Vision For the Blind

ARSHAM ATIGHECHI
HUSSEIN HAIDARI
Abstract

How can we improve the living standards of the visually impaired using an Arduino?

Living with visual impairment could potentially be one of the hardest things one could do. Constantly having to worry about obstacles and carrying a stick to feel your way forward. In order to combat this difficulty, we have designed Vision for the Blind.

One of the usages of ultrasonic sensors are to measure the distance from objects using sound waves. These sensors, in combination with Piezo buzzers, have been used in Vision for the Blind to warn the user of incoming obstacles by emitting sound from the buzzers. The volume of the buzzers is controlled by using a digital potentiometer and varies depending on the measured distance. All three are connected to an Arduino which has been coded to perform the given task.

keywords: Mechatronics, ultrasound sensors, buzzers, distance measurement, digital potentiometer.
Referat

Hur kan vi öka levnadsstandarden för visuellt nedsatta människor med hjälp av en Arduino?

Att leva med visuell nedsättning är självklart ett stort problem som medför svårigheter, att alltid vara orolig över hinder som kan finnas i vägen eller bära en pinne med sig för att hitta sin väg. För att bekämpa de här svårigheterna har vi konstruerat Vision for the Blind.


Nyckelord: Mekatronik, ultraljuds sensorer, sumrar, avståndsmätning, digital potentiometer.
Acknowledgements

We would like to thank Nihad Subasic for always being available for assistance throughout the project. We would like to thank Staffan Qvarnstrom for handling all the orders placed. Finally, we would like to thank the teaching assistants who have been available in the workshops with help during tough times, with tips, brain storming and general knowledge.

Arsham Atighechi, Hussein Haidari
Stockholm, May, 2018
# Contents

Abbreviations .......................................................................................................................... 1

1 Introduction .......................................................................................................................... 1
   1.1 Background ..................................................................................................................... 1
   1.2 Purpose .......................................................................................................................... 2
   1.3 Scope .............................................................................................................................. 2
   1.4 Method ............................................................................................................................ 2

2 Theory .................................................................................................................................. 3
   2.1 Ultrasound ....................................................................................................................... 3
      2.1.1 HC-SR04-Ultrasonic ranging module ....................................................................... 3
   2.2 The Piezo material ......................................................................................................... 4
      2.2.1 Piezo Buzzer .............................................................................................................. 4
   2.3 Microcontroller Unit ....................................................................................................... 4
   2.4 Potentiometer ................................................................................................................ 5
      2.4.1 Digital Potentiometer .............................................................................................. 5

3 Demonstrator ...................................................................................................................... 6
   3.1 Arduino .......................................................................................................................... 6
   3.2 Software ......................................................................................................................... 6
      3.2.1 Code ......................................................................................................................... 7
      3.2.2 The Newping Library ............................................................................................. 7
   3.3 Sensors .......................................................................................................................... 8
   3.4 Volume Control ............................................................................................................. 8
      3.4.1 Volume Control with Digital Potentiometer ............................................................. 8
   3.5 Instrument Case ............................................................................................................ 9
      3.5.1 Front of case ............................................................................................................ 9
      3.5.2 The Harness ............................................................................................................ 10
      3.5.3 The Cap and Buzzers ............................................................................................. 11
   3.6 Battery .......................................................................................................................... 12

4 Results and testing ............................................................................................................. 13
   4.1 Testing ........................................................................................................................... 13
      4.1.1 Practical tests .......................................................................................................... 13
List of Figures

2.1 Field of view [19] ................................................. 4
3.1 Flowchart of software, created in draw.io ........................ 7
3.2 Sensors as seen from above, with $\alpha = 20^\circ$. Each pair of cylinders represents one sensor, created in Solid Edge ST9 ............................... 8
3.3 Front of the case .................................................. 10
3.4 The Harness ....................................................... 11
3.5 The Cap as seen from different angles ............................ 12
4.1 Measuring at 120 cm, created using Excel ......................... 15
4.2 Measuring at 240 cm, created using Excel ......................... 15
4.3 Measuring at 399 cm, created using Excel ......................... 16
4.4 Vertical angle of sensors as seen from the side, created in Krita 17
4.5 Horizontal angles of sensors as seen from above, created in Krita 18
B.1 With movement, distance at 120 cm, created using Excel ........ 30
B.2 Without movement, distance at 120 cm, created using Excel ........ 31
B.3 With movement, distance at 240 cm, created using Excel ........ 32
B.4 Without movement, distance at 240 cm, created using Excel ........ 33
B.5 Without movement, distance at 399 cm, created using Excel ........ 34
C.1 Completed project ................................................. 35
C.2 Progress picture 1 ................................................. 36
C.3 Progress picture 2 ................................................. 37
C.4 Progress picture 3 ................................................. 38
C.5 Progress picture 4 ................................................. 39
C.6 Progress picture 5 ................................................. 40
C.7 Progress picture 6 ................................................. 41
C.8 Progress picture 7 ................................................. 42
C.9 Progress picture 8 ................................................. 43
C.10 Progress picture 9 ............................................... 44
C.11 Progress picture 10 ............................................. 45
C.12 Progress picture 11 ............................................. 46
C.13 Progress picture 12 ............................................. 47
C.14 Progress picture 13 ............................................. 48
List of Tables

