DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2017

Sound following robot

Ljudföljande robot

KARL ENOKSSON

BOHAN ZHOU

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Sound Following Robot

A study on sound localization

KARL ENOKSSON
BOHAN ZHOU

Bachelor's Thesis at ITM
Supervisor: Naveen Mohan
Examiner: Nihad Subasic

TRITA MMK 2017:19 MDAB 637

Abstract

There are many different areas of use for sound localization. This concept is not only used to localize a person that is talking but can also be applied for finding a person in need

One method of localizing the position of a sound source is to use several microphones to register the difference of time in which each microphone detects the same sound. Using this information and trigonometry, the direction of the sound source can be calculated. The objective of this thesis is to investigate how precisely the position of a sound source can be determined using the aforementioned technique whilst varying the distance and angle of the sound source.

In order to explore the capabilities of TDOA and test the obtainable accuracy, a demonstrator was built. On a complete car chassis, four microphones were mounted and used to determine the direction towards the sound source. Thereafter the robot rotated towards the sound source with an IMU keeping track of how much it had rotated. After this movement a comparison was made between the robots direction and the actual direction of the sound source.

Lastly an ultrasonic sensor was placed on the robot for obstacle detection whilst tracking the sound. The vehicle traveled straight forward until the ultrasonic sensor deemed that an object was too close.

The results show that an increased distance yields a more accurate sound localization and that there are some angles in which the sound localization functioned better.

Referat

Ljudföljande robot

Idag finns det många olika användningsområden för ljudlokalisering. Konceptet används inte enbart till att lokalisera en person som pratar men kan också appliceras för att hitta en person i nöd.

En metod för att lokalisera en ljudkällas position innebär kortfattat att med flera mikrofoner registrera de olika tiderna då ljudet når de olika mikrofonerna. Utifrån denna information kan riktningen till ljudkällans position beräknas med hjälp av trigonometri. Målet med denna rapport är att undersöka hur precist en ljudkällas position kan beräknas med den ovannämnda teknik genom att variera avståndet och vinkeln till ljudkällan.

I syfte att genomföra tester byggdes en prototyp. På ett färdigbyggt chassi monterades fyra mikrofoner som användes för att bestämma riktningen till ljudkällan. Därefter roterade roboten mot ljudkällan med hjälp av en IMU som håller reda på hur mycket den har roterat. Efter denna rörelse utfördes en jämförelse mellan robotens riktning och ljudets faktiska riktning.

Slutligen placerades en ultraljudssensor på roboten för att detektera objekt när den spårade ljudet. Fordonet färdades rakt fram tills ett objekt låg för nära ultraljudssensorn.

Resultaten visar att ett ökat avstånd ger en mer nogrann ljudlokalisering samt att för vissa vinklar fungerade ljudlokaliseringen bättre.

Acknowledgements

We would like to thank our supervisor Naveen Mohan for fruitful discussions and supervision throughout the project. We would also like to thank the lab assistants for their patience, advice and feedback. Finally we whould like to thank all the other FIM students for sharing this experience with us. Special thanks goes to Johan Danmo and Johan Abrahamsson for encouraging us and discussing questions regarding the report.

Karl Enoksson & Bohan Zhou
Stockholm, May, 2017

Contents

Nomenclature

Symbols

β	Angle [°]
$\circ$	Degree
Δt	Time difference
θ	Angle [°]
d	Distance [m]
Hz	Hertz
rad	Radians
V	Volt
v	Speed of sound

Abbreviation

$2D$	Two Dimensional
DC	Direct current
IDE	Integrated Development Enviroment
IMU	Inertial Measurement Unit
$MATLAB$	Matrix Laboratories
PWM	Pulse Width Modulation
SEK	Swedish krona
$TDOA$	Time Difference Of arrival
USB	Universal Serial Bus

Chapter 1

Introduction

1.1 Background

Nowadays sound localization is used in various sorts of applications. For instance sound localization is used in humanoid robots to determine where a person is located when assistance is required. The sound localization concept can also be used to get the direction of an animal's position while hunting. The focus of this study will lay in how an Arduino microcontroller and four microphones can be used to locate the source of a sound. The accuracy for the sound localization should be good enough to steer a vehicle, that the microphones are mounted on, towards the sound source.

With a difference in arrival time of the sound to the microphones placed in different locations on the vehicle, the direction to the sound source can be calculated. This sound localization concept is called Time Difference of Arrival (TDOA) and could be applied and used in cases such as locating a person in need.

1.2 Purpose

This report's purpose is to investigate whether an Arduino microcontroller can sufficiently interpret a sound source, calculate the data and exhibit the correct output to the motors so that a vehicle can steer and go to the position of the sound source. This will be analyzed whilst distances and angles towards the sound source are varied. The reason for using an Arduino microcontroller is due to its simplicity and that it has sufficient amount of pins for this project.

The research question to be answered in this report is therefore:

> *"How does different distances and angles towards the sound source affect sound localization using TDOA with an Arduino microcontroller"*

The result from this study might be useful in giving a broader perspective on methods for localizing a specific sound source. This kind of sound localization can thereafter be applied to solve problems such as locating a person that is in need or tracking a person carrying a sound emitter.

1.3 Scope

The scope of this project will be to construct a land vehicle that can track the direction of a sound source with four microphones. Thereafter the vehicle will rotate towards the sound source and travel forward.

There are several different factors not taken into account or that have been limited to an extent in this study. One of these factors is that there is an ultrasonic sensor that will detect objects that lay in the path of the vehicle when it travels forward towards the sound source. However the vehicle will not avoid the object and adjust its path accordingly to continue its route towards the sound source. Instead it will slow down to a halt in front of the object until either the vehicle or the obstacle is physically moved and a new sound is present afterwards.

There has also been a specific budget of 1000 SEK that this project should not exceed. This has limited the amount and types of microphones that are used on the robot. In addition, the demonstrator will only be able to locate sounds in a 180° section. This area is defined as the whole right side of the demonstrator.

1.4 Method

On a complete car chassis, four microphones were mounted. These microphones were placed in a cross-like shape with one in the front, one in the rear and one on each side of the demonstrator. The reasoning behind the placements of the microphones will be described further in the chapter "Theory". Once a sound is detected by the microphone on the right of the demonstrator a timer is started. When the same sound is detected on the left microphone the timer stopped and the recorded time is the time difference between the two microphones. Based on this time difference and information regarding if the sound reached the front microphone or the rear microphone first the direction of the sound can be calculated.

To travel towards the sound source the demonstrator first turned until the vehicle was on a straight path to the sound source. After this step the vehicle continued in a straight path until the ultrasonic sensor detected an obstacle located too close in front of the demonstrator.

1.5 Similar projects

There has been a need in researching the field of sound localization due to its complexity with reverberations or in other words sound waves that bounce back off of walls and objects [Li et al., 2012]. In most of these studies there is a usage of four microphones to determine where the sound originated. Despite this it is possible to only use three microphones to get both the direction and the distance to the sound source using time difference of arrival and trigonometry [Hamada et al., 2012]. However there will be a usage of four microphones in this project to simplify the sound localization.

