till the objects distance returns a value above 50, the obstacle detection, is not efficient as it would require a better sensor that would detect objects at 360 degrees while the ultrasonic sensor only detects objects at the 0 degree which is in front of the robot.

## 7.3 testing microphone sound detection

In order for the sound localization to work, the sound detection of the microphones need to be installed in 3.3.1 we can see the different ports needed to install the microphone, voltage goes in the middle of the analogue port, ground is installed to the left of the port and then the analogue goes to the right, we install the second one as well and now the microphones are ready to go through tests, the sound sensor testing program was an attempt to test sound localization but failed at doing so, and was instead tested on the main program, other than the failed sound localization stage, the robot is meant to print the readings of the microphones, the sound sensors detect sound by detecting vibrations, even if there is no sound sometimes it might detect something when the code is running before turning on the servo motors, the sensors don’t detect anything, however, the sensors give off different readings after they are turned on as the motors vibrate, this problem cause the biggest issue in sound detection as it was required more intense sound for the sensor to detect anything, and we also needed to keep changing the number at which the sensor will stop listening, where the number would be the max readings that the sensor returns plus one, this will ensure that once the sound wave is sent at a high intensity the robot will detect the sound and the algorithm will start to work itself out from there.

## 7.4 testing sound localisation algorithm

The process to test the robot algorithm went really good at first, the robot was responding to negative and positive values and going to their perspective locations, once the robot moved it is supposed to then find the angel using the ITD value, this process failed as the ITD was always different even when sent from the same location, as for the reason it was not clear to me but was predicted to be a program error and was not sure of a fix, the ITD formula is not something I should rely on alone, as it is not reliable in its current form, so the testing was changed by making the locate stage go on a listen loop using ITD to determine how much it needs to move but not directly using is as before, and then using the ultrasonic sensor to detect if sound source is in front of the robot, which has shown to be working other than the times where a big ITD is given and the robot spins 9 times before stopping, this issue was fixed by making the ITD be set to a fixed value if it was determined that it has reached a high number.

## 7.5 section conclusion

The testing phases helped by testing aspects of the program before adding them to the main program; this helps to individually test parts of the project before adding them to main program or to fix bugs without messing with other parts of the program.

# 8- project changes

Throughout the year the project has gone through some changes, through the program and the machine used for the project; these changes affected some of the progress but also helped the progress; without these changes the achieved result would be more difficult to reach.

## 8.1 language and machine

The machine that the project was planned on first was supposed to be the pepper robot, using the android studio program with the QISDK program, as I had some experience with java, I was confident I would be able to use the program, however, I faced many difficulties with the program, as I do not have access to the pepper robot all the time, I wanted to use the simulation however the simulation did not work on my laptop and was buggy in the labs, on top the pepper libraries were not installed, the QISDK program is new and learning the program showed to be a challenge and time consuming, even when trying to test some code online it would not work on the university lab program or my own computer, the other choice I had with the pepper was to use the choreograph program, this program already has sound localisation functions, in the process defeating the purpose of the project and making custom function is more complicated, so I made the decision to change programs and robot, which is what is being used for the project currently, Arduino was easier to learn as it has so many courses and applications that already use it, ENoRo robot was also easier than the robot and I get to use it freely without a simulation, it is also a modifiable so we can remove and add components into the robot, pepper In general is a more advanced robot, again defeating the purpose on the project for using an algorithm that works with any machine.

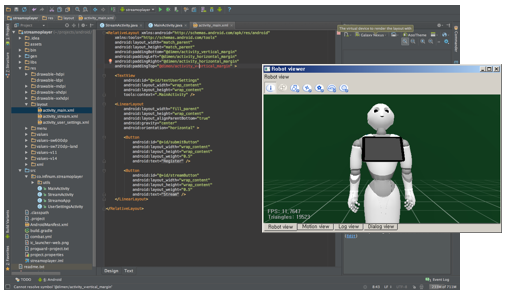
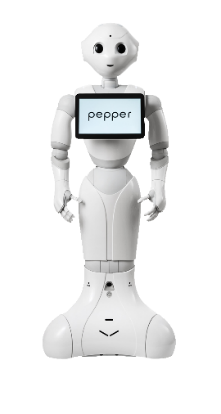
 