3.1 Frequencies for lower buzzers ........................................... 11
4.1 Testing methodology ....................................................... 14
4.2 Questionnaire results ...................................................... 14
4.3 Test 6, vertical FOV ....................................................... 18
4.4 Test 7, horizontal FOV .................................................... 19
4.5 FOV angles ................................................................. 19
D.1 Vertical FOV .............................................................. 51
D.2 Horizontal FOV ............................................................ 51

Abbreviations

AC  Alternating Current
DP  Digital Potentiometer
FOV  Field of View
I/O-pins  Input/Output pins
IDE  Integrated Development Environment
MCU  Microprocessor Unit
SPI  Serial Peripheral Interface
Chapter 1

Introduction

In this bachelor’s thesis the use of sound and distance measurements have been studied closely in order to make them work together to act as a visual aid for the visually impaired. Most components are placed inside or on a small box. The box is then easily carried around by the user whilst simultaneously alerting them of their surroundings using small buzzers.

1.1 Background

Being blind could be a very difficult experience. Living in absolute darkness with very limited external assistance to help you guide you through the obstacles of life.

In order to improve the living standards of the visually impaired, a further look into how their day to day activities and daily lives could be improved, using a microcontroller unit (MCU).

1.2 Purpose

The purpose of this project is to produce a prototype of a device that can detect obstacles which may appear in the visually impaired way using sensors and alerting them using audible cues. The following research questions have been studied:

- How can we use a soundsystem to alert the visually impaired of obstacles in their way?
- How can we make the device practical?
- How reliable is the HC-SR04 ultrasound sensor in this kind of project?
1.3 Scope

Due to limitations set by KTH in the form of both time and budget, the following constraints have been placed on the project.

Five sensors will be used, which serves the purpose of measuring distance. These sensors need to be able to work in conjunction with each other. The price of the ultrasound sensors are quite set in stone and are more expensive compared to the rest of the components needed. Even though more sensors would most likely result in a more sensitive device that can be more reliable for the user, with each addition of another sensor another buzzer needs to be added. This would eventually leave us over budget.

For the sound system, buzzers will be used due to their small size and still being able to produce a loud enough sound, the larger the buzzer the louder the sound which it can produce. The buzzers are also very cheap to acquire as well as being very simple to use.

For the harness something simple and cheap will be used. The purpose of the harness is only to carry the device and is only required to be able to carry that weight as well as being comfortable. The harness will therefore be sown by ourselves, using an elastic material.

1.4 Method

In order to better understand the behaviour of the buzzers they were tested rigorously. What frequencies were the most comfortable to listen to, how fast each buzzer should activate as well as the order of how the buzzers emitted sound. To find the best suitable digital potentiometer, the buzzers were tested with different constant resistances. The resistances were then exchanged to bigger and bigger resistances to see the upper and lower limits of the volume.

The ultrasound sensors were then tested together with the digital potentiometer to make sure that the speed in which the sensors measured the distance was compatible with the digital potentiometer (DP). The speed in which the DP could switch between different channels was also tested.

Once the components worked together the case was constructed. The buzzers were then attached to a cap in specific positions to give the user the best sense of their surroundings.

Finally, the sensors field of view (FOV) and distances were tested to see how large the measuring angles and accuracy were.
Chapter 2

Theory

2.1 Ultrasound

The frequency range of what humans are capable of hearing is between 20 Hz to 20 000 Hz. Any frequency outside of that range will not be recognized as audible sound for humans. Sound waves with frequencies above 20 000 Hz are known as ultrasound [1].

Using ultrasound as a distance measuring tool is very common and has been used in many different fields, due to the high efficiency as well as allowing distance measurement without physical touch [2, 8].

2.1.1 HC-SR04-Ultrasonic ranging module

In order to measure the distance from an object, a HC-SR04 is used. The module uses one ultrasonic transmitter and receiver which, when a short pulse of voltage is applied, sends out 8 waves of ultrasound from the transmitter. These waves travel towards the obstacle and once hit, the echo travels back and is captured using the receiver. Due to the fact that sound travels at a very specific speed in air, we can calculate the distance the sound waves have traveled by using the speed it traveled at and the time it has taken for the echo to return using equation 2.1.

\[
d = \frac{t}{2c}
\]  

In equation 2.1, \(d\) is the measured distance, \(t\) is the time it takes for the sensor to register the echo and \(c\) is the speed of sound in air, which is roughly 344 m/s.

The ultrasound waves will not travel in one straight line. They will travel at an angle from the source of sound. This angle determines the FOV, which is the surface area where the sensors can register objects and determine the distance from the object. This means that the sound waves will cover an increasingly larger surface area until they reach the maximum range of roughly four meters [2, 11].
The waves are sent out in a conical shape according to Figure 2.1 [16, 17].

![Figure 2.1: Field of view][19]

### 2.2 The Piezo material

A few materials are known to be Piezo materials. These materials have a tendency to behave differently once an electric field is applied to them. One of the properties are very similar to applying stress to any material. With enough voltage, a large enough electric field is produced which will cause the Piezo material to deform in conjunction with the electric field.

Another property of the Piezo material is the reverse of the previous property. If you quickly snap or deform a Piezo material a large surge of electricity will jolt through the material which can be used to quickly produce i.e sparks [3, 12, 18].

