Bachelor thesis

Computer Science + 300hp

SafeWalk

A tool for the visually impaired

Halmstad 2022 May
Mohammad Omar Abdullah, Gabriel Färm

HÖGSKOLAN
I HALMSTAD

Abstract

This thesis aims to implement a system called SafeWalk to help visually impaired people to sense obstacles in their path and provide them with more independence and more effortless life routines without any external help. SafeWalk is a low power consuming system designed to replace the daily used white cane. It consists of sensors, a buzzer, a vibrating motor, and a software application. The system will be installed on the user's shoe. It detects the objects in front of the user using an Ultrasonic sensor that measures the distance to the targeted object by transmitting sound waves. This will produce vibration in the shoe and causes the buzzer to start playing when the object gets closer. Similarly, using the provided application, signals will be received in the user's smartphone, which will begin to vibrate when the object is detected. The system works just as planned. The final prototype senses objects in the walking path very well. The connection to the app was strong and never disconnected while in use. The phone vibrates stronger if an object on the ground gets closer; if there is an object in the air, the buzzer and vibrator signal the user.

Table Of Contents

List Of Figures And Tables

1. Introduction

Today most visually impaired people use a white cane to help them walk and know where they are going, but it is not always perfect. According to Rowena Portch, a person who uses a white cane has many problems. She mentions that one of the most annoying problems is in crowded areas. People do not look at what they are doing, stand in the way, etc. (1). This often leads to people stepping and "kicking" the cane so that the person drops it. According to a study of 300 visually impaired people, at least once per month, 7% of these people experience falling (2).

SafeWalk is a product meant to help blind/partly blind people to detect objects without using the white cane that is used today. We want to exchange the white cane with our product for several reasons. Some of the most important ones are so that the person can use both hands if they are carrying something. If they were to fall to the ground, they would have two hands to catch themselves, and due to the technological improvement so far, we believe that the more projects around this area, will bring more innovations and will help visually impaired people. We want to create a product that will see when the ground is uneven to ensure that people with vision impairment do not trip while they walk. The product will be attached to the shoe, and then there will be an ultrasonic sensor facing forward, and that way, it will see where the objects are. The user will then receive a vibrating signal to know how far away the object is.

The most significant reason for developing this product is that we want blind people to have access to excellent and helpful technology. We believe that this will change the way for blind people to walk and feel safer. According to a study with blind people around 34 years old, one good example of a problem is the sidewalk. One of the most challenging things is walking on sidewalks because they can be unpredictable (3). This is where our product will shine. The sensor scans the surface and sends a signal to the user to know if there is a root from a tree pushing up the concrete or perhaps an electric scooter on the ground.

We have done a lot of background research to understand what kind of problems blind people face every day while walking. We wanted to get information from the people themself and not only studies. We found a site called *Quora*, a forum where people can post questions, and other people can answer them. On this specific question, "What are the problems that the visually impaired face with the white cane?" we found many problems that we believe our product can solve. Penina Winisdatter shared a few issues and personal experiences. Penina mentions that her cane has been stolen more than once because she forgot it when she walked away with a friend who guided her. As mentioned before, Rowena Portch shared information on this post, and she also said that small dogs often see the white cane as a threat and start biting it. She also mentions that she almost daily pushes the cane into dog scat due to lazy dog owners. Kim Paulk shared information that we found very disturbing, "There are many challenges with the use of a white cane, from the tip to the fact that they do not detect anything higher than our waist/chest.".

SafeWalk is a device that is attached to the front of the shoe. At the front of the device, there is an ultrasonic sensor that sends a sound wave at 40kHz, which is above the frequency of human hearing. If there is an object in front of the sensor, the wave will bounce on the object back to the sensor, and that is how we know that an object is in the walking path. This sensor will send a signal to an app on your smartphone, which will then vibrate depending on how far away the object is. There is also a sensor facing up in the air with an angle, which is used to detect obstacles hanging over the walking path. When the sensor receives the wave, it will send a signal to the buzzer as well to a vibration motor.