Figure 22 QISDK simulation [24] Figure 23 Pepper robot [23]

8.2 algorithm

A major change in the project was the change in algorithm, specifically the 'locating’ stage of the algorithm, the algorithm discussed in 4.6 uses the ITD to find the angel however this process failed during the testing stage and was changed to use the ITD but the robot will not confirm that it reached the sound until the ultra-sonic sensor sees it in its view combining 4.6 and 5.2 together to create an algorithm, although this change affects the efficiency of the program it helps by giving a temporary solution to prove that the algorithm would work fully with some changes to the program.

# 9- future development

If this project were to be extended further, the things that needed to be worked on to improve the project would be as follows, the first thing is ensuring the ITD algorithm is working efficiently with the program, as of now, the ITD formula does not work properly, as the plan was to get it to work properly but due to some error in the program it did not, second is improving the algorithm as in 4.7 and 5.1 the thermotical algorithms mentioned in those sections could improve the navigation of the robot and how fast can the robot determine the sound location, lastly testing the algorithm on other forms of mobile robots, flying robots, humanoid......etc to further prove the efficiency of the algorithm. and finally working out better theories for the algorithm to further improve on the current theory.

# 10- project management

The secret to a project’s success is good management, project planning, and adaptive thinking thanks to all the tools provided we can see the steps taken for the project and how it started.

## 10.1 Jira

The Jira software helps by setting tasks, stories, epics on the website where contributors can see and edit in this case me being the only contributor, and supervising members can give feedback on the set of tasks and other aspects of the project, like meetings and demonstrations, there are is 45 contributions in total, any set in backlog describes important details about the project and includes the references for background reading, selected for development and progress usually is in the working stage of the project and once it has been done it is moved to the ‘done’ section, these section can be seen on a Kanban board and timeline of contributions on a cumulative flow diagram.