#### 2.2.1 Piezo Buzzer

The Piezo buzzer uses a Piezo ceramic material. Once applied with an Alternating Current (AC) the Piezo buzzer will deform in conjunction with the frequency of the AC. When the material inside the buzzer deforms back and forth, a sound is produced. The pitch of the sound that is made will change together with increasing or decreasing frequency of the AC. A higher frequency will produce a higher pitch of sound while a lower frequency will give a lower pitch of sound [3].

### 2.3 Microcontroller Unit

In order to make all these different components work together a microcontroller unit (MCU) is used. These come in different forms and sizes, Raspberry Pi, BeagleBoard, Arduino, to name a few. In this project an Arduino Uno is used.
2.4. POTENTIOMETER

An MCU is a small microchip which is programmable through specific software provided by the manufacturer. In the case of Arduino the software is called Arduino Integrated Development Environment (IDE). The Arduino Uno is programmable using the C language which is a very fast and memory efficient programming language, while at the same time being very versatile [5].

2.4 Potentiometer

One of the most fundamental laws of electricity is Ohms Law which describes the relationship between voltage, current and resistance [4].

\[ U = RI \]  \hspace{1cm} (2.2)

Equation (2.2) states that with a increasing, or decreasing resistance, the voltage will either decrease or increase, given that the current stays constant.

To achieve a varying resistance a potentiometer can be used. The potentiometer is a resistance that has a knob on it. The knob can be turned to increase or decrease the value of the resistance.

This is very applicable for audio use. By controlling the resistance the current that goes into the circuit can be adjusted due to the fact that the current stays near constant. By increasing the resistance between audio outlet and Arduino, while sending a current at a specific frequency to the audio outlet the volume can be reduced [13].

2.4.1 Digital Potentiometer

Compared to a regular potentiometer, instead of relying on the position of the knob to determine the value of the resistance, the DP works in steps. Each step can be seen as multiple series of resistors connected to each other. With each additional step, the active wiper arm changes. This causes another resistor to be included in the series and in turn increases the resistance. The higher amount of steps the DP has, the smaller the value of the resistance from one step to another. A 10k Ω DP with 256 steps will allow an increase of around 39 Ω from each position to the next.

The DP has multiple channels which can be controlled. Each channel works as one resistor, and the active one can be changed using an MCU. Sending a high signal to the active one, and a low signal to the other ones changes whichever channel is active [7].
Chapter 3

Demonstrator

This chapter describes the construction as well as the components used and how a few of the more important problems were solved.

3.1 Arduino

The usage of an Arduino uno adds one big limiting factor - the amount of pins available. The Arduino uno has a total of 20 pins available, 6 analog pins and 14 digital pins [5]. The analog pins can easily be rewritten as digital pins and can therefore be used as both digital or analog.

Due to the usage of a lot of small components the availability of pins become very limited. The five sensors alone would require ten pins in total, as well as five for the buzzers and finally four for the DP. Additionally the zero and one digital input/output pin (I/O pins) are used for communicating with the computer, they are the receive and transfer pins. This causes pin zero and pin one to behave slightly unpredictably compared to the other pins. At times, the pins can be used as normal, but at other times they are completely unusable and do not properly produce a stable current [16]. As a result, the effective number of I/O pins available are reduced to 18. Using all components without extra libraries, would require a total of 19 pins. In order to solve this issue, without swapping to a larger MCU, the NewPing library is used.

3.2 Software

This section describe the thought process behind how the code works. For full source code see appendix B.
3.2. SOFTWARE

3.2.1 Code
The code for the Arduino is written with the C programming language, and the general overview of how the code works is shown in figure 3.1. Upon start up the Arduino uses the two lower sensors to measure the distance from the ground which is stored and used later on in the program. By using a simple loop each sensor measures the distance between the sensor and any object within the sensor’s reach, after each loop there is a slight delay before the next sensor starts measuring again. Depending on which sensor is active the corresponding buzzer will produce a tone. The length of each tone is also controlled using a delay.

Due to the usage of the NewPing library the Arduino’s second timer becomes occupied. The same timer is used by the standard tone library causing an interference. In order to solve this a similar library is used which works in the same way but on the Arduino’s first timer instead. This removes the interference and allows the NewPing and tone libraries to work simultaneously.

![Figure 3.1: Flowchart of software, created in draw.io](image)

3.2.2 The Newping Library
The NewPing library serves multiple functions. It allows easier handling of multiple sensors, as well as increasing the speed in which they can measure distance. The lowest time between each measurement is 29 milliseconds. Without the NewPing library, the pulses that were sent into the sensors to function slowed down the whole distance measuring process a lot, and as a result, one full cycle took too long. The NewPing library also allows usage of the same pin for the trigger and echo signals for the sensors, which effectively reduces the required pins for each sensor by half. This leaves more than enough pins for other components [6].
3.3 Sensors

All five of the sensors serve the function of measuring the distance in front of them. However they do not work under the same conditions. The three forward facing sensors work at max distance, while the two sensors that are facing slightly downwards have a limit on their reach. The reason for this is as the two sensors will always be pointing downwards, they will always be in contact with the ground as their range is roughly four meters. By limiting the downwards facing sensors and excluding them from the volume control the constant noise from the buzzers are reduced. The downwards facing sensors will only produce a noise if an object is within a set distance or if an object is too far away.