Similar projects with a vehicle traveling to the sound source has already been made [Huang et al., 1997]. The difference between the mentioned study and this thesis is that the focus of this thesis will lay in how well sound localization using time difference of arrival functions while varying the distance and angle. The angle is measured from the front microphone towards the right side of the demonstrator and is described further in the chapter "Results".

Chapter 2

Theory

2.1 Sound

Sound are longitudinal waves that propagate through a medium such as air or water. A sound wave can be visualized as a sinusoidal wave with a certain frequency, that is how frequently the sound wave oscillates [Young and Freedmen, 2014, p. 565]. In Figure 2.1 two different sound waves can be observed.

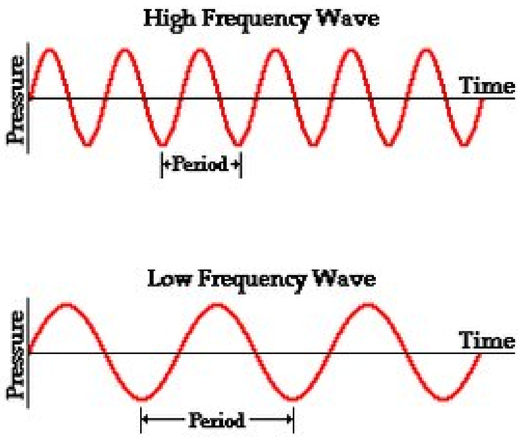


Figure 2.1: Different sinusoidal sound waves [Drezek, 2017].

The first diagram shows a sound wave that has a higher frequency than the diagram below since the period is shorter. It is also noteworthy that the amplitude of the two signals are the same. For a human the audible range is between 20 Hz and 20 000 Hz [Hyperphysics, 2017].

2.2 Microphone

The general concept of a microphone's functionality is that a thin membrane, called a diaphragm, vibrates in the presence of a sound. The amplitude of the vibration is thereafter converted to an electric signal via a coil and a magnet. The electrical signal can thereby be interpreted by the microcontroller to detect the sound level. When a louder sound is present the membrane vibrates with a higher amplitude thus giving a higher electrical signal [Audio-Technica, 2017]. Figure 2.2 shows a microphone in cross section.

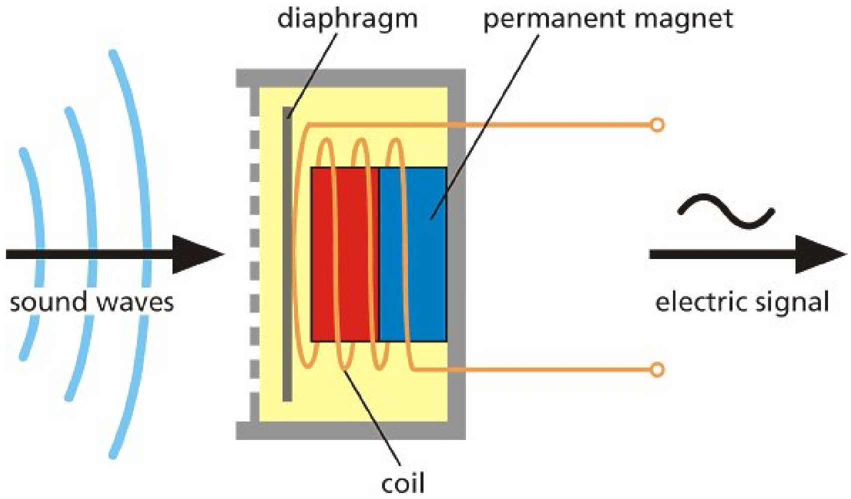


Figure 2.2: Cross section of a microphone [Hampton, 2015]

2.3 TDOA - Time Difference Of Arrival

In this project, TDOA is used to calculate the angle that a sound originated from, as seen from the demonstrator's perspective. Locating a sound source can be done by recording the different times with which a sound wave reaches the microphones. If the distance between the microphones is known as well as the TDOA between them, the angle which the sound came from can be calculated with trigonometry assuming sound propagates through space in a spherical manner with constant speed.

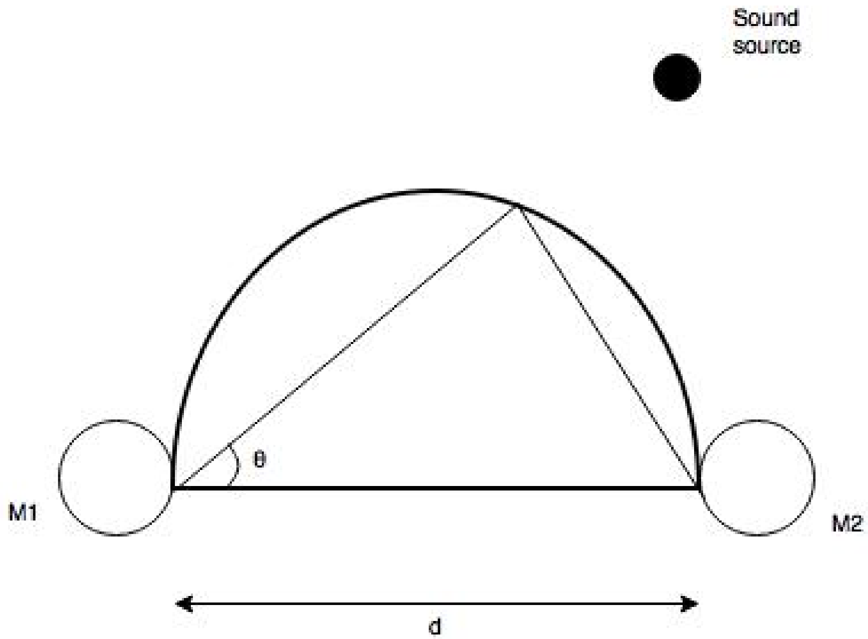


Figure 2.3: TDOA for two microphones

From Figure 2.3 the angle θ can be calculated as

$$\theta = \arccos(\frac{\Delta t * v}{d}) \tag{2.1}$$

where Δt is the time difference, v the speed of sound and d the distance between the microphones as shown in Figure 2.3. All figures are made in the free software draw.io unless otherwise stated.

The way TDOA is implemented in this project is that a timer starts when a specific microphone gets a signal. When the microphone directly opposite to the previous mentioned microphone detects a sound the timer stops and the resulting time is the TDOA between the microphones. Note that only two microphones are needed for this calculation. The other two microphones are used for determining whether the sound source originates from the front or back. This is especially useful since the TDOA is identical for sounds located orthogonally to the microphones and at the same distance between the microphones. For clarification, see Figure 2.4 below.

