Final Year Project

Assistant Navigator - Obstacle Avoidance for the Visually Impaired

Zach Dunne

Student ID: 18416966

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science with Data Science

Supervisor: Professor Kechadi

UCD School of Computer Science

University College Dublin

April 26, 2022

Table of Contents

Note: *A private Gitlab repository containing my project code has been made available to the academic supervisor and moderator at: `https://csgitlab.ucd.ie/ZachD99/fyp_obstacle_avoidance`*

Abstract

The main aim of this project is to benefit those with visual impairments through the recent advancements in digital devices and communication. More precisely, this project was undertaken with the intention of developing a device that is capable of assisting the visually impaired with avoiding obstacles. This question has already been tackled in different ways in the literature. However, the idea behind this approach is to use ultrasonic sensors on a helmet that survey the area in front, and to the left and right of the user. This approach is different to the existing solutions. Ultrasonic sound waves are emitted from the sensors' transmitters, bounce off nearby objects and are then received by the sensors' receivers. The time taken for this to happen is used to find the distance from the sensor to the object. If this distance is below a pre-set threshold, the corresponding buzzer will begin to beep. This alerts the user of the obstacle and allows them time to alter their course. The frequency of the beeping gives the user an idea of how far away the obstacle is. This device successfully fulfils its purpose of aiding visually compromised individuals avoiding collisions with obstacles in everyday life.

Chapter 1: **Project Specification**

Assistant Navigator:
Obstacle Avoidance for the Visually Impaired

Academic Supervisor: Prof. Tahar Kechadi

1.1 Introduction to the Problem

Visually impaired individuals require aid to move around in indoor and outdoor environments. Individuals who cannot navigate autonomously rely on other people or devices to guide them through life. Technology, with recent, significant advances in ICT, must play a role in helping these people to manoeuvre around obstacles so that they can travel safely and with more confidence.

1.2 The Challenges

The main challenge for this project was to design a system that is capable of detecting obstacles for visually impaired individuals. It should also notify the user to help them avoid colliding with an obstacle. The goal was to help those with visual impairments to navigate as well as those with unimpaired vision can. This had to be achieved on a minimal budget using inexpensive, freely available hardware and software.

The second challenge in this project is to develop a system that is efficient without being cumbersome to use. In other words, the device should be light, portable, ergonomic, and located in an area that is not obstructive to the user. As not all visually impaired individuals are completely blind, the device should not block the user's eyes, further hindering their vision. Placing a device on the user's head leaves their hands free to be used and does not obstruct the eyes.

The third challenge is that the device should ideally be easy to set up and use. Vision should not be a requirement to set up the device. If there are multiple different buttons in different places, it would be difficult for a visually impaired person to set up the device. Ideally, there should be one button to press or one cable to plug in to start up the device.

1.3 Approach

I chose to develop a device that is located on the head, so the user's hands are free. This allows the user to focus all of their attention on navigation without having to hold a device in their hands. Ultrasonic sensors were used to detect obstacles in the user's path and then relay this information to the user, alerting them of the obstacle and advising them to change their direction of movement. This system involves the user wearing a helmet that carries a device that can survey the area to the left, right, and in front of them, via ultrasonic sensors. This device processes the data that it receives from the sensors using an Arduino microcontroller to detect obstacles before notifying the user of them so that they can be easily avoided. These choices constitute an excellent solution to all aforementioned challenges.

1.4 System Specifications

The implemented system is comprised of three HC-SR04 Ultrasonic Sensors and three buzzers mounted to the left, right and middle of a helmet. The sensors are connected via DuPont wires to an open-source microcontroller Arduino UNO R3 board. The microcontroller processes the distances that are read by each sensor. If a distance comes within a given threshold, the board then sends a signal to the corresponding buzzer to say that there is an obstacle nearby, in the specified direction. This threshold is a global variable that can be set to the user's preference within the software implementation and then loaded onto the board before usage commences. The buzzer then beeps at a rate that is inversely proportional to the distance that is observed. The system is powered by a 5V battery, connected to the microcontroller within the helmet by means of a USB power cable.

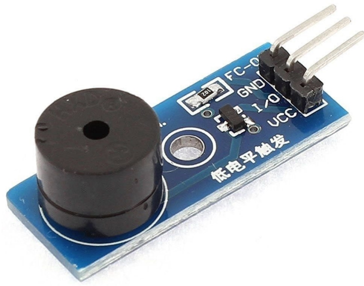


Figure 1.1: Arduino Buzzer

- Arduino™ UNO R3 Microcontroller
- 3 HC-SR04 Ultrasonic Sensors
- 3 5V Arduino Buzzers (Fig 1.1)
- DuPont wires

1.4.1 Arduino™ UNO R3 Microcontroller

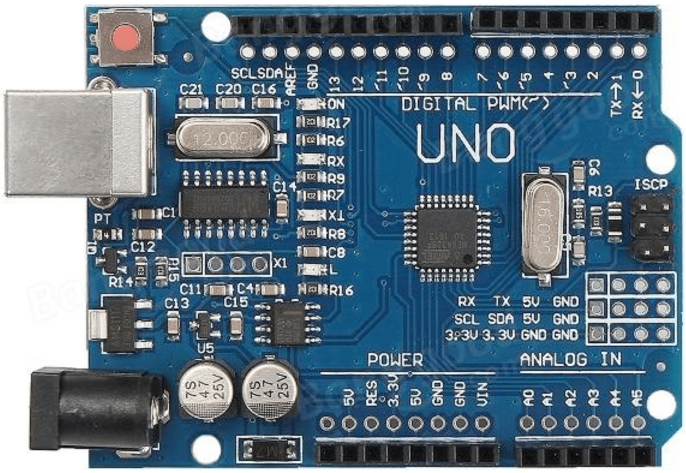


Figure 1.2: Arduino™ UNO R3[1]

The Arduino™ UNO R3 [2] seen in Figure 1.2 is an inexpensive, open-source microcontroller board that can read a variation of inputs and turn them into outputs. Software is implemented in the Arduino IDE and uploaded to the board by USB connection to a computer. So long as the board is connected to a 5V battery/power supply, it will infinitely run the code that has been most recently uploaded to it. The board has 14 digital input/output pins, six of which can be used as PWM (Pulse Width Modulation) outputs, 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button [3]. Pulse width modulation is a powerful technique for controlling analog circuits with a microprocessor's digital outputs [4]. The board includes an 8-bit ATmega328P high-performance, low-power microchip.