![Figure 3.2: Sensors as seen from above, with $\alpha = 20^\circ$. Each pair of cylinders represents one sensor, created in Solid Edge ST9](image)

3.4 Volume Control

In order to alert the user of their surroundings the DP work together with the sensors and Arduino to allow the buzzers to produce an appropriate sound at an appropriate volume.

3.4.1 Volume Control with Digital Potentiometer

The measured distance provided by the sensors is used in conjunction with the DP. Depending on the distance of the incoming object the DP is set at a certain value. The value of the DP increases as the distance to the object decreases. Meaning that if an object is very close, the resistance of the DP will be very low. Once the DP is set to a certain value, the current is sent to the correct Piezo buzzer and the buzzer will produce a sound at a certain volume depending on the value of the DP.

Since only three out of the five buzzers are dependant on volume control, a four channel DP is sufficient. The buzzers that are connected to the downwards facing sensors will not have any resistance connected to them, in order to keep them at maximum volume.
3.5 Instrument Case

As practicality is key, a simple and small instrument case is required. The requirement is to store all components inside without any external component other than the sensors as well as the buzzers. For the actual case, wood was chosen as the preferred material for a few reasons,

- Easy to modify afterwards

- Fairly cheap

- Lightweight

It should be noted that the first point is the most important one. This is because if the downward facing sensors were in need of a change in angle, the wood could be grinded down to reduce the angle, and make the sensors face downwards more. In case a larger angle is needed, the platform which the sensors are placed on could be removed and replaced with new ones.

Within the case, the Arduino is secured using small screws together with the circuit board that divides voltage and ground to the different components. The battery pack is also fastened in the case using small screws.

The DP is a SOIC-24 packaged integrated circuit, which needs its own circuit board to be usable. The circuit board chosen was a premade breakout board and the DP was carefully soldered onto it. It is very important that the soldering is well made, as risk of overloading the circuit is high.

3.5.1 Front of case

Each sensor is attached to the front of a case on a small elevated wooden platform which has been firmly glued on to the front of the case. Four of these have been angled differently in order to have a larger spread of where the sensors can reach. With a larger spread, the sensors have a smaller chance of having a blind spot.

Additionally another circuit board is attached to the front of the case which distributes voltage as well as ground to each and every sensor.
3.5.2 The Harness

In order for the user to be able to easily wear the case, as well as having it firmly secured to the body, a harness was created. The harness uses a very elastic material which can be worn and is fastened using Velcro. The Velcro is attached to the harness in multiple location, allowing for different sizes and lengths of the straps which allows the harness to be usable by people of different lengths and sizes while still securing the case onto the body.
3.5. INSTRUMENT CASE

(a) The Harness (b) Size adjustment with Velcro

Figure 3.4: The Harness

3.5.3 The Cap and Buzzers

In order to mimic a surround-sound system, multiple buzzers are used, one for each of the five sensors. The buzzers are attached to the cap in different locations, and communicates with the Arduino and DP using wires which lead to the case. Due to the position of the buzzers on the cap, distinguishing where the actual sound is coming from becomes easier. The buzzers are all angled towards the user to make it even easier to assess where the sound is coming from. The bottom buzzers are placed into a 3D printed case with a small hole for a safety pin. The safety pin is then easily attached to the users collar or the harness. The two bottom buzzers work on two different frequencies according to Table 3.1.

<table>
<thead>
<tr>
<th>Distance interval [cm]</th>
<th>frequency [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 - x</td>
<td>400</td>
</tr>
<tr>
<td>300 - 400</td>
<td>2000</td>
</tr>
</tbody>
</table>

Where x is 80% of the measured height above ground.

The difference between frequencies is used to alert the user from large changes of height as measured by the lower sensors. The lower frequencies are used for large obstacles in front of the user. The higher frequency is used when there is an increase of distance, i.e the user is walking down flights of stairs.
CHAPTER 3. DEMONSTRATOR

Figure 3.5: The Cap as seen from different angles

3.6 Battery

As each component is usable at five volts, no external battery pack is required and the voltage from the Arduino’s five volt pin is sufficient to run the components [9, 7, 10]. The Arduino itself has a higher operating voltage at nine to twelve volts [5]. This allows for a very small battery pack which is easily fitted into the case together with a nine volt battery to power the Arduino.
Chapter 4

Results and testing

In this chapter the testing methodology is presented as well as the results from the tests.

4.1 Testing

4.1.1 Practical tests

During testing ten people were asked the following three questions and asked to give the device a rating from one to ten, one being really difficult and ten being very easy.

1. How difficult was the case to wear? (in terms of weight and tightness of harness)
2. How difficult was the cap to wear?
3. How well could you follow where the sound is coming from?

They were also asked to walk around with the device and give their general thoughts on the device.

4.1.2 Reliability tests

In order to better understand the reliability of the HC-SR04 sensors a few tests were done. The methodology of the tests are presented in Table 4.1. The movement section indicates whether the measured sample is taken while the object is moving or if the object is standing still. As the device will almost always be used during movement, the behaviour of the sensors during movement as well as the results produced during movement are important. To measure the field of view of the sensors, the sensors were placed on a height above ground and a large object was slowly moved into the sensors FOV to see at what height the sensors would register the object.
Once the value registered started fluctuating between the distance from the object and distance from the wall placed behind the object, the distance between object and the mid point between transmitter and receiver was measured. By using this distance and the distance to the object, the angle of both the horizontal and vertical FOV were calculated. See appendix D for pictures of how the FOV testing was done. Once the FOV tests were done the error of margin as well as standard deviation was calculated using a t-test.