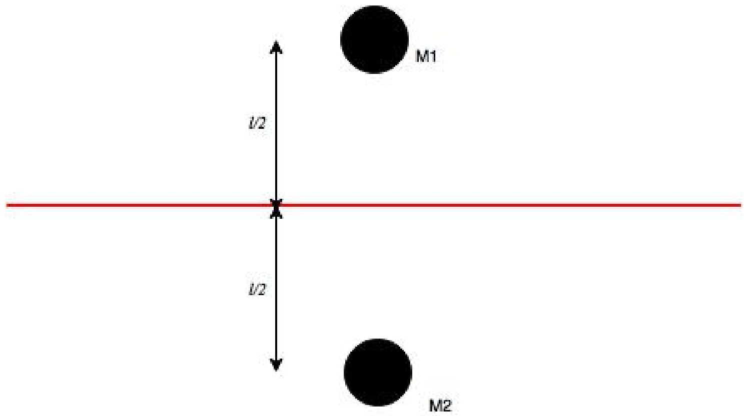


Figure 2.4: Red lines shows where the use of two microphones cannot reliably find the sound source

2.3.1 Placement of microphones

The placement of the microphones is an important factor for localizing the sound. The four microphones were mounted in a cross-like shape on the chassis. This configuration is, in theory, sufficient enough to determine sounds from any direction in a 2D-plane. See Figure 2.5 for the placement and naming of the microphones as seen from above.

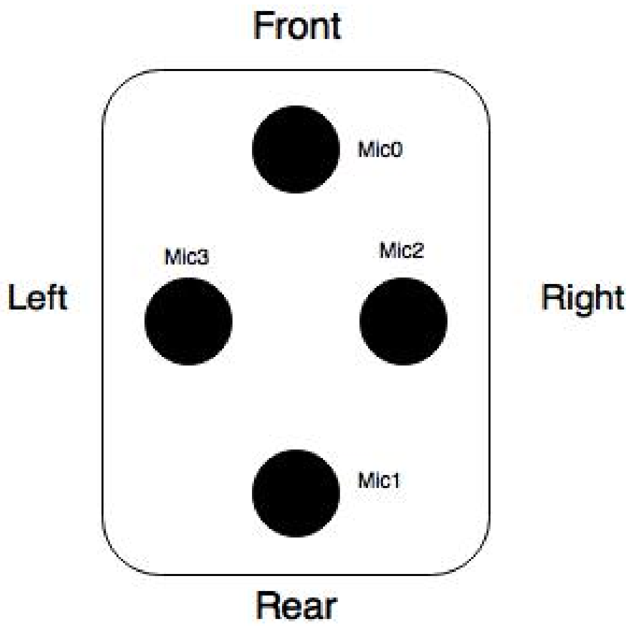


Figure 2.5: Placement of microphones on demonstrator as seen from above

2.4 Interrupt

An interrupt is a signal to the processor that is emitted by the software to indicate that an action needs to be executed immediately. The interrupt will alert the processor to a condition that has high priority. This will lead to the processor pausing the current code that is executed to prioritize a function called an interrupt handler. After the interrupt handler is compiled the processor will continue where it left off and resume to normal activities [Minasi, 1993]. It is important that an interrupt handler is short so that the CPU can return to its main task in a timely manner.

Since the Arduino analogRead function was not fast enough for the demonstrator's purposes, port manipulation and interrupts were used instead. In this project interrupts is not only used to constantly check whether a microphone has picked up a sound or not but also as a microsecond timer.

2.5 PWM - Pulse Width Modulation

A digital signal is a discrete signal which can either be zero or one, where zero corresponds to low signal and one corresponds to high signal. An analog signal is a continuous signal that can vary in the spectrum zero to one. In order to have only one channel for output but both a digital and analog signal, Pulse Width Modulation (PWM) can be used [Johansson, 2013, p. 7:45]. A PWM signal is a pulse of ones and zeroes, shifting at a high speed, allowing the output voltage to be set as a percentage of the maximal output depending on the percentage of time where there is a high output. With a PWM signal usually between 0 and 255, a PWM signal of 0 implicates 0 % of the total output voltage and a PWM signal of 255 implicates 100 % of the total output voltage. If a signal is 127 the signal will be zero half of the time and one the other half of the time. The duty cycle is the percentage of the amount of time the signal is one, in the last example the duty cycle is 50 %.

Figure 2.6 below shows different duty cycles for an Arduino that has 5 V as output voltage.

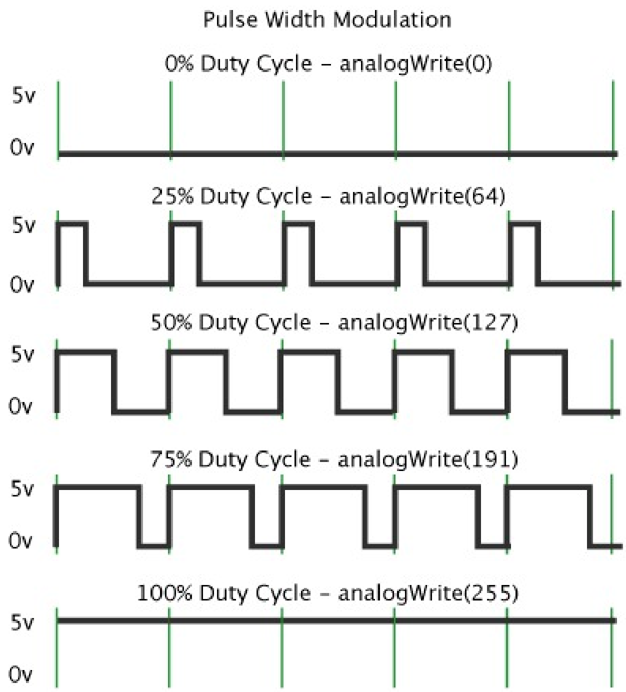


Figure 2.6: Different PWM signals ranging from 0 to 255 [Hirzel, 2017].

Chapter 3

Demonstrator

3.1 Problem formulation

In order to answer the research question *"How does different distances and angles towards the sound source affect sound localization using TDOA with an Arduino microcontroller"*, a series of tests were done. The answer to the question will lay in analyzing the angular difference between where the sound source was and the angle the demonstrator rotated with, which is the angle calculated via TDOA.

3.2 Software

When the demonstrator is to travel towards a sound source two things need to be considered: The angle it needs to rotate with to face the source and the distance it needs to travel until it reaches the source. The angle of the sound source is determined by comparing the time difference of arrival with which the sound reaches two of the microphones as described in section 2.3.

To solve the problem with identical TDOA, two additional microphones were added and placed in the front and rear of the robot. These sound sensors are triggered by interrupts to determine which side the sound arrives first.

Figure 3.1 visualizes the main program as a flow chart. The main program starts with including the necessary libraries and the initialization of the pins. Thereafter, in the main loop, the program waits until microphone 2 registers a sound and starts an interrupt timer immediately afterwards. This timer stops when microphone 3 has registered a sound and the time is equivalent to the TDOA. For numeration of the microphones, see Figure 2.5.

Based on another interrupt, the side on which the sound arrived first is known. Using the TDOA alongside with the knowledge of which side the sound originates from, the angle towards the sound source is calculated and the demonstrator is rotated to this angle. Consequently the demonstrator will only need to follow a straight path until there is an object too close to the robot.

Flowcharts for the interrupts are shown in Figure 3.2. The microphone interrupt is the flowchart to the left and the timer interrupt is the flow chart to the right.