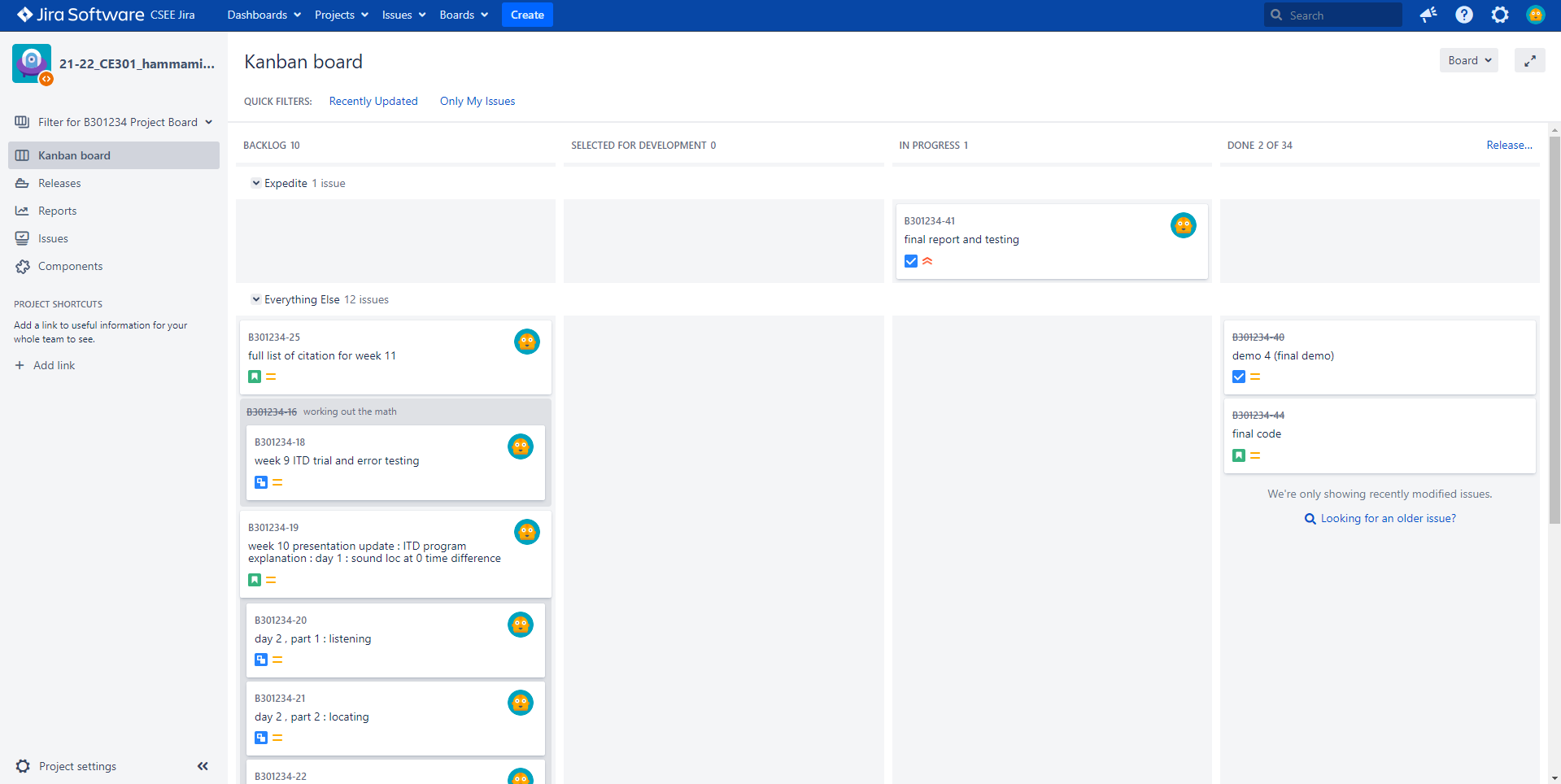


Figure 24 Kanban board

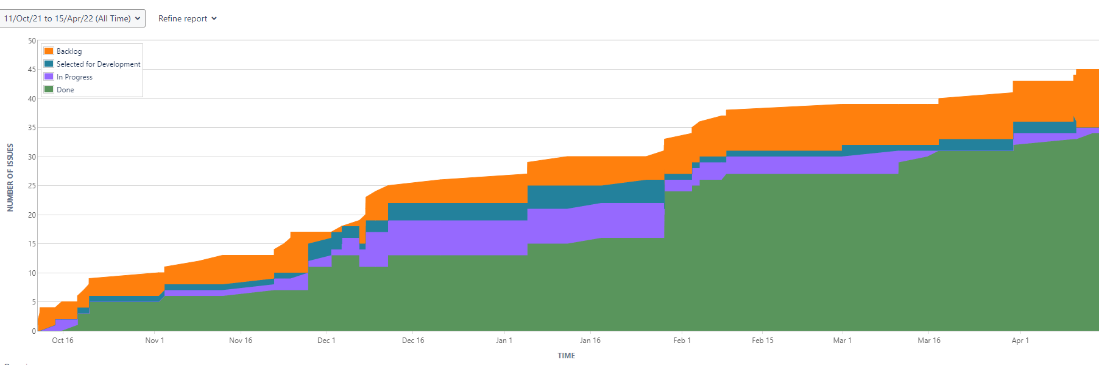


Figure 25 timeline

Challenge week: this week focused on the start-up for the project, background research choosing the robot and the program that will go through the tests looking at similar projects, 5 contributions.   
  
Pepper robot: before using EMoRo, I also had to choose which program I needed to use, the pepper was decided to be the robot however I had to choose between ROS and QISDK on the robot, it was decided to use QISDK but not much progress has gone into it as the system was buggy and the project wasn’t really getting much progress at that stage, this caused a huge setback in the project as I wanted to test the robot and the program before looking at the math behind it, 8 contributions.   
  
Math and IOE: after the setback I had to act fast to get some progress with the project, so I decided to work on the theoretical part of the project, before working on anything practical, this was a beneficial decision as I used the small research gathered in challenge week and improved on it to create an algorithm from the ground up using similar research, this section made a big section of all the backlog information, this where the idea of the three program stages was set, other research sections included some research on sound decibels and sound intensity those section were implemented into the theory later, most of the research on this section went into the intern oral examination presentation and the presentation itself was set in this section of contributions, 13 contributions.   
  