Multiple attempted tests for distances greater than 400 cm were done, however past the 400 cm point the results become very unreliable and proper values could not be extracted from the tests.

<table>
<thead>
<tr>
<th>Test number</th>
<th>Distance [cm]</th>
<th>Movement</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>Yes</td>
<td>Wall placed slightly behind moving object.</td>
</tr>
<tr>
<td>3</td>
<td>240</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>240</td>
<td>Yes</td>
<td>Wall placed further behind moving object at 362 cm.</td>
</tr>
<tr>
<td>5</td>
<td>399</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>No</td>
<td>Vertical FOV test, varying distances between 6 to 190 cm.</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>No</td>
<td>Horizontal FOV test, varying distances.</td>
</tr>
</tbody>
</table>

### 4.2 Results

#### 4.2.1 Practical test results

The results of the practical tests and questions are presented in Table 4.2.

<table>
<thead>
<tr>
<th>Question</th>
<th>Average score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.2</td>
</tr>
<tr>
<td>2</td>
<td>10.0</td>
</tr>
<tr>
<td>3</td>
<td>7.0</td>
</tr>
</tbody>
</table>

The general consensus of the subjects were that the device was indeed very easy to wear. Some did mention that they had difficulties putting on the harness but once the harness was on they could still move around freely without any restraints. Furthermore some did say that they could barely notice the device being on. Being able to follow the sound varied a lot from person to person, some said that they had a hard time understanding where the sound was coming from whilst others thought it was very simple to know which buzzer was beeping at what time.

When asked about the general experience when using the device one key thing was mentioned by every one. The device sensitivity was very low and that the ef-
4.2. RESULTS

effective distance in which they would hear that something was approaching was very low.

4.2.2 Measuring test results

Below are the results of tests one to five. Larger pictures of the tests can be found in appendix C.

(a) Test 1. distance interval 118.5-121.5 cm
(b) Test 2. distance interval 0-180 cm

![Figure 4.1: Measuring at 120 cm, created using Excel](image)

(a) Test 3. distance interval 239.5-244.5 cm
(b) Test 4. distance interval 0-400 cm

![Figure 4.2: Measuring at 240cm, created using Excel](image)
(a) test 5. distance interval 397.5-402.5 cm

Figure 4.3: Measuring at 399 cm, created using Excel
4.2. RESULTS

4.2.3 FOV testing

The results of the vertical FOV testing are presented in Table 4.3 and the horizontal FOV tests are presented in Table 4.4. Figure 4.4 shows the vertical angles of sensors, while figure 4.5 shows the horizontal angles of the sensors. The letters a to e represents each sides respective length.

Figure 4.4: Vertical angle of sensors as seen from the side, created in Krita
### Table 4.3: Test 6, vertical FOV

<table>
<thead>
<tr>
<th>Test number</th>
<th>length of a [cm]</th>
<th>length of b [cm]</th>
<th>$\alpha = \arctan(a/b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>64</td>
<td>$8^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>75</td>
<td>$7.7^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>181</td>
<td>$9.4^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>40</td>
<td>$8.5^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>190</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>110</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>159</td>
<td>$7.9^\circ$</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>127</td>
<td>$8.0^\circ$</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>110</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>171</td>
<td>$8.0^\circ$</td>
</tr>
</tbody>
</table>

Figure 4.5: Horizontal angles of sensors as seen from above, created in Krita

The results from Table 4.3 and Table 4.4 are concluded and presented in Table 4.5 with a confidence level of 95%:
4.2. RESULTS

Table 4.4: Test 7, horizontal FOV

<table>
<thead>
<tr>
<th>Test number</th>
<th>length of c [cm]</th>
<th>length of d [cm]</th>
<th>length of e [cm]</th>
<th>( \omega = \arctan(d/e) )</th>
<th>( \phi = \arctan(e/c) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51</td>
<td>12</td>
<td>5.5</td>
<td>13.2°</td>
<td>6.0°</td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>28.5</td>
<td>12.5</td>
<td>15.0°</td>
<td>7.5°</td>
</tr>
<tr>
<td>3</td>
<td>124</td>
<td>32</td>
<td>15</td>
<td>14.5°</td>
<td>6.9°</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>40</td>
<td>16</td>
<td>14.9°</td>
<td>6.1°</td>
</tr>
<tr>
<td>5</td>
<td>157</td>
<td>45</td>
<td>18.2</td>
<td>16.0°</td>
<td>6.6°</td>
</tr>
<tr>
<td>6</td>
<td>198</td>
<td>50.4</td>
<td>22.9</td>
<td>14.3°</td>
<td>6.6°</td>
</tr>
<tr>
<td>7</td>
<td>179</td>
<td>45.1</td>
<td>19.8</td>
<td>14.1°</td>
<td>6.3°</td>
</tr>
<tr>
<td>8</td>
<td>165</td>
<td>44.7</td>
<td>18.7</td>
<td>15.0°</td>
<td>6.4°</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>12</td>
<td>2.1</td>
<td>14.3°</td>
<td>6.0°</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>2.1</td>
<td>0.9</td>
<td>14.7°</td>
<td>6.4°</td>
</tr>
</tbody>
</table>