1.4.2 HC-SR04 Ultrasonic Sensor

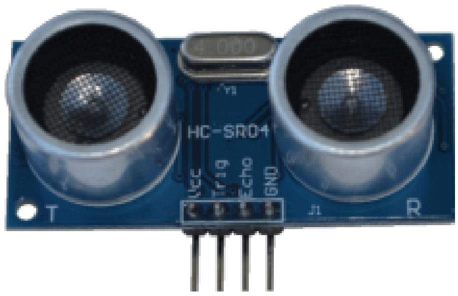


Figure 1.3: HC-SR04 Ultrasonic Sensor [5]

The HC-SR04 seen in Figure 1.3 is a low-cost ultrasonic distance sensor that converts electrical signals into ultrasonic sound waves. It then receives these waves should they rebound from an obstacle. There are four pins on the sensor, TRIG, ECHO, VCC and GND. To operate, the VCC pin must be connected to the 5V source on the microcontroller, the GND pin must be connected to the microcontroller's GROUND, and the TRIG and ECHO pins must be connected to the digital I/O (Input/Output) pins on the board. When the TRIG pin is set to HIGH for 10µs, 8 pulses of 40KHz sound waves are emmited from the sensor [5]. Following this, the ECHO pin will go HIGH until the same set of pulses return to the receiver on the sensor, at which point the pin will return to LOW. If there are no nearby objects, the ECHO pin times out to LOW after 38µs. The duration for which the ECHO pin was at HIGH is the time that the waves took to travel to an object and back. This is the time that is used to calculate the distance.

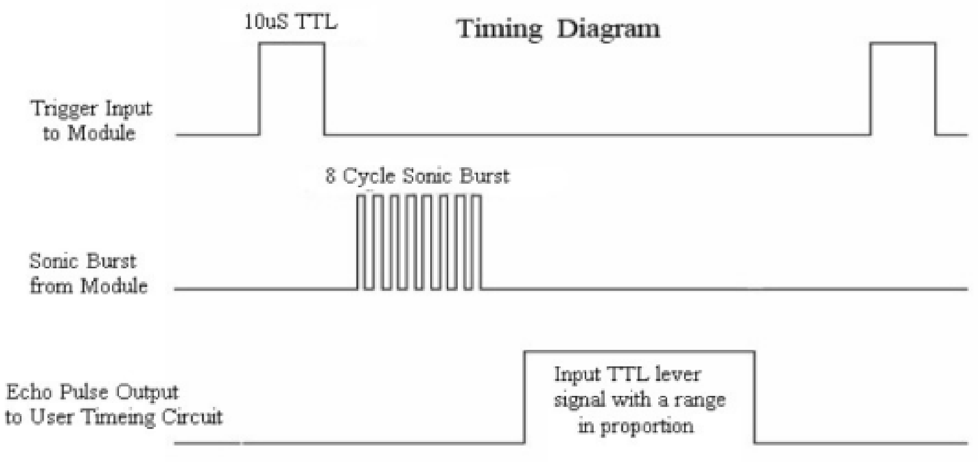


Figure 1.4: HC-SR04 Timing Diagram [5]

HC-SR04 Specifications [6]:

- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2cm - 400 cm
- Resolution: 0.3 cm
- Measuring Angle: 30°
- Trigger Input Pulse width: 10µs
- Dimension: 45mm x 20mm x 15mm
- Weight: ∼10g

Chapter 2: **Introduction**

2.1 Aims & Goals

The main goal of this project is to assist visually impaired people while they are navigating throughout daily life. This can be achieved in an effective manner by making use of ultrasonic sensors. The sensors can essentially replace the user's eyes and detect obstacles in their stead. This particular solution is quite intuitive and can aid a blind person in navigation with very little training time. A user simply places the helmet on their head and walks in a given direction. Once they hear a beep from a given direction, they are instantly aware that there is an obstacle nearby. When the user alters their course to avoid the obstacle, the beeping concludes - given they have successfully navigated past it.

My aim is to develop a helmet with three ultrasonic sensors, one pointing forwards, one pointing to the left, and one pointing to the right. This should provide the user with a sufficiently wide field of view as each sensor has an effectual cone of view of around 30°. It is my intention to achieve this through the use of an Arduino board.

2.2 Arduino Technology

Arduino boards are open-source, programmable microcontrollers that electronics aficionados around the world use to develop all types of hardware projects. The boards can be used to process information from a plethora of input devices such as sensors, buttons, switches [7], and turn it into information to be sent to output devices such as graphic displays, buzzers, speakers, servos, motors etc. The software written for Arduino boards is written in a dialect of C++ [8] and is quite accessible for any level of programming skill.