EMoRo: after the pepper robot failed, I have decided to change the robot and program to work with this section included testing the robot with the Arduino program, training with the Arduino program, modifications for the robot and implementing other aspects for testing the various aspects of the robot, 6 contributions.  
  
Code demo: the program had different versions to reach what it has now, some of these demos were part of the testing phase for emoro robot as it was used to test its movement angel turning, ultrasonic sensor, these 2 demoes of the EMoRo robot were created before modifying the robot to use the 2 microphones, the 2 other implement sound localisation where the first one was used to test the microphones and the second one was the first program where the robot follows sound waves, 3 contributions + 2 contributions shared with ‘ EMoRo’.  
  
Final steps: the final steps include improvements in most sections, the theory for ITD was improved, the code was finalized after many drafts and includes contributions for finalizing the report, we can also see a video demonstration of the program, 5 contributions .

Other contributions: the other contributions are small sections relevant to most sections; this includes things like supervisor feedback, weekly updates and updating references to the relevant ongoing research, 10 contributions.

# 10.2 GitLab

This software lets us develop our software and save it in a secure area, we can also safe different files in there and create many other files, during the first period of the project I haven't used git lab as much, as no code was required, however many other aspect of the project are set on GitLab, these directories include, the demo code where all the testing programs and code drafts are there, report drafts where the report draft are included, presentations including challenge week and IOE power points, open day materials includes the poster and the abstract, research including some old background reading and theory testing which has been changed since and lastly final documents which includes the final draft of the main program code and will have a copy of the last draft of the capstone report.

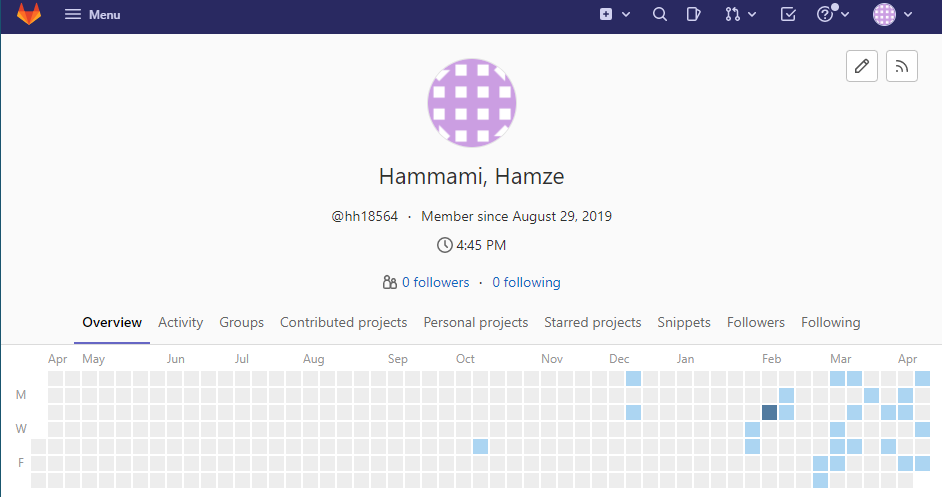


Figure 26 GitLab timeline October – April 15

# 11-project conclusion

The result of the project was a success, from the demo video we can see the robot moving around fixing its position to face the speaker, while the practical work has proven to work out, the theories in sections 4 and 5 could prove to provide an easy and cheap sound localization, although many robots use more than 2 microphones to achieve that, this project proves possible for the use of only to, the program needs to be improved further, even us humans only need 2 ears to pinpoint the source of the sound, and this project proves we could implement that onto a robot without the need of more components.

# 12 code GitLab

The final code is available through this link, small comments are added into the program:

<https://cseegit.essex.ac.uk/ce301_21-22/CE301_hammami_hamze/-/blob/master/final%20documents/final_code.ino>

# 13-citation

## 13.1 content citation

[1] Cohen, Michael. (2010). Under-explored dimensions in spatial sound.

[2] BBC Bitesize. 2021. Bearings - Using bearings in trigonometry - National 5 Maths Revision - BBC Bitesize. [online] ] [Accessed 24 November 2021].