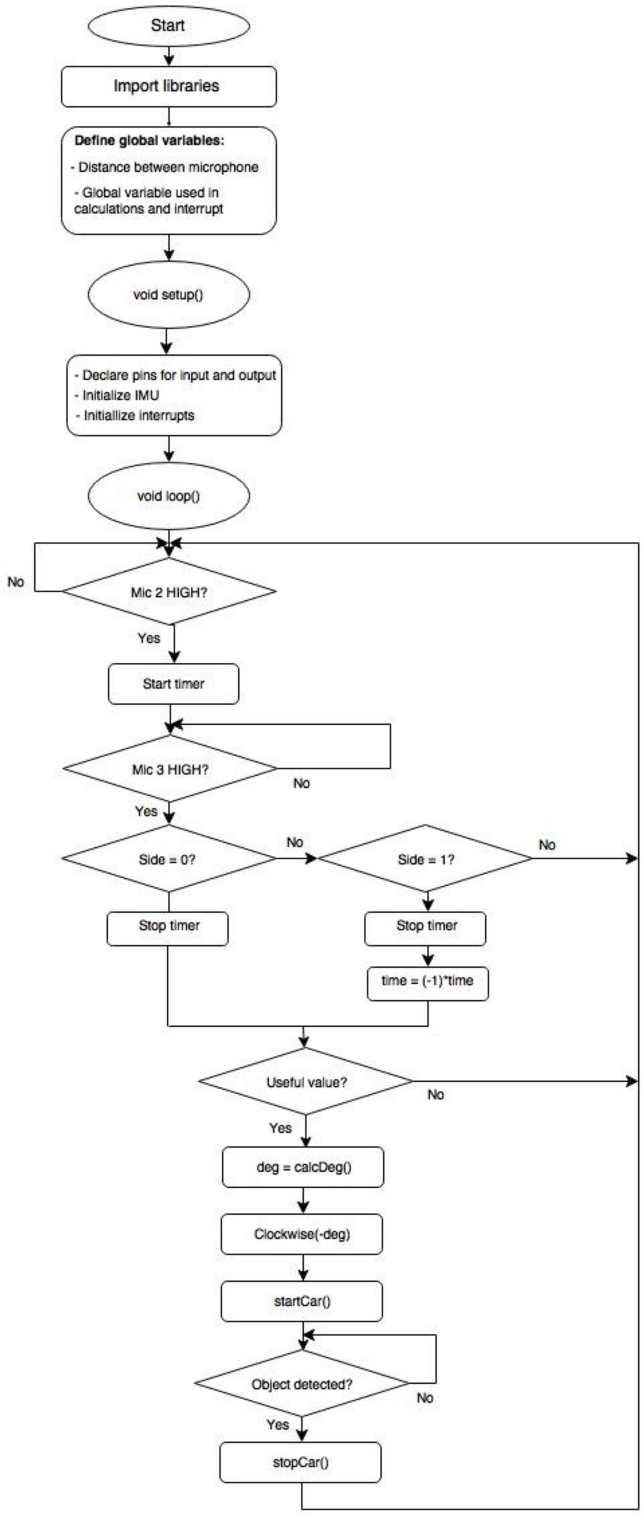


Figure 3.1: Programming flowchart for main program.

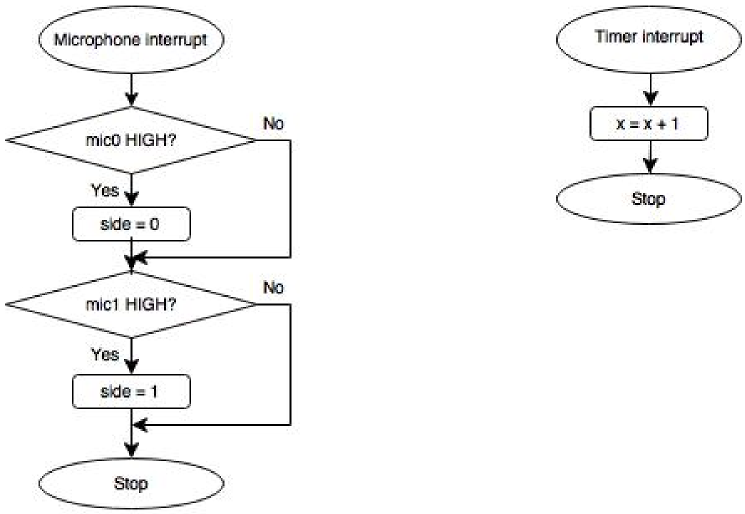


Figure 3.2: Programming flowchart for interrupts

As previously mentioned interrupts were used both as a timer and to determine which side the sound originated from. The timer interrupt counts up after a certain amount of time and is divided by a time factor to scale the time into microseconds after the timer is stopped. Regarding the second interrupt, it is triggered by a high signal on microphone 0 or microphone 1. Based on which of these microphone gets a high signal first, the interrupt sets a variable to a specific value that corresponds to that microphone. Refer to Appendix A for the whole Arduino code.

The two libraries used for the IMU was created by Jeff Rowberg [Rowberg, 2017]. Code for the ultrasonic sensor was written by Dejan Nedelkovski [Nedelkovski, 2015]. Ideas and inspiration for implementing TDOA on an Arduino platform was taken from a project on sound localization [Gusland and Waaler, 2014].

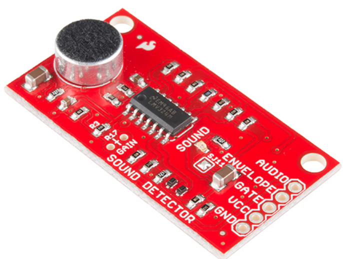
3.3 Hardware

In this section the hardware used to build the demonstrator will be described further. All of the electronic components used in this study are connected to the Arduino UNO. The hardware coupling of the setup can be found in Appendix B.

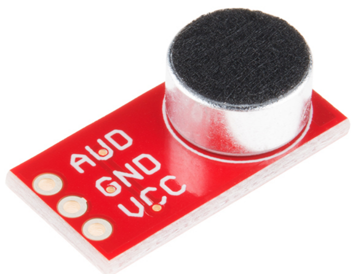
3.3.1 Microphones

In the demonstrator two different types of microphones were used due to problems with ordering of parts. In total four microphones were mounted on the chassis in which one of these microphones was of model SEN-12642 with a built in amplifier [Sparkfun, 2017c] and the other three were electret microphones of model BOB-12758, both from Sparkfun [Sparkfun, 2017b].

The threshold for when the first mentioned microphone detects sound can be adjusted by the usage of a resistor. A higher resistance implicates a lower threshold and vice versa.



(a) Microphone with built-in amplifier [Sparkfun, 2017c]



(b) Electret microphone [Sparkfun, 2017b]

Figure 3.3: The two different types of microphones used

3.3.2 H-bridge

An H-bridge allows currents to go through components in either direction, allowing motors to be run in both ways. The use of an H-bridge also allows the use of an external power supply to the motors [Johansson, 2013, p. 7:47].

The H-bridge used in the demonstrator is of model L298N Dual H-Bridge Motor Controller. The H-bridge is designed for running two different motors simultaneously. Since the demonstrator will have 4 motors in total, one for each wheel, and only one H-bridge is used, it is necessary to connect two DC-motors parallel on each side of the H-bridge. See Appendix B for how the motors are connected. This configuration resulted in simultaneous control of the two DC motors on each side.