Table 4.5: FOV angles

<table>
<thead>
<tr>
<th>angle</th>
<th>mean</th>
<th>standard deviation</th>
<th>upper limit</th>
<th>lower limit</th>
<th>margin of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>8.1°</td>
<td>0.483°</td>
<td>9.4°</td>
<td>7.7°</td>
<td>0.345°</td>
</tr>
<tr>
<td>( \omega )</td>
<td>14.6°</td>
<td>0.740°</td>
<td>16.0°</td>
<td>13.2°</td>
<td>0.523°</td>
</tr>
<tr>
<td>( \phi )</td>
<td>6.5°</td>
<td>0.460°</td>
<td>6°</td>
<td>7.5°</td>
<td>0.323°</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion and Discussion

5.1 Discussion

As mentioned in the previous section, the test subjects all thought that the device was very easy to wear, but unfortunately the sensitivity of the sensors were low. This, however, is not necessarily due to the sensors FOV but partly due to the DP. The DP in use was a DP with a maximum resistance of 10k ohm. This caused the volume range of the buzzers to be very low. Originally a 100k ohm DP was supposed to be in use together with a different set of buzzers. The other buzzers would produce their highest volume between zero to roughly 1k ohm and the lowest volume would be produced at slightly below 100k ohm. The buzzers in use had their highest volume between zero and 500 ohm, while the lowest volume was reached already at around 4k ohm.

As seen from the results presented in section 4.2.2 the ultrasound sensors are not completely accurate. Even in the case where there is no movement the sensors have an error margin of maximum three centimeters including some noise. However, an error margin of three cm does not affect the actual sound emitted enough to be an issue. If a 100k DP is used together with the correct buzzers, three cm would be equivalent to approximately two steps, or 800 ohm. This value is not enough to give a noticeable difference in volume as heard by the user, meaning that they would perceive the volume of 800 ohm and 400 ohm to be the same. This is also applied if a 10k DP is used with the different set of buzzers. This means that an error margin of even three cm is completely acceptable.

From Table 4.5 it is concluded that the measuring angle of the HC-SR04 is not constant across each side. Some of the extreme deviated values could be explained by errors such as incorrect reading of measured values. However the biggest reason could be explained by the design of the HC-SR04. The sensors used in this project are not completely symmetrical, one side of the sensor is the receiver and the other side, is the transmitter. If an object is placed on the transmitter side of the sensor,
5.2. CONCLUSION

the ultrasound waves have to travel a further distance to the receiver. This means that the sensors are more effective to register objects which is on the receiver side of the sensors. The case is designed in such way that the side sensors, presented in Figure 3.2, have the same angle. However one side of the case has the FOV slightly shifted towards the middle, due to the different measuring angles presented in Table 4.5.

The horizontal FOV together with the vertical FOV and the angles of the sensors gives the device very few blind spots. While blind spots are very narrow, they do still exist. This results in sensors not being able to register objects with small surface areas that are placed in certain positions. However with a small enough surface area the object might be rendered insignificant.

The elastic material chosen for the harness was very good. Since it is elastic it tightens around the user and causes the case to be very secure with little movement. The Velcro allows for even further size adjustment to make the case even more secure.

5.2 Conclusion

With this thesis, we have looked further into how we can answer the research questions presented in section 1.2. We have succeeded in answering all three questions. The combination of buzzers connected to a DP together with an Arduino and ultrasonic sensor gives the user a overview of where things are around them. However the DP needs to have a big enough resistance to be able to give the buzzers multiple levels of varying volume.

The device have been designed in such ways to not disrupt the users everyday activities and be easy to put on and take off. Every person who tested the device gave a positive feedback regarding the harness and the case.

The reliability of the HC-SR04 ultrasound sensors have been tested and discussed in section 4.2. Even though the HC-SR04 is not accurate all the time, in this type of project the error margin does not have a large impact due to how the buzzers work. This means that using other distance measuring techniques to improve the accuracy is not a necessity, neither is improving the accuracy of the HC-SR04. The vertical and horizontal FOV angles are presented in Table 4.5. With those angles it is concluded that the FOV of the HC-SR04 is large enough to cover an area when in use with multiple sensors.
Chapter 6

Improvements

The most important improvement that needs to be done is the acquisition and proper mounting of the 100k DP. This would allow the sensors to work at a longer effective range which, in turn, will give the user a better sense of how far away an incoming object is.

In order to reduce some of the blind spots or adding different angles to the device a larger MCU or multiple MCUs are needed. The ATmega2560 would be a viable option. The dimensions for the case do already support the bigger size of the Arduino mega. With its 54 digital I/O pins and 16 analog input pins a multitude of extra sensors and buzzers can be added [14]. This would however also require a change in DP. Either using a DP with more channels such as the AD5206 [7] or have multiple DP work together using one as an serial peripheral interface (SPI) master and more as SPI slaves [15]. Equipping the device with more sensors would require an extension to the case since the device is very small, very compact and does not allow for much room on the front of the case. There is, however, room on the harness to add more sensors to receive an understanding of what is happening behind, or to the sides of the user. It is important to be wary of any changes done to the actual case, with a bigger case comes more difficulty wearing it, as well as adding to the weight of the whole device.