2.3 Motivation

It is extremely important for technology to be utilised for the benefit of those that are not fortunate enough to have unimpaired vision. Whether the person has impaired vision or is completely blind, solutions like the one that has been developed throughout the course of this project can help them to move around day to day in a safe manner. Historically, blind people have availed of more simple solutions such as a white cane [9] to survey the area in front of them or a trained guide dog to lead them around obstacles [10]. While these solutions have been tried and tested and are very successful, recent advancements in technology may provide a new solution that visually impaired individuals can place more trust in.

Chapter 3: **Related Work and Ideas**

Over 200 million people worldwide suffer from visual impairment [11]. Around 40 million of these people are completely blind, while the others have moderate to severe visual impairment (MSVI). While the majority of blind persons use the traditional white cane or a guide dog to navigate, recent advancements in technology may allow for a more effective solution. Artificial intelligence has made image processing for obstacle avoidance more accessible than ever.

3.1 The NavBelt & The GuideCane

The NavBelt and the GuideCane [12] are two proposed solutions to this issue, both invented at the University of Michigan's Mobile Robotics Lab between 1989 and 1995. Both devices use obstacle avoidance systems that were originally developed for robots.

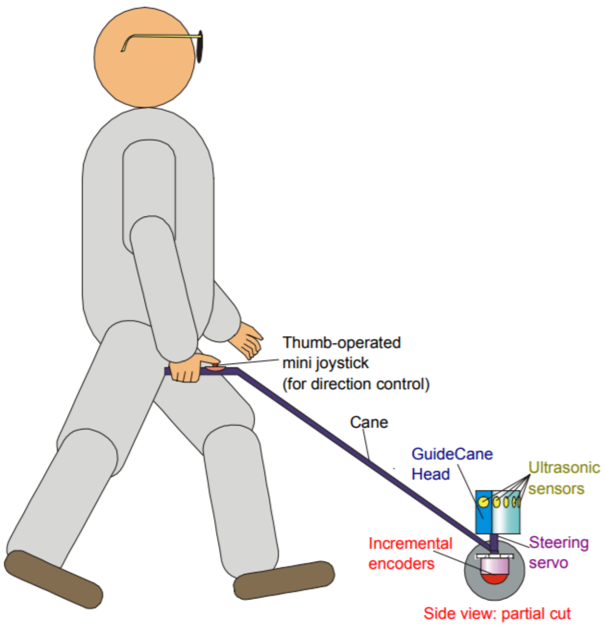


Figure 3.1: The GuideCane [12]

The NavBelt is a wearable belt with ultrasonic sensors that is connected to a small computer which is worn like a backpack. Signals from the sensors are processed by the computer, obstacle avoidance algorithms are applied and the computer relays the information to the user via stereophonic headphones. The disadvantage of this solution is that the user is required to spend at least 40 hours training to be able to walk at a normal pace using the NavBelt. The reason for so much training is that the information being relayed to the user through stereophonic headphones varies in volume, pitch, transmission rate and the direction from which the audio is coming. The user would have to learn what a variation in each of these characteristics corresponds to before they could use the NavBelt as an aid for walking at a normal rate. Also, the NavBelt does not have sensors facing up/down to detect low-hanging obstacles such as signposts or sudden drop-offs such as kerbs.

On the other hand, the GuideCane, as seen in Figure 3.1, is essentially a "robotic guide-dog" [12]. The user holds the cane in front of them as they would with a regular white cane, except the base of it rolls on wheels to support its weight. There are 10 ultrasonic sensors on the GuideCane, which relay information to a servomotor that steers the wheels right and left to avoid obstacles. The user simply follows the direction that the wheels are facing and walks as normal. This is a more intuitive solution than the NavBelt, the user does not need to spend any time training to use it. However, as with the NavBelt, the GuideCane fails to detect low-hanging obstacles.

3.2 Microsoft Kinect Depth Sensor

A more modern proposal for this problem was made by Filipe et al. in 2012 [13]. In this study, the Microsoft Kinect sensor [14] which is comprised of an RGB camera as well as a depth sensor (by means of pairing an IR emitter with an IR camera) was used to detect obstacles for visually impaired persons, and notify them when there was an obstacle within 2m.

Network outputs	Targets (correct outputs)				
	No obstacle	Obstacle	Upstairs	Downstairs	Total
Obstacle	205 (28.7%)	0 (0%)	1 (0.1%)	0 (0%)	**99.3%**
No obstacle	0 (0%)	152 (21.3%)	2 (0.3%)	2 (0.3%)	**98.1%**
Upstairs	2 (0.3%)	0 (0%)	222 (31.9%)	0 (0%)	**99.1%**
Downstairs	0 (0%)	0 (0%)	0 (0%)	128 (17.9%)	**100%**
Total	**98.7%**	**100%**	**98.7%**	**98.5%**	**99%**

Figure 3.2: Confusion Matrix [13]

Figure 3.2 illustrates the confusion matrix obtained by this study when a neural network was trained on a dataset of images and tested on 714 unseen inputs. The Kinect sensor was able to detect obstacles of each variety with impressively high accuracy. This solution drastically improves upon that of the NavBelt and GuideCane as it can detect drop-offs such as downward staircases with an accuracy of 100%.

The shortcomings of this proposal, as outlined by the authors themselves, involve the lack of mobility (as the Kinect must be connected to a computer) and the failure to detect obstacles that are in direct, bright sunlight and/or covered by water [13]. This calls for a more feasible solution whereby the user can easily and comfortably carry around a device that will aid them in navigating around obstacles in any area.

3.3 Smartphone-based Obstacle Avoidance System

Another solution to this problem is to employ a device that everyone already has access to nowadays, smartphones. This eliminates the need for additional (possibly quite expensive) equipment as in the previous solutions.