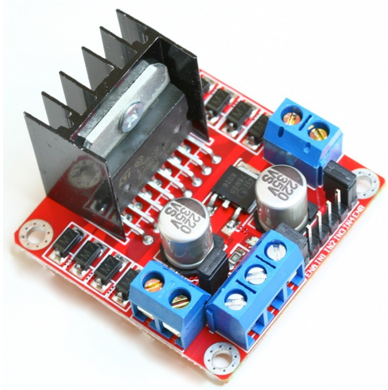


Figure 3.4: L298N Dual H-Bridge Motor Controller [Lampatronics, 2017]

3.3.3 DC motor

The acronym DC in DC motor stands for direct current. This implicates that a direct current is used to supply the motor with electricity. Commonly a DC motor consists of an odd number of poles and the ingoing current direction is what determines the rotational direction of the motor. To reverse the direction of the motor an H-bridge can be used. The smaller variants of DC motors usually comprises of a permanent magnet and poles. The poles are connected to the rotor that is made up of a coil to create a magnetic field when the motor rotates. This magnetic field changes direction when the motor rotates one cycle [Lazaridis, 2010].

For propulsion of the demonstrator four DC motors with built-in gearboxes will be used, one for each of the four wheels. The DC motor used in this project is the DAGU DG02S-A130 GEARMOTOR [DAGU, 2013].

Figure 3.5: DC motor with wheel attached [Sparkfun, 2017a]

3.3.4 Arduino UNO

The Arduino UNO is a 8-bit microcontroller that has the ATmega328P as microprocessor which can be seen as the brain of the Arduino. It has 6 analog input pins for conversion to digital signal and 14 digital input/output pins, out of which 6 enable a PWM signal. The Arduino IDE, or integrated development environment, is the software used to program the Arduino. This is an included software that is based on C/C++. Uploading of the program is done from the computer's USB port to the Arduino's USB B-type port.

The Arduino UNO is an open source hardware and was used in this project based on its simplicity and the fact that it had sufficient amount of pins for the requirements. All of the controlled devices and microphones were connected to the Arduino [Arduino, 2017].

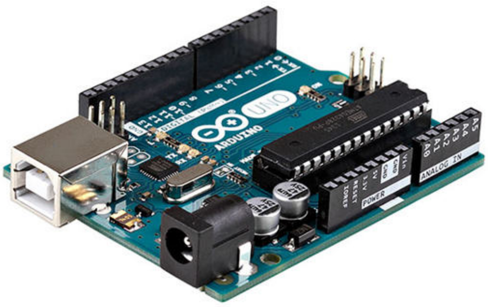


Figure 3.6: Arduino UNO [Arduino, 2017]

3.3.5 Ultrasonic sensor

An ultrasonic sensor is used for calculating the distance to an object. The sensor emits a high frequency sound pulse that is above the audible range for human ears. When the pulse is reflected back by an interfering object the sensor receives the sound pulse. The ultrasonic sensor records the elapsed time between the emitting and receiving of the sound wave. The distance can then be calculated with the time it takes for the pulse to reflect back and the speed of sound [Electronics, 2017].

The ultrasonic sensor is used to prevent the demonstrator from crashing into obstacles that potentially may lay in the path of the vehicle. The model used was the Ultrasonic Ranging Module HC - SR04 as depicted in Figure 3.7.

Figure 3.7: Ultrasonic sensor HC - SR04 [Electrokit, 2017]

Chapter 4

Results

In this section the results of the tests will be presented. In order to answer the research question *"How does different distances and angles towards the sound source affect sound localization using TDOA with an Arduino microcontroller"*, two different tests were performed.

In the first test, the demonstrator was set at a constant angle, measured from the front microphone towards the right side of the demonstrator. This angle is marked as β in Figure 4.1.

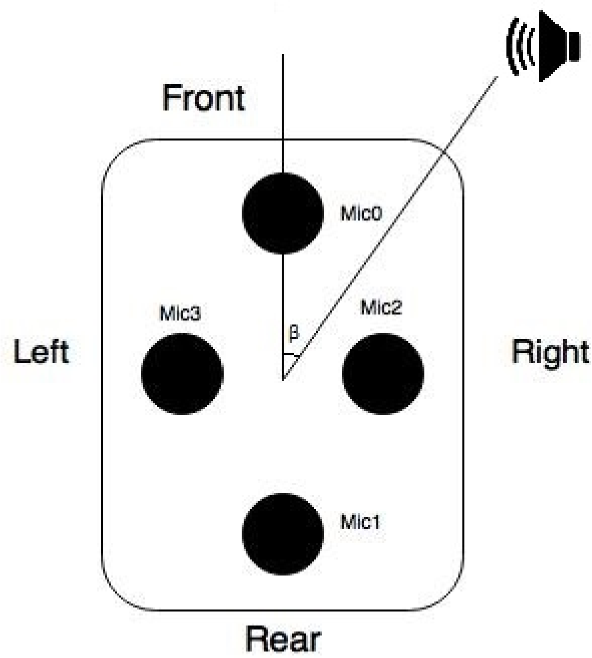


Figure 4.1: Angle between the front and the generated sound

With $\beta = 45°$ distinct clapping noises were made at three different distances and three iterations were made at each distance. The distances taken into account in this thesis were 25 cm, 75 cm and 125 cm from the demonstrator. The clapping noises triggered the microphones and the demonstrator entered its turning state, rotating towards the position it had calculated that the sound originated from. When the rotating movement had finished, the demonstrator came to a halt and the angular difference between the cars direction and the actual direction towards the clap was measured. After all of the measurements were performed the mean value for angular difference at each distance was calculated. The greatest distance, 125 cm, turned out to be the most accurate and became the distance used in the second test. This test was done in order to find out how different distances between the sound source and demonstrator affects sound localization. The results from the first test can be found in Table 4.1, Table 4.2 and Table 4.3 below.

Table 4.1: Tests made to see how distance affects sound localization for a sound source located at 45° and 25 cm away

Angular difference whilst varying distance		
Angle [°]	Distance [cm]	Angular difference [°]
45	25	20
45	25	18
45	25	27
	Mean	21.67

Table 4.2: Tests made to see how distance affects sound localization for a sound source located at 45° and 75 cm away

Angular difference whilst varying distance		
Angle [°]	Distance [cm]	Angular angle [°]
45	75	14
45	75	15
45	75	18
	Mean	15.67

Table 4.3: Tests made to see how distance affects sound localization for a sound source located at 45° and 125 cm away

Angular difference whilst varying distance		
Angle [°]	Distance [cm]	Angular difference [°]
45	125	-7
45	125	17
45	125	15
	Mean	8.33

In the second test, the distance between the clap and the demonstrator was kept constant at 125 cm and β was varied. The angles that were taken into consideration were 30°, 60°, 120° and 150°. Five iterations were done at each angle and the results from the second test can be found in Table C.1, Table C.2, Table C.3 and Table C.4 in Appendix C. This test was performed in order to find out how different angles towards the sound source affects sound localization.

Figure 4.2 shows the measured angular differences from the second test plotted against the angle β. Every evaluated value of β has its corresponding color to distinguish them apart. Blue corresponds to 30°, pink to 60°, black to 120° and red to 150°. Each initial angle, β, also has its respective mean value depicted as a brown 'X'.