[3] Aaronson, N. and Hartmann, W., 2021. Testing, correcting, and extending the Woodworth model for interaural time difference [Accessed 24 November 2021].

[4] Tollin, D. and Yin, T., 2021. Sound Localization: Neural Mechanisms. (PDF was restricted since last accessed in October)

[5] Courses.washington.edu. 2021. [online] Available at: <https://courses.washington.edu/psy333/lecture\_pdfs/Week9\_Day2.pdf> [Accessed 27 November 2021].

[6] Citeseerx.ist.psu.edu. 2021. [online] Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.642.3837&rep=rep1&type=pdf> [Accessed 27 November 2021].

[7] Courses.lumenlearning.com. 2021. Sound Intensity and Sound Level | Physics. [Accessed 7 December 2021].

[8] Murray, John & Erwin, Harry & Wermter, Stefan. (2004). Robotics sound-source localization and tracking using interaural time difference and cross-correlation.

[9] Nabiyev, Nabi & Malekzadeh, Saber. (2021). Anomalous Sound Localization Estimation.

[10] Doc.aldebaran.com. 2021. ALSoundLocalization — Aldebaran 2.5.11.14a documentation. [Accessed 11 October 2021].

[11] Gracey, B., 2022. Sound pressure terms and definitions - acoustic glossary. [online] Acoustic-glossary.co.uk. Available at: <https://www.acoustic-glossary.co.uk/sound-pressure.htm>.

[12] "EMoRo", *Emoro.eu*, 2022. [Online]. Available: https://www.emoro.eu/. [Accessed: 8- Jan- 2022].

[13] “Arduino” - Home, *Arduino.cc*, 2022. [Online]. Available: https://www.arduino.cc/. [Accessed: 10- Jam- 2022].

[14] "Getting started with Arduino - Arduino Video Tutorial | LinkedIn Learning, formerly Lynda.com", *LinkedIn*, 2022. [Online]. Available: <https://www.linkedin.com/learning/learning-arduino-foundations-2/getting-started-with-arduino-14275737?autoplay=true&contextUrn=urn%3Ali%3AlyndaLearningPath%3A5cd45b50498e57f90fbf63ce&u=51088249> [Accessed: 10- Jan- 2022].

[15] "Hearing", *Hearing*, 2022. [Online]. Available: https://www.open.edu/openlearn/science-maths-technology/biology/hearing/content-section-12.4.

[16] "KY-038 Microphone sound sensor - SensorKit", *Sensorkit.joy-it.net*, 2022. [Online]. Available: <https://sensorkit.joy-it.net/en/sensors/ky-038>.

[17] "EMoRo 2560 - Arduino Reference", *Arduino.cc*, 2022. [Online]. Available: https://www.arduino.cc/reference/en/libraries/emoro-2560/.

[18] https://www.youtube.com/watch?v=t0P2uQP2U0k

[19] https://www.youtube.com/watch?v=TOidBG7mi-w

## 13.2 figures citation

Repeated reference: [5] Courses.washington.edu. 2021. [online] Available at: <https://courses.washington.edu/psy333/lecture\_pdfs/Week9\_Day2.pdf> [Accessed 27 November 2021].

[20] "EMoRo robot kit", *Emoro.eu*, 2022.

[21] "GLAM module", *Emoro.eu*, 2022.

[22] "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino | Random Nerd Tutorials", *Random Nerd Tutorials*, 2022.

[23]"Pepper the humanoid and programmable robot | SoftBank Robotics", *Softbankrobotics.com*, 2022. [Online]. Available: https://www.softbankrobotics.com/emea/en/pepper.

[24] "Pepper SDK for Android — QiSDK", *Qisdk.softbankrobotics.com*, 2022. [Online]. Available: https://qisdk.softbankrobotics.com/sdk/doc/pepper-sdk/index.html. [Accessed: 14- Apr- 2022].

[25] Sun, Liang & Zhong, Xuan & Yost, William. (2015). Dynamic binaural sound source localization with interaural time difference cues: Artificial listeners. Journal of the Acoustical Society of America.