In a 2017 study, Lin, Lee and Chiang [15] designed a smartphone-based guidance system for the visually impaired. It involves the user's smartphone continuously capturing the scene in front of them, processing the image data, then returning information back to the user by voice notification.

This system not only tells the user how far the obstacle is away and the direction it is in, but also what kind of obstacle it is. It makes use of a deep learning algorithm to process the images captured by the phone to recognise objects of various types. When the smartphone has access to a 4G or WiFi connection, it sends the image data to a remote server with a GPU that is more powerful than that of the smartphone for faster processing and recognition. When the smartphone cannot access the internet, it relies solely on the power of the smartphone to process the images and repeatedly tries to reconnect to the server. This results in slower image processing and obstacle recognition.

This solution would be ideal for a visually impaired individual who frequents built-up urban areas as there is usually a stable 4G connection in those areas. This would allow the system to avail of the processing power of the server's GPU at all times. However, if this is not the case and the individual rarely has access to the internet, they would need a very powerful smartphone to match the "online" mode's performance. If the smartphone is not powerful enough, this could potentially lead to misinformation being relayed to the user or late notifications about obstacles which could result in serious harm or injury.

Furthermore, continuously capturing image data and processing it (or sending it to the server to be processed) could be very heavy on battery consumption. This could limit the duration that this system could be used for. For example, if this activity drains a typical smartphone battery in a matter of 1-2 hours, the user would be left without guidance once the phone dies. Thus, this solution would not be suitable for visually impaired persons who regularly embark on journeys longer than the duration for which their phone would be able to run this system. This could be overcome by a smartphone with a larger battery or by the use of portable phone charger(s) that can supply power to the phone on-the-go.

Another drawback to this system is that it would not be immediately obvious to other pedestrians that the user is visually impaired. When an individual uses the traditional white cane, it is evident that they have some form of visual impairment and other pedestrians are made aware. This leads to the user being given more room to manoeuvre and prevents them from being bumped into as they walk. Using this system in conjunction with the white cane could lead to the user being more confident when navigating urban areas.

This system could be improved by using a more computationally powerful server and by training the deep learning algorithm on a more extensive assortment of obstacle types.

Chapter 4: **Outline of Approach**

4.1 Core Idea: Ultrasonic Waves

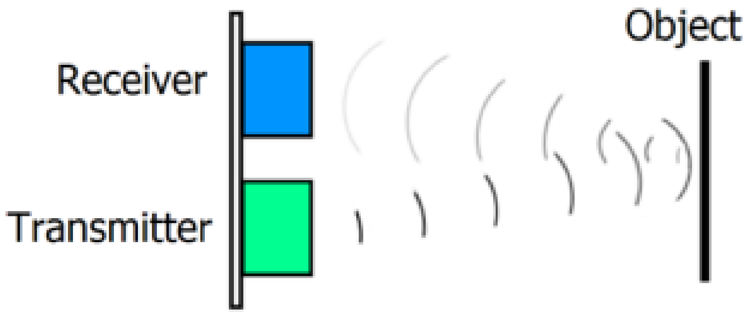


Figure 4.1: Ultrasonic waves rebounding to sensor [16]

The main idea behind this project relies on the use of ultrasonic waves. Ultrasonic waves are sound waves with a frequency greater than 20KHz. They travel at the speed of sound. As the speed of sound is a known value, these waves can be used to calculate distances as seen in the diagram in Figure 4.2. The HC-SR04 sensors emit high-frequency sound (approximately 40KHz [5]) above the frequency that humans are capable of hearing. The sensors then listen out for sounds that match this frequency to bounce off surfaces and return to the sensor. Using the speed of sound v, and the time the wave took to return to the sensor t, the distance to the object d can be calculated using equation 4.1:

$$d = \frac{v \times t}{2} \tag{4.1}$$

As v and t are known in this equation, d can be calculated very easily. The distance is divided by two because the time t is the total time that the sound wave took to travel to the surface and back to the sensor. The distance we are interested in is half of this, the distance from the sensor to the object.

This design choice was made due to the fact that using ultrasonic sensors is a computationally and financially inexpensive way to measure distance. Once this simple distance calculation above is made, the result can be used to make a decision and pass a signal to its corresponding buzzer if necessary.