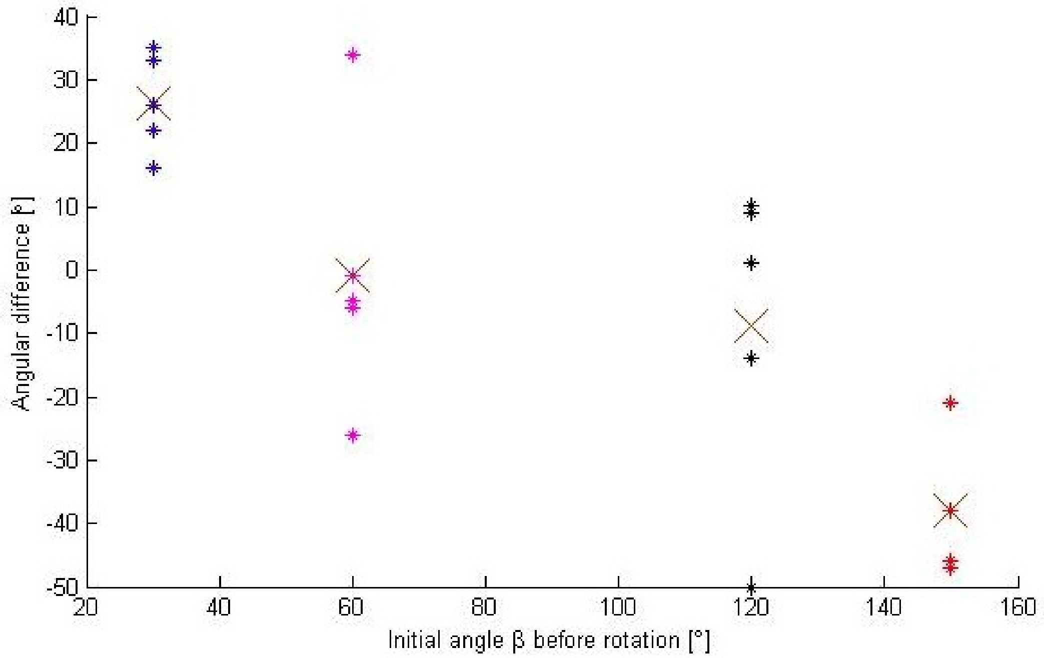


Figure 4.2: Results from the second test. Plotted in MATLAB™

Chapter 5

Discussion and conclusions

5.1 Discussion

From the first test, the trend of the results points to the fact that a greater distance between the sound source and microphones used for TDOA yields a greater accuracy with regards to determining what direction a sound originated from. This may be due to the fact that sound propagates through air spherically and as the radius of a sphere increases, its surface becomes more and more flat. As a consequence, the angle from which the sound originated from would be calculated more precisely if it came from further away.

Regarding the second test, the software had a major impact on how the sound localization functions. The demonstrator is limited to sound localization in an 180° interval and since several unique directions can yield identical TDOA:s, four microphones were mounted in a cross-like shape creating four quadrants in the space between them, just like a Cartesian coordinate system. To solve the problem with identical TDOA:s, the amount of degrees the demonstrator rotated was dependent on what quadrant the sound came from. Since the two models of microphones had different sensitivities, the demonstrator might wrongly act as if the sound originated from a certain quadrant when it in reality came from a different one. Even though one of the microphone's sensitivity could be adjusted by adding a resistor, identical sensitivities were not successfully achieved and the problem persisted. This phenomenon might explain the fact that the values noted in C.4 are so off. The sound localizations dependency on the correct quadrant is also the reason for the chosen values of β. Every tested β value lays well defined in a quadrant. It is also noteworthy that when the sound localization functioned properly it was very accurate. At best there was only an angular difference of 1°.

The precision of the sound localization for different angles varied drastically. Figure 4.2 shows that the mean value for the angular difference is the smallest at 60°and 120°. When β was 30°the demonstrator tends to rotate too much and for β equal to

150 the demonstrator tends to turn too little. Therefor the optimal angle for sound localization is around 60°and 120°for the demonstrator.

There are several factors that could have affected the demonstrators performance. The yaw values that were read from the IMU were constantly drifting downwards which is a source for errors. However, the drifting of the yaw occurred at a much slower rate than the demonstrators rotating speed so the error from the IMU was deemed acceptable.

The initial idea was to use only one kind of microphone for the TDOA calculations. The only reason for using two different kinds were that the delivery time was too long so temporary microphones were used and ended up being in the final build of the demonstrator since not all of the initially ordered microphones arrived on time. One problem that arose was that the sensitivity differed between the two kinds of microphones which led to inaccuracies in the calculations.

5.2 Conclusions

After performing tests on the demonstrator and analyzing the data the research question *"How does different distances and angles towards the sound source affect sound localization using TDOA with an Arduino microcontroller"* can be answered.

A greater distance between the sound source and microphones yields greater accuracy when utilizing TDOA as a mean for sound localization. When the TDOA algorithm has pinpointed the correct quadrant it can be very precise in locating a sound source.

When the sound originated from certain angles the sound localization was very accurate. After tests were conducted on the demonstrator it was found that $\beta = 60°$ and $\beta = 120°$ gave the best results.

Chapter 6

Recommendations for future work

A PID-regulator could be implemented to make the robot actively follow a moving sound emitter and perhaps keep a certain distance to that emitter. It was an early stage idea to construct the robot so that it would follow a sound emitter in this manner. A possible area of use for this kind of project would be shopping malls where each autonomous following shopping cart would have a corresponding sound emitter placed on each of the customers. The frequency of the sound emitted would suitably be above or under the human limit for what is hearable. Obstacle avoidance would be necessary in this kind of application as well to ensure that the robot follows the user at all times.

When the robot has rotated towards the sound source it is to go forward until the ultrasonic sensor signals that an object is close to the front of the robot. Instead, the distance that the robot should traverse after localizing the sound source could be calculated using data about the intensity of the sound from the microphones. This could be implemented if the sound intensity is known for the sound source since the intensity of the sound is proportional to the distance between source and listener to the power of minus two. This idea could not be investigated further in this project because of the late arrival of the kind of microphone that are able to read sound intensity.

As of now the demonstrator does not have any kind of obstacle detection except for the ultrasonic sensor placed at the front of the robot. This sensor is only used to detect an object in a straight path from the demonstrators point of view and there is also no obstacle avoidance in the software.

To solve the drifting of the IMU values a 9 degrees of freedom IMU would be more reliable. For future work in TDOA it is also recommended to stick to one microphone model to avoid problems with different microphone sensitivities.

Bibliography

Arduino (2017). Arduino uno. Accessed: 2017-04-03. Available from: http://www.arduino.org/products/boards/arduino-uno.

Audio-Technica (2017). What a microphone does. Accessed: 2017-04-14. Available from: http://www.audio-technica.com/cms/site/b0d226992d31e25d/.

DAGU (2013). Dagu off set gear motor acessorios 1:48. Accessed: 2017-03-23. Available from: http://www.dagurobot.com/goods.php?id=86.

Drezek, E. (2017). How do frequencies affect pitch? Accessed: 2017-03-23. Available from: http://eaninquiry1.weebly.com/how-do-frequencies-affect-pitch.html.