The measuring angle being uneven across the HC-SR04 led to the transmitter side of the sensors being less effective than the echo side. In order to solve this, the case should not be made symmetrical. By increasing the angle of the right most sensor, as seen from the front, the measuring angle of the trigger output is moved further to the right. Alternatively, one additional platform can be added to the case so that the total number of sensors is increased to six. This would require one additional buzzer as well and slight modifications to the Arduino code, there is, however, enough room on the Arduino uno to add up to two additional sensors and buzzers without changing the MCU.
From the feedback during the tests, some said that they had difficulties putting on the harness. This is due to the fact that the straps of the harness are too short. By increasing the length of the straps the area where the user puts their arms become bigger which makes the harness easier to use.
Bibliography


BIBLIOGRAPHY

    available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/
             HCSR04.pdf
    Visited on: (13/02/2018)

    available: https://www.Arduino.cc/documents/datasheets/
             PIEZO-PKM22EPFH4001-B0.pdf
    visited on: (13/02/2018)

             2085
    Visited on: (10/4/2018)

    Visited on: (15/4/2018)

             4315056
    Visited on: (25/4/2018)

    Arduino.cc/usa/Arduino-mega-2560-rev3

    cc/en/Reference/SPI

    com/obstacle-avoidance-robotic-vehicle/ Visited on: (29/4/2018)

[17] Sensors, "A Hybrid Bionic Image Sensor Achieving FOV Extension and
    1042/htm Visited on: (25/4/2018)

[18] Janschek Klaus, Mechatronic Systems Design: Metods, Models, Concepts,

    academy/field-of-view-fov-21050/ Visited on: (25/4/2018)
Appendix A

Arduino Code

/*
 * Program: Vision for the Blind
 * info: A program that together with sensors and buzzers use the measured
 * distance from the sensors to change the amount of steps in a
 * digital potentiometer to increase/decrease resistance and in turn
 * control the value of the buzzers.
 * date: 2018-05-09
 * names: Arsham Atighechi, Hussein Haidari
 * KTH
 * Bachelors degree in Mechatronics MF133X
 * CMAST 3
 */

#include <NewPing.h> //libraries used
#include <SPI.h>
#include <NewTone.h>

#define NUMBER_OF_SENSORS 5
#define DELAY_BETWEEN_BEEPS 250
#define NOTE_DURATIONS 250

int SonarPins[5] = {2,3,4,8,9};
int SoundLevel;
int i;
int HeightOfUser1 = 0;
int HeightOfUser2 = 0;
int AverageHeight = 0;
int DistanceArray[5];
int ChipSelect = 10;
int ChannelSelect[3] = {0,1,3}; //Digital potentiometers channels that are used
float VolumeMultiplier = 0.05; //255 steps, max distance is 400 from sensors. //255/400 when using 100k digital potentiometer, //0.05 (trying different values) with 10k

int BuzzerPin[5] = {A1,6,A2,5,7};

int MeasureHeight(int HeightOfUser);

NewPing sonar[NUMBER_OF_SENSORS] = { //The NewPing library requires //the sensors to be initialized
NewPing(3,3,400), //upper sensor to the left when viewed from the front
NewPing(4,4,400), //upper middle
NewPing(8,8,400), //upper right
NewPing(9,9,400), //down left
NewPing(2,2,400)}; //down right

void setup() {
    //put your setup code here, to run once:
    Serial.begin(9600);
    PiezoSetup(BuzzerPin);
    pinMode(ChipSelect,OUTPUT);
    AverageHeight = MeasureHeight();

    SPI.begin(); //required for communication with digital potentiometer //through SPI interface
}

void loop() {
    //put your main code here, to run repeatedly:
    MeasureDistance();
    PlayTone();

}

int MeasureHeight() { //measures the height above ground of the //lower sensors
    for (i=0;i<10;i++){
        HeightOfUser1 = HeightOfUser1+sonar[3].ping_cm(); //sample 10 //measurements of the 2 lower sensors
}
APPENDIX A. ARDUINO CODE

```
HeightOfUser2 = HeightOfUser2 + sonar[4].ping_cm();
}
HeightOfUser1 = HeightOfUser1 / 10; // count the
// average of the 10 measurements
HeightOfUser2 = HeightOfUser2 / 10;
AverageHeight = (HeightOfUser1 + HeightOfUser2) / 2;
// Serial.print(AverageHeight);
return AverageHeight;
}

void MeasureDistance()
{
    for (i = 0; i < NUMBER_OF_SENSORS; i++) {
        // Loop through depending
        // on how many sensors used (5)
        DistanceArray[i] = sonar[i].ping_cm(); // store each value in array
delay(40);
    }
}

void PiezoSetup(int PinArray[5])
{
    // set the buzzer pins as output
    for (i = 0; i < 5; i++) {
        pinMode(PinArray[i], OUTPUT);
    }
}

// void PlayTone(int pin, int counter)
void PlayTone()
{
    // plays the sound
    for (i = 0; i < 5; i++) {
        // loops through each buzzer
        SoundLevel = DistanceArray[i] * VolumeMultiplier; // calculate correct
        // value for digital potentiometer
        if (i < 3) // if its the upper buzzers
        {
            DigiPotControl(ChannelSelect[i], SoundLevel); // sets the
            // resistance to correct channel
            NewTone(BuzzerPin[i], 400); // play tone on that channel
delay(NOTE_DURATION);
            noNewTone(BuzzerPin[i]); // turn tone off
delay(DELAY_BETWEEN_BEEPS); // delay before next tone starts
        }
        else // lower buzzers
        {
            // lower buzzers
        }
    }
```