4.2 System Design

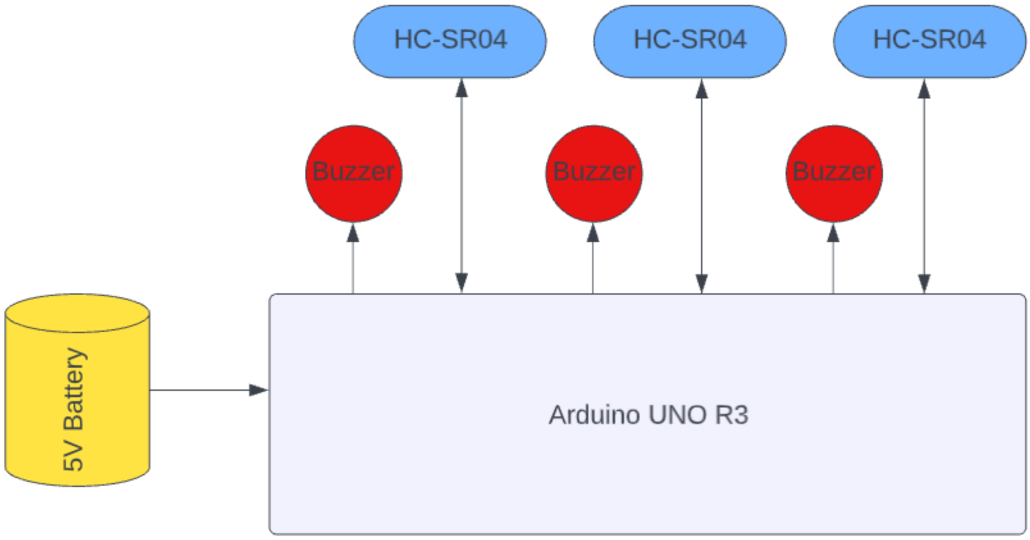


Figure 4.2: System Architecture

The simplified system architecture of this device can be seen in Figure 4.2. The microcontroller board is connected to a 5V battery via a USB connection. Three HC-SR04 sensors and three buzzers are connected to the microcontroller via DuPont wires shown in Figure 4.3. As there were limited ground and power ports on the microcontroller, the ground wires and power wires for all three sensors and buzzers were stripped and connected into one respective wire each using terminal block connectors seen in Figure 4.4 so that they could be connected to the microcontroller without the need for a breadboard. With this design, the system is fully portable when powered by a battery. The next step was to attach the board, battery, sensors and buzzers to a helmet to create a mobile, usable device that a visually impaired individual could simply place on their head and begin walking. The sensors would have to be carefully placed so that there was minimal overlap between the sensors' fields of view while making sure there were no blind spots either.

Figure 4.3: Male-to-Female DuPont wires

Figure 4.4: Terminal block wire connector

Software was implemented specifically for this system which measures the distance at each sensor and sends a signal to the corresponding buzzer if the distance is within a threshold. This threshold can be chosen to suit the user through a global variable that is set within the code. Once the code was written, the board was connected to a computer and the code was uploaded onto it. This code is kept on the board even when it is powered off using the Arduino's non-volatile flash memory [17].

Chapter 5: **Implementation**

5.1 Software Implementation

The software for this project was developed using the Arduino IDE [18]. In the code, which can be found here [19], there are three functions being called infinitely in the main function *loop()*. The first of which, *measure()*, is called three times, once for each directional sensor. This function works by setting the trigPin LOW for 2μs to make sure it is not firing beforehand, then setting it HIGH for 10μs to release the ultrasonic pulses. It is then set back to LOW. The duration for which the echo pin is HIGH for, i.e. the time it is waiting for the pulse to come back, is found with the built-in function *pulseIn()*. This value is then multiplied by the speed of sound and divided by two to find the distance from the sensor to the obstacle. The value that it returns is the distance that has been calculated from that particular sensor's measurement and is saved to a corresponding variable. Once all three distances have been calculated, the *findMinimum()* function finds the smallest of the three so that the nearest obstacle can be prioritised.

Algorithm 1 measure(trigPin, echoPin) <- passed pins for each sensor

```
write LOW to the trigPin
wait 2μs
write HIGH to the trigPin
wait 10μs
write LOW to the trigPin
duration <- pulseIn(echoPin, HIGH) - times how long echoPin was high
distance <- (duration*0.0343)/2
return distance
```

Next, the *beep()* function seen below in Algorithm 2 is called to decide which buzzer should beep, if any. This function is composed of a series of *if* statements which check whether or not the distances at each sensor have come within the threshold. A buzzer will only be told to beep if the distance is less than the threshold *AND* matches the *minDistance* found in the previous step. This ensures that there are only beeps coming from one direction at a time - the direction of the nearest obstacle. Having more than one buzzer going off at any given time would lead to great confusion.

Algorithm 2 beep()

```
if distanceMid < distanceThreshold AND distanceMid == minDistance then
    send signal to middle buzzer
end if
*repeat for left and right*
```

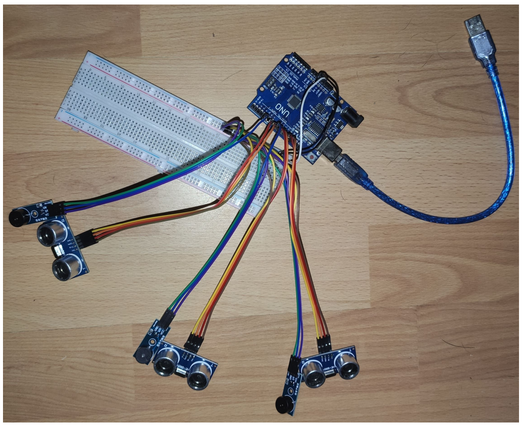
Finally, the delay before the next loop begins is set with the following line of code:

$$delay(5 * minDistance));$$

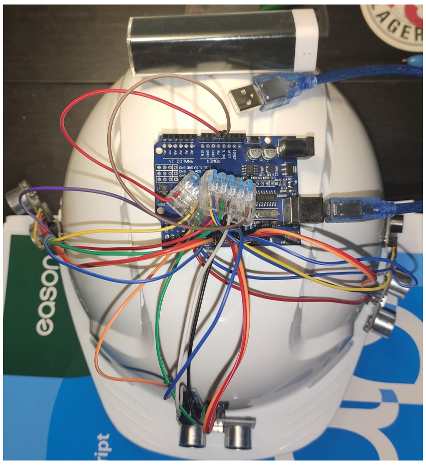
When there is an obstacle within the threshold, this delay corresponds to the time between beeps. The delay is scaled by the *minDistance* value so that the time between beeps is proportional to the distance the object is away. A value of 5 was chosen after some trial and error. This produces a greater sense of awareness and notifies the user that a certain obstacle is getting closer as the beeps get more frequent.

5.2 Hardware Implementation

5.2.1 Experimental Hardware



(a) Testing hardware



(b) Top view

Figure 5.1: Hardware during testing

To test the software, the above hardware in Figure 5.1a was implemented. Pictured here are the Arduino microcontroller, the three HC-SR04 sensors, and the three buzzers connected via a breadboard for experimentation. Once functional, the system was attached to a helmet and powered by the battery seen in the top of Figure 5.1b. At this stage, the side sensors were moved around to find positions that minimised overlaps and blind spots. Once finalised, the respective positions were marked on a different helmet (pictured below in Fig. 5.2) and holes were drilled to fit the sensors through from the inside.