Electrokit (2017). Avståndsmätare ultraljud hc-sr04 2-400 cm. Accessed: 2017-04-02. Available from: https://www.electrokit.com/avstandsmatare-ultraljud-hcsr04-2-400cm.52412.

Electronics (2017). What is an ultrasonic sensor? Accessed: 2017-04-13. Available from: http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html.

Gusland, D. and Waaler, J. S. (2014). Lydlokalisering. Accessed: 2017-04-19. Available from: https://www.dropbox.com/sh/ozv4ijf6usx0nn3/AABJg4WpNF0kR6R3kMqaTNoxa?dl=0.

Hamada, N., Kasprzak, W., and Przybysz, P. (2012). Auditory scene analysis by time-delay analysis with three microphones. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–8. IEEE.

Hampton, A. (2015). Sounds. Accessed: 2017-03-23. Available from: https://comp1alexihampson.wordpress.com/2015/02/09/sounds/.

Hirzel, T. (2017). Pwm. Accessed: 2017-03-23. Available from: https://www.arduino.cc/en/Tutorial/PWM.

Huang, J., Supaongprapa, T., Terakura, I., Ohnishi, N., and Sugie, N. (1997). Mobile robot and sound localization. In *Intelligent Robots and Systems, 1997.*

IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on, volume 2, pages 683–689. IEEE.

Hyperphysics (2017). Sensitivity of human ear. Accessed: 2017-04-11. Available from: `http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/earsens.html`.

Johansson, H. (2013). *Elektroteknik*. Department of Machine Design KTH.

Lampatronics (2017). L298n motor driver board module for arduino. Accessed: 2017-04-03. Available from: `http://lampatronics.com/product/l298n-motor-driver-board-module-for-arduino/`.

Lazaridis, G. (2010). How dc motors are made and how they work. Accessed: 2017-04-03. Available from: `http://pcbheaven.com/wikipages/How_DC_Motors_Work/`.

Li, X., Shen, M., Wang, W., and Liu, H. (04 Jul 2012). Real – time sound source localization for a mobile robot based on the guided spectral-temporal position method. *International Journal of Advanced Robotic Systems*, pages 1–8.

Minasi, M. (1993). Interrupts made easy. Accessed: 2017-04-13. Available from: `http://www.atarimagazines.com/compute/issue149/60_Interrupts_made_easy.php`.

Nedelkovski, D. (2015). Ultrasonic sensor. Accessed: 2017-05-16. Available from: `http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/`.

Rowberg, J. (2017). i2cdevlib. Accessed: 2017-05-17. Available from: `https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050`.

Sparkfun (2017a). Hobby motor and encoder kit. Accessed: 2017-04-10. Available from: `https://www.sparkfun.com/products/13260`.

Sparkfun (2017b). Sparkfun electret microphone breakout. Accessed: 2017-04-10. Available from: `https://www.sparkfun.com/products/12758`.

Sparkfun (2017c). Sparkfun sound detector. Accessed: 2017-03-23. Available from: `https://www.sparkfun.com/products/12642`.

Young, H. D. and Freedmen, R. A. (2014). *Sears and Zemansky's University Physics with modern physics technology update*. Pearson, 13th edition.

Appendix A

Arduino code

The Arduino code from this thesis can be found in the following pages. The code was programmed in the included software Arduino IDE.

```
/*
  Authors: Karl Enoksson and Bohan Zhou
  Project name: Sound following robot
  Course: MF133X
  TRITA-number: MMK 2017:19 MDAB 637
  Date: 2017-05-21
*/

//Including general libraries
#include <math.h>
//Including libraries for IMU
#include <I2Cdev.h>
#include <MPU6050.h>
#include <Wire.h>
//Including libraries for interrupts
#include <avr/interrupt.h>

//Declaration of variables and constants used in calculations
int distance = 31; //distance between microphones in cm
float time1;
double timefactor = 0.049; //timefactor to scale interrupt counter so that it counts in microseconds
float deg;
int rearFront;
double offset;


//Declaration of variables used by ultrasonic sensor
long duration;
int dist;
//Declaration of pins used by ultrasonic sensor
#define trigPin 8
#define echoPin 3

//Declaration of volatile variables used in interrupts
volatile int side = 5;
volatile float x = 0;

//Declaration of variables and constants used by IMU
MPU6050 accelgyro;

int16_t ax, ay, az, gx, gy, gz;

double timeStep, time0, timePrev;
double arx, ary, arz, grx, gry, grz, gsx, gsy, gsz, rx, ry, rz;

int i;
double gyroScale = 131;
```

```
//Declaration of pins used by motors
//Right side motor control
int ctrl_IN1 = 7;
int ctrl_IN2 = 6;
int ctrl_ENABLE1 = 9; //Enable right
//Left side motor control
int ctrl_IN3 = 5;
int ctrl_IN4 = 4;
int ctrl_ENABLE2 = 10; //Enable left


void setup() {
  DDRC = B000000; //Set I/O pins for port C

  //Setup IMU
  Wire.begin();
  accelgyro.initialize();
  time0 = millis();
  i = 1;

  // Set pins to motor as output
  pinMode(ctrl_IN1, OUTPUT);
  pinMode(ctrl_IN2, OUTPUT);
  pinMode(ctrl_IN3, OUTPUT);
  pinMode(ctrl_IN4, OUTPUT);
  pinMode(ctrl_ENABLE1, OUTPUT);
  pinMode(ctrl_ENABLE2, OUTPUT);

  //Set pins to ultrasonic sensor as input or output
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  //Setup for interrupt
  cli();
  //Set timer interrupt
  TCCR2A = 0; // Set entire TCCR2A register to 0
  TCCR2B = 0; // Same for TCCR2B
  TCNT2 = 0; //Initialize counter value to 0
  // Set compare match register for increments
  OCR2A = 40;
  TCCR2A |= (1 << WGM21); // Turn on CTC mode
  TCCR2B |= (1 << CS21); // Set CS21 bit
  TIMSK2 |= (1 << OCIE2A); // Enable timer compare interrupt

  //Set interrupt on change
  PCICR |= 0b00000010; //Enables Port C Pin Change Interrupts
  PCMSK1 |= 0b00000011; //Trigger pins for interrupt
  sei();
}
```

```
void loop() {

  time1 = 0; //Reset time for TDOA after each loop
  side = 5; //Set default value for side
  offset = IMUreading(); //Gets offset value for the demonstrator
  grz = offset; //Define grz before while loop


  if ((PINC & B000100) != 0) {
    x = 0; //Start timer for TDOA if microphone 2 registers sound
    while ((PINC & B001000) == 0) {  //Loop until microphone 3 registers sound
    }

    time1 = x / timefactor; //timefactor scales interrupt counter so that it counts in microseconds


    if (abs(time1) < 912 && abs(time1) > 20 && side != 5) {
      // 912 microsec is the longest time it takes for sound to travel between the microphones

      if (side == 1) {
        time1 = (-1) * time1; //Make time1 negative if sound reaches microphone 1 first
      }

      deg = (-1) * calcDeg(time1, 2, side); //Make deg negative to allow clockwise turn due to IMU

      while ( grz - offset > deg ) { //Rotate demonstrator until in a stright line towards sound source
        Clockwise(); //Rotates the demonstrator in the clockwise direction
        grz = IMUreading();
      }

      stopCar(); //Stops the rotational movement
      delay(250);

      dist = ultrasonicSensor(); //Define dist before while loop


      while (dist > 6) { //The demonstrator is to move forward until an object is detected too close in front
        startCar();  // Demonstrator travels straight forward
        dist = ultrasonicSensor();
      }

      stopCar();  //Stop car when an object is too close
    }
  }
}
```