if((AverageHeight*0.8)>DistanceArray[i]) //if something within 90%
//of the measured height
{
    NewTone(BuzzerPin[i],400); //play the tone, low frequency
    delay(NOTE_DURATION);
    noNewTone(BuzzerPin[i]);
    delay(DELAY_BETWEEN_BEEPS);
}
else if(DistanceArray[i]>300 and DistanceArray[i]<400) //if something
//is far away from the lower sensors
{
    NewTone(BuzzerPin[i],2000); //play high frequency tone
    delay(NOTE_DURATION);
    noNewTone(BuzzerPin[i]);
    delay(DELAY_BETWEEN_BEEPS);
}
else //if neither condition is fulfilled do nothing
{
}

delay(DELAY_BETWEEN_BEEPS);

void DigiPotControl(int channel, int steps) { //to control
    //the right channel and value
    digitalWrite(ChipSelect, LOW);
    //turn the ChipSelect pin low to select the chip
    SPI.transfer(channel);
    //use SPI to transfer the amount of steps (resistance)
    //to correct channel
    SPI.transfer(steps);
    //turn ChipSelect high to the select chip
    digitalWrite(ChipSelect, HIGH);
Appendix B

Results from tests

Figure B.1: With movement, distance at 120cm, created using Excel
Figure B.2: Without movement, distance at 120cm, created using Excel
Figure B.3: With movement, distance at 240cm, created using Excel
Figure B.4: Without movement, distance at 240 cm, created using Excel
APPENDIX B. RESULTS FROM TESTS

Figure B.5: Without movement, distance at 399cm, created using Excel
Appendix C

Progress pictures

Photos of completed as well as uncompleted project.

Figure C.1: Completed project
APPENDIX C. PROGRESS PICTURES

Figure C.2: Progress picture 1
Figure C.3: Progress picture 2
Figure C.4: Progress picture 3
Figure C.5: Progress picture 4
Figure C.6: Progress picture 5
Figure C.7: Progress picture 6
Figure C.8: Progress picture 7
Figure C.10: Progress picture 9
Figure C.11: Progress picture 10
Figure C.12: Progress picture 11
Figure C.14: Progress picture 13
Figure C.15: Progress picture 14
Figure C.16: Progress picture 15
Appendix D

FOV testing

Table D.1: Vertical FOV

<table>
<thead>
<tr>
<th>Test number</th>
<th>length of $a$ [cm]</th>
<th>length of $b$ [cm]</th>
<th>$\alpha = \arctan(a/b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>64</td>
<td>$8^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>75</td>
<td>$7.7^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>181</td>
<td>$9.4^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>40</td>
<td>$8.5^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>190</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>110</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>159</td>
<td>$7.9^\circ$</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>127</td>
<td>$8.0^\circ$</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>110</td>
<td>$7.8^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>171</td>
<td>$8.0^\circ$</td>
</tr>
</tbody>
</table>

Table D.2: Horizontal FOV

<table>
<thead>
<tr>
<th>Test number</th>
<th>length of $c$ [cm]</th>
<th>length of $d$ [cm]</th>
<th>length of $e$ [cm]</th>
<th>$\omega = \arctan(d/e)$</th>
<th>$\phi = \arctan(c/e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51</td>
<td>12</td>
<td>5.5</td>
<td>$13.2^\circ$</td>
<td>$6.0^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>28.5</td>
<td>12.5</td>
<td>$15.0^\circ$</td>
<td>$7.5^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>124</td>
<td>32</td>
<td>15</td>
<td>$14.5^\circ$</td>
<td>$6.9^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>40</td>
<td>16</td>
<td>$14.9^\circ$</td>
<td>$6.1^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>157</td>
<td>45</td>
<td>18.2</td>
<td>$16.0^\circ$</td>
<td>$6.6^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>198</td>
<td>50.4</td>
<td>22.9</td>
<td>$14.3^\circ$</td>
<td>$6.6^\circ$</td>
</tr>
<tr>
<td>7</td>
<td>179</td>
<td>45.1</td>
<td>19.8</td>
<td>$14.1^\circ$</td>
<td>$6.3^\circ$</td>
</tr>
<tr>
<td>8</td>
<td>165</td>
<td>44.7</td>
<td>18.7</td>
<td>$15.0^\circ$</td>
<td>$6.4^\circ$</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>12</td>
<td>2.1</td>
<td>$14.3^\circ$</td>
<td>$6.0^\circ$</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>2.1</td>
<td>0.9</td>
<td>$14.7^\circ$</td>
<td>$6.4^\circ$</td>
</tr>
</tbody>
</table>

The ruler is used to measure the height from the sensor to the top of the object once the sensor starts registering the object. The distance from the sensor to the
object is read from the serial communication with the computer and then confirmed with the measuring tape.

Figure D.1: FOV 1, vertical
The large object is slowly moved into the FOV of the sensors and once it starts to register the distance is read through the serial communication with the computer and then confirmed using the measuring tape. The distance from the object and the measuring tape is then measured to be able to calculate the angle.
Figure D.3: FOV 3, horizontal
Figure D.4: FOV 4, horizontal