5.2.2 Final Hardware

(a) Front view

(b) Top view

Figure 5.2: Front and top view of hardware

Once the hardware was fully functional, the physical elements of this device were attached to a construction worker's hardhat, as seen in Figures 5.2a and 5.2b. Holes were drilled in the correct locations on the front, left, and right of the helmet so that the sensors could be wired from the inside. The Arduino microcontroller was attached to the topside of the inner hardhat suspension, away from the user's head. The sensors were placed in the drilled holes from the inside and the buzzers were fixed to the inside of the helmet using double-sided tape. The 5V battery was also attached inside the helmet and connected to the board.

Chapter 6: **Results of Selected Approach**

6.1 The Deliverable

(a) Device being worn

(b) Inside of helmet

Figure 6.1: Final Deliverable

The main deliverable of this project is the helmet pictured above in Figure 6.1. This helmet has three HC-SR04 ultrasonic sensors configured with three corresponding buzzers attached within the helmet. The rear strap seen in Figure 6.1a can be adjusted to fit heads of any size, so it is suitable for any person with impaired vision. When worn correctly, the sensors point directly forward, to the left, and to the right of the user. If there are obstacles within the threshold in all 3 directions, the device will prioritise the closest one and only the buzzer for that direction will beep. The inside of the helmet can be seen in Figure 6.1b, with all components of the device attached to the topside of the inner suspension, away from the user's head. The microcontroller and battery are fixed to the inner suspension so that they cannot come loose and move around within the helmet. The sensors and buzzers are fixed to the inside of the helmet above the left and right ears, and above the forehead.

6.2 Overall Results

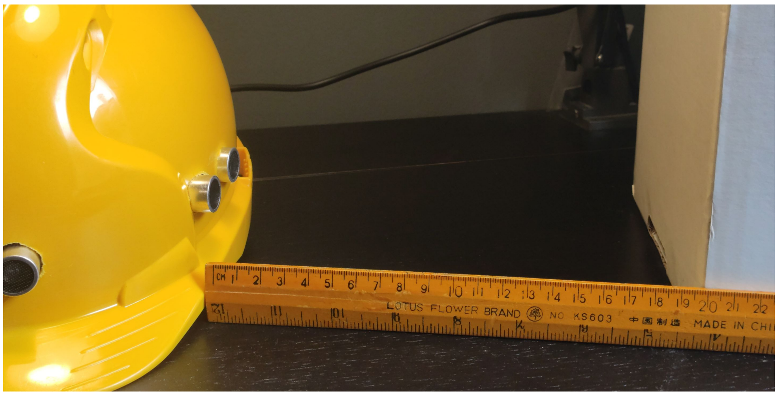
The result of this project is a device that can assist the visually impaired in avoiding obstacles. The user places the helmet on their head and walks in the desired direction until they come across an obstacle. The buzzers will notify the user that the obstacle is there. The frequency of the beeps and the direction from which they are coming lets the user know how far away the obstacle is and in what direction it is. If there are multiple obstacles within the pre-set threshold, the device will only tell the buzzer that is in the direction of the nearest obstacle to beep. This avoids confusion for the user - if multiple buzzers were beeping, it would become almost impossible to distinguish between them.

The device was tested by putting obstacles in front of, and to the left of the helmet. As seen below in Figure 6.2, when there were no obstacles within a threshold of 30cm, the software printed the following message to the serial monitor in the Arduino IDE when the device was connected to a computer.

```
No obstacles within threshold!
No obstacles within threshold!
No obstacles within threshold!
```

Figure 6.2: Serial monitor with no obstacles

When an obstacle was placed 20cm to the left of the device as pictured in Figure 6.3a, the software printed the message seen in Figure 6.3b to the serial monitor.



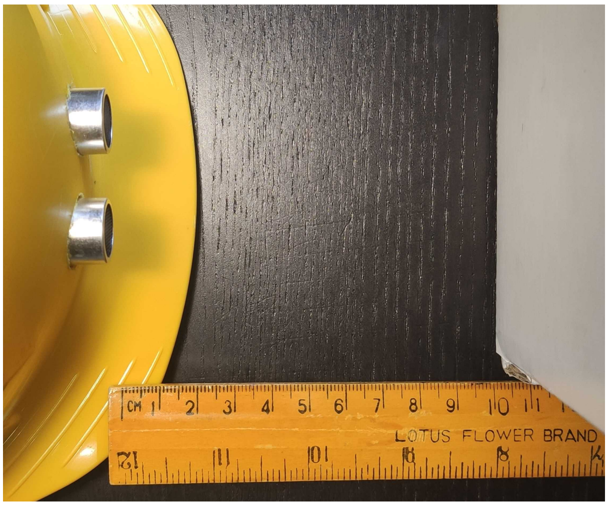
(a) Obstacle 20cm to the left

```
Nearest obstacle is to the left! (20.15 cm away)
Nearest obstacle is to the left! (20.15 cm away)
Nearest obstacle is to the left! (20.15 cm away)
```

(b) Serial monitor with an obstacle 20cm to the left

Figure 6.3: Testing with an obstacle 20cm to the left

When an obstacle was placed 10cm in front of the device as pictured in Figure 6.4a, the software printed the message seen in Figure 6.4b to the serial monitor.