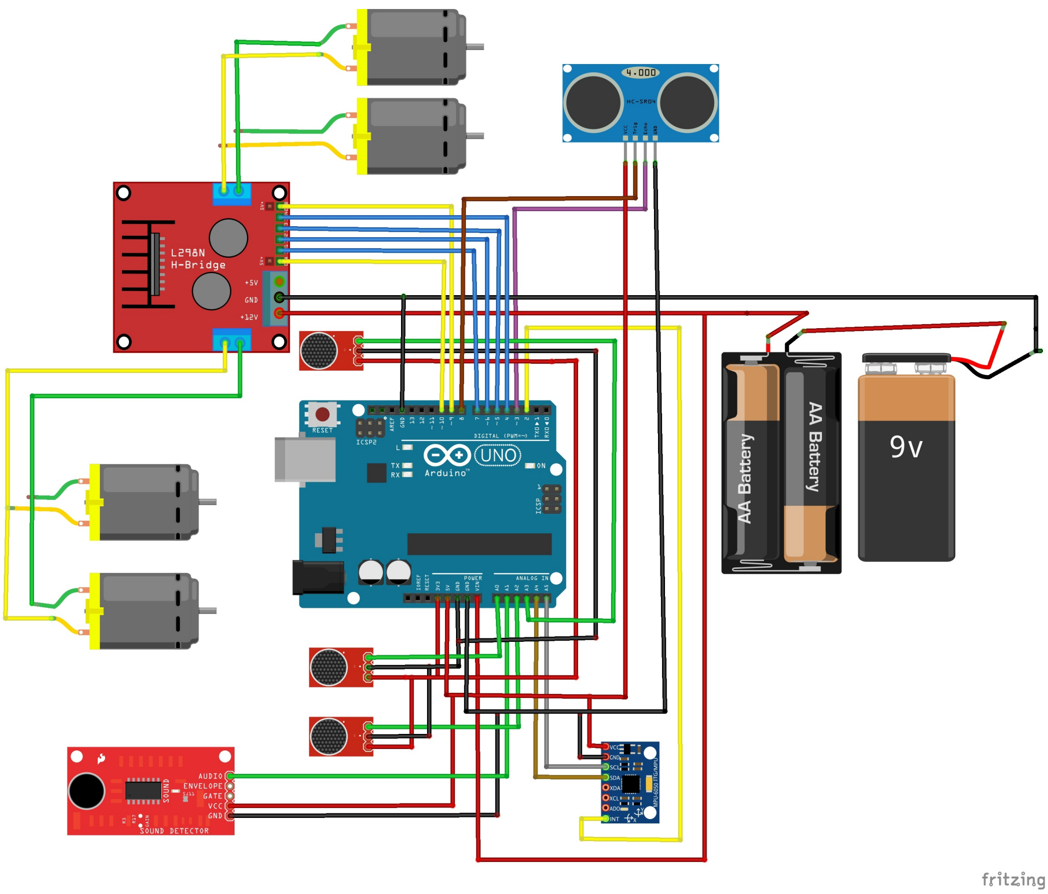
```
//Function lets the demonstrator travel foward
void startCar() {
  analogWrite(ctrl_ENABLE1, 65);
  analogWrite(ctrl_ENABLE2, 65);
  digitalWrite(ctrl_IN1, LOW);
  digitalWrite(ctrl_IN2, HIGH);
  digitalWrite(ctrl_IN3, HIGH);
  digitalWrite(ctrl_IN4, LOW);
}


//Function stops the demonstrator
void stopCar() {
  analogWrite(ctrl_ENABLE1, 100);
  analogWrite(ctrl_ENABLE2, 100);
  digitalWrite(ctrl_IN1, LOW);
  digitalWrite(ctrl_IN2, LOW);
  digitalWrite(ctrl_IN3, LOW);
  digitalWrite(ctrl_IN4, LOW);
}


//Function rotates the demonstrator in the clockwise direction
void Clockwise() {
  analogWrite(ctrl_ENABLE1, 95);
  analogWrite(ctrl_ENABLE2, 95);
  digitalWrite(ctrl_IN1, HIGH);
  digitalWrite(ctrl_IN2, LOW);
  digitalWrite(ctrl_IN3, HIGH);
  digitalWrite(ctrl_IN4, LOW);
}


//Function returns the yaw value
double IMUreading() {
  timePrev = time0;
  time0 = millis();
  timeStep = (time0 - timePrev) / 1000; //time-step in s

  //Collect readings
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  //Apply gyro scale from datasheet
  gsz = gz / gyroScale;
```

```
  //Set initial values equal to accel values
  if (i == 1) {
    //Calculate accelerometer angles
    arz = (180 / PI ) * atan(sqrt(square(ay) + square(ax)) / az);
    grz = arz; //grz is the demonstrators yaw
  }
  //Integrate to find the gyro angle
  else {
    grz = grz + (timeStep * gsz);

  }
  i++;
  return grz;

}


//Function returns distance to an object in front of the demonstrator
int ultrasonicSensor() {
  //Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  //Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  //Calculate the distance and returning the value
  return duration * 0.034 / 2;
}
```

Appendix B

Hardware coupling

This appendix gives a visual representation of how all the components are connected. This figure was made in the free software Fritzing.



Appendix C

Angular difference whilst varying angle

Following tables show test data when the angle towards the sound source was varied and the distance was set to 125 cm.

Table C.1: Tests made to see how different angles affects sound localization for a sound source located at 30° and 125 cm away

Angular difference whilst varying angle		
Angle [°]	Distance [cm]	Angular difference [°]
30	125	22
30	125	33
30	125	35
30	125	16
30	125	26
	Mean	26.4

Table C.2: Tests made to see how different angles affects sound localization for a sound source located at 60° and 125 cm away

Angular difference whilst varying angle		
Angle [°]	Distance [cm]	Angular difference [°]
60	125	-1
60	125	34
60	125	-5
60	125	-26
60	125	-6
	Mean	-0.8

Table C.3: Tests made to see how different angles affects sound localization for a sound source located at 120° and 125 cm away

Angular difference whilst varying angle		
Angle [°]	Distance [cm]	Angular difference [°]
120	125	10
120	125	9
120	125	1
120	125	-14
120	125	-50
	Mean	-8.8

Table C.4: Tests made to see how different angles affects sound localization for a sound source located at 150° and 125 cm away

Angular difference whilst varying angle		
Angle [°]	Distance [cm]	Angular difference [°]
150	125	-38
150	125	-46
150	125	-38
150	125	-21
150	125	-47
	Mean	-38

TRITA MMK 2017:19 MDAB 637