(a) Obstacle 10cm in front

```
Nearest obstacle is in front! (10.03 cm away)
Nearest obstacle is in front! (10.03 cm away)
Nearest obstacle is in front! (10.03 cm away)
```

(b) Serial monitor with an obstacle 10cm in front

Figure 6.4: Testing with an obstacle 10cm in front

Finally, when there was an obstacle 10cm in front of the device and an obstacle 20cm to the left of the device simultaneously, only the front buzzer beeped and the serial monitor showed that the device was only reporting on the obstacle that was closer.

Chapter 7: **Project Workplan**

7.1 Gantt Chart

The Gantt chart in Figure 7.1 outlines how this project will be approached. I have outlined the following tasks in the order that I plan on completing them:

- Data Research: Search for open-source data sets to test the proposed solution on.
- Data Collection & Cleaning: Collect data found in the first task and pre-process it.
- System Architecture: Outline system architecture, specifications and requirements. Plan implementation.
- System Implementation: Develop proposed system that can take in the data and use it to perform obstacle detection and avoidance.
- Testing & Evaluation: Test the proposed solution on the data obtained. Test physically with the sensor. Evaluate the results.
- Report: Finish writing the final draft of the report.

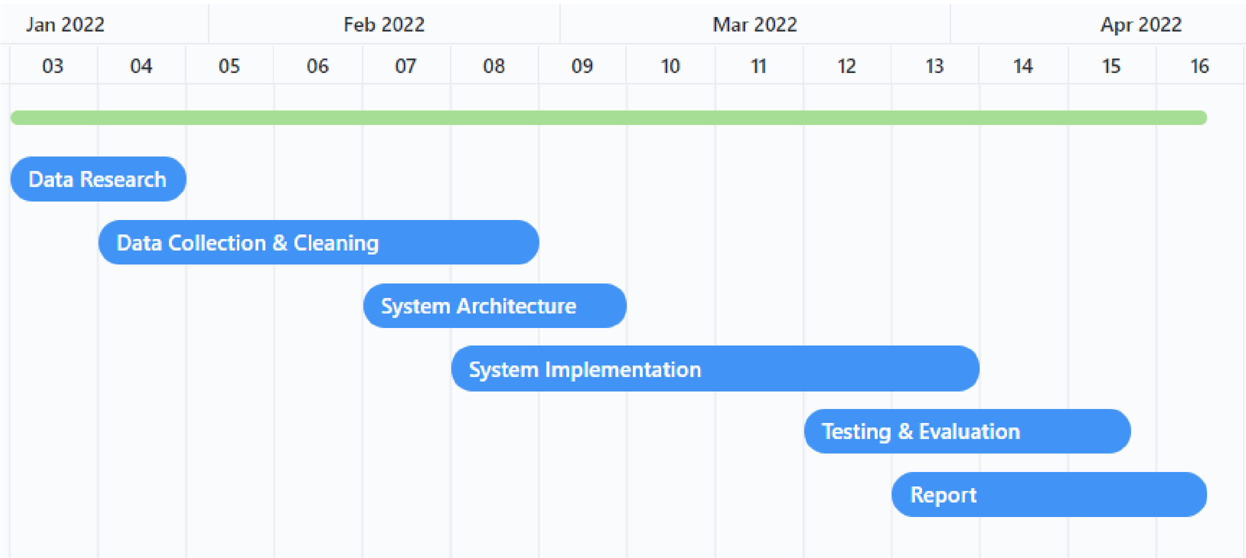


Figure 7.1: Gantt Chart for Project Workplan.

Chapter 8: **Future Work & Other Applications**

8.1 Future Work

Following the study presented in this report, there are some improvements to the proposed device that could be made in the future with extended time and resources. More powerful, more expensive ultrasonic sensors with higher frequency could be used to increase the range and/or field of view of the device. With this addition, a battery with a higher capacity would also have to be added. As the HC-SR04 have a wide field of view but not a very tall one, additional sensors could be added beside each of the existing ones, rotated by 90° to detect obstacles above and below the vertical limits of the current sensors. Additionally, storage could be added to the system to collect the data that the sensors are reading in and the decisions that are being made. This data could be analysed to ensure that the device is working to its highest capacity.

An addition that would be very beneficial to this system is mobile app integration. Adding the WiFi module *ESP8266* [20] to the device would mean that it could be connected to a mobile hot-spot. Through this channel of communication, it would be possible to send voice notifications to the user via an app and notify them of upcoming obstacles to their headphones. Also, cellular capabilities could be added to the device using the Arduino GSM Shield [21]. If very small distances or zero centimeters were being read in for a prolonged period of time, the user may be in trouble - they might have fallen or hit their head. In this scenario, the GSM module could be used to send cellular alerts (text messages) to their next of kin or their local emergency services. If the *SKG13BL* GPS module [22] was added, the user's location could also be sent in this situation.

To find any weaknesses to eliminate and improvements to make, this device could be put to the test with live human trials. The average training time could give an indication to how long it takes a user to build up confidence and trust in the device. Through this kind of testing, unforeseen issues may be found and fixed if possible.

With some modifications to the software, a feature that calculates the speed of obstacles coming towards the user could be added to this system. By measuring distances at regular intervals and finding the difference between readings, the distance the obstacle has travelled in that regular interval is obtained. By simply dividing this distance by the interval, you calculate the speed at which the obstacle is travelling relative to the user. With this information, the frequency of the beeps could be further increased when an object is moving towards the user at high speed. Of course, if the user and the object are moving towards each other, this would calculate a relative velocity that is higher than the object's true velocity.

8.2 Other Potential Applications

Taking inspiration from the nature of the helmet, this device could also be put to use by construction workers who would already be wearing a hardhat. In a scenario where a worker is not paying attention and there is a girder or some other object being lifted towards them, the beeping would alert them to it and they could move out of the way if necessary. This may involve the addition of sensors pointing behind the user so that even if they are paying attention to their surroundings, they could be notified of objects approaching them from behind.

Another application where this solution may prove useful is on the helmet of firefighters. Frequently, firefighters are tasked with moving through very smoky buildings with very low visibility. Without knowing the layout of the building, it may be hard to avoid walking into walls etc. Having this device on their head may assist them in navigating through this kind of scenario in the same way that it assists visually impaired individuals.

Using the HC-SR04 sensors, you can also design a device that works very similarly to this one which measures the speed of objects. As described in Section 8.1, the speed of an object can be measured by taking the difference between two distance readings and dividing by the time between them being made. This would only measure the true velocity of an object if the device was stationary. However, these particular sensors do not have a very extensive range, so more powerful sensors would need to be used for measuring speeds of objects more than a few meters away.

Chapter 9: **Summary and Conclusion**

9.1 Summary

The proposed device achieves the goals that were set out at the beginning of this project. The aim was to develop a device that is capable of helping the blind or visually impaired to avoid walking into obstacles. This is precisely what the device does.

Through the use of HC-SR04 ultrasonic sensors, this device can detect obstacles in the user's path and alert the user to them using buzzers. When a user is wearing the device and an obstacle comes within the pre-set threshold in front of one of the sensors, the buzzer for that direction will start beeping. As the user gets closer to the obstacle, the beeping will speed up. If the obstacle is in front of the user and they alter their course to the left slightly, the right sensor will pick up on the obstacle and the user will know that they are no longer heading straight towards it.

This project acts as a proof of concept for a solution to the problem of visual impairment. As outlined in Section 8.1, range, field of view and battery life are all areas where this device could be improved by using more powerful sensors and larger batteries.

9.2 Conclusion

In conclusion, this project was a great success. The solution that has been developed proves that ultrasonic sensors can be used to assist the navigation of the visually impaired by helping them to avoid obstacles. The sensors measure the distance to the nearest obstacle with relatively high accuracy, the board processes this information and sends a signal to the corresponding buzzer should the distance be less than the pre-set threshold. The buzzers notify the user of the upcoming obstacles, allowing them time to alter their course and avoid them.

Acknowledgements

I would like to express my gratitude to my academic supervisor, Professor Tahar Kechadi, for his guidance throughout this project.

I would also like to thank my father, for gifting me my first Arduino starter kit a number of years prior to this project.

Bibliography

1. Boskany, N. & Wady, S. *DATA CENTER ENVIRONMENT MONITORING AND COOLING SYSTEM BASED ON ARDUINO AND GSM NETWORK* in (Apr. 2019), 10–17.
2. *Arduino UNO R3 microcontroller documentation* https://docs.arduino.cc/hardware/uno-rev3.
3. T.L, A., C.O, O. & C.E, E. GSM based Smart Security System using Arduino. *IJARCCE* **8,** 32–42. https://ijarcce.com/wp-content/uploads/2019/10/IJARCCE.2019.81005.pdf (2019).
4. Barr, M. Pulse width modulation. *Embedded Systems Programming* **14,** 103–104 (2001).
5. Morgan, E. J. HC-SR04 ultrasonic sensor (2014).
6. *HC-SR04 User Manual* https://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf.
7. Badamasi, Y. A. *The working principle of an Arduino* in *2014 11th international conference on electronics, computer and computation (ICECCO)* (2014), 1–4.
8. Stroustrup, B. *The C++ programming language: Pearson Education India* 1995.
9. Strong, P. The history of the white cane. *Paths to Literacy* (2009).
10. Whitmarsh, L. The benefits of guide dog ownership. *Visual impairment research* **7,** 27–42 (2005).
11. Ackland, P., Resnikoff, S. & Bourne, R. *World blindness and visual impairment: despite many successes, the problem is growing* 2017. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820628/.
12. Shoval, S., Ulrich, I., Borenstein, J., *et al.* Computerized obstacle avoidance systems for the blind and visually impaired. *Intelligent systems and technologies in rehabilitation engineering,* 414–448 (2000).
13. Filipe, V. *et al.* Blind navigation support system based on Microsoft Kinect. *Procedia Computer Science* **14,** 94–101 (2012).
14. Zhang, Z. Microsoft kinect sensor and its effect. *IEEE multimedia* **19,** 4–10 (2012).
15. Lin, B.-S., Lee, C.-C. & Chiang, P.-Y. Simple smartphone-based guiding system for visually impaired people. *Sensors* **17,** 1371 (2017).
16. Zhmud, V., Kondratiev, N., Kuznetsov, K., Trubin, V. & Dimitrov, L. *Application of ultrasonic sensor for measuring distances in robotics* in *Journal of Physics: Conference Series* **1015** (2018), 032189.
17. Monk, S. *Programming Arduino: getting started with sketches* (McGraw-Hill Education, 2016).
18. *Arduino integrated development environment software:* https://www.arduino.cc/en/software.
19. *My private Gitlab Code Repo (made available to academic supervisor and moderator):* https://csgitlab.ucd.ie/ZachD99/fyp_obstacle_avoidance.
20. Mehta, M. ESP 8266: A breakthrough in wireless sensor networks and internet of things. *International Journal of Electronics and Communication Engineering & Technology* **6,** 7–11 (2015).
21. Nayyar, A. & Puri, V. *A review of Arduino board's, Lilypad's & Arduino shields* in *2016 3rd international conference on computing for sustainable global development (INDIACom)* (2016), 1485–1492.

22. *SKG13BL Global Positioning System Module* https://studylib.net/doc/18129796/gps-module-datasheet.