SafeWalk is supposed to be affordable compared to the competitors. However, not only is it supposed to be cheaper, but it is also a better product since it has two sensors to sense the objects, both on the ground and in the air. The two sensors have their own place where it signals the user. The sensor facing straight forward signals the user through the app, and the sensor that is aimed up in the air signals the user through a buzzer and vibrator that is placed in the device. The purpose of the two sensors is to generate two different types of signals. If the targeted object is in the air, the user will be warned with acoustic signals as well as vibration feedback. Similarly, for the sensor facing straight forward, if it detects an object in its path, it will generate vibration in the smartphone. The sensors are connected to different pins on the Arduino to differentiate where the signals are sent. This makes it possible to choose which pin should send the signal to the app or the buzzer and vibrator. Using both the app and the device makes it easy for the user to distinguish if the object is on the ground or in the air. Having a software application increases our ability to add more functional tools to the system, such as voice command, 3d-Mapping, a navigation system, and an object detector and recognizer.

Chapter 1 (Introduction) introduces the problem statement and describes the importance of this project. *Chapter 2 (Background)* consists of the background on the white cane and Bluetooth module. *Chapter 3 (Theory)* describes the scientific methods and processes. The reader will learn more about the ultrasonic sensor's technology and how Bluetooth works. *Chapter 4 (Materials and methods)* includes the methods and components we used to complete this project. It also includes a subsection called "HW & Block diagram" which explains how the components were connected, wiring diagrams, and the flow of the program. This section illustrates the workflow of the program and the interaction between the hardware and software, and how the software application was implemented. *Chapter 5 (Results & Discussion)* concludes the result that has been made from the whole project. In this chapter, we present how the system was tested in general. But also how each device and component needed to be tested. Moreover, the problems found in the system after testing and the solution that could be implemented to solve the problems will be analyzed in this chapter. *Chapter 6 (Conclusion)* summarizes the entire project and thesis. This chapter analyzed the achievements of this work.

2. Background

This chapter of the thesis includes the history of the white cane, other related projects, research and studies in this field, and knowledge and information about how Bluetooth-based devices work.

2.1 Similar tools (White cane)

Throughout the history, canes, sticks, and staffs have been used as traveling aid for the visually impaired and blind. Records from biblical times show that visually impaired and blind people used a Shepherd's staff to help them during solitary travel. In 1921, James Briggs was caught in a terrible accident, causing his sight to disappear. James felt threatened by all the new motor vehicles around his city while walking, so he painted the cane white to make it easier for them to see him. Although it was not until ten years later, in 1931, that the white cane established a presence in society. At this time, the white can was a symbol to let other people know that a person had visual disabilities. In North America, the white cane was introduced by a Lion's Clubs member who watched a blind man attempting to cross a dark road. This man could barely see the cane nor the man holding the cane. He, therefore, helped the man paint the cane white, which later turned out to be the new standard. When the blind veterans from WWII returned to America, the way of using the white cane was improved. Doctor Richard Hoover developed the "Hoover" or "long cane" method of cane traveling. This helped the cane return to its original role and be a tool for mobility (4).

A massive struggle for blind people is handling the cane when they are not using it. For example, since it is so long, getting it into a car can be a difficult task if the car is a smaller type, same situation in a bus, airplane, train, etc. There will always be a problem with storing a long cane.

Smart shoe technology is not new, and many research papers and products have been implemented. There are also many projects/products developed to solve everyday routine work for blind and impaired people. Examples of these technologies are 3D mapping and navigated smart shoes (5), Real-Time Object detection using AI and Neural networks (6), wearable obstacle detection systems integrated into textile structures (7), IoT-based Smart Shoes for the blind (8), and much more. According to WHO[1] statistics (9), about 37 million individuals are affected by blindness. Consequently, many technologies are developed to fulfill the needs of visually impaired people.

---

[1] World Health Organization

There are a few projects and products like ours. Some examples are; InnoMake, Mini Guide, and UltraCane (10). We see much potential in these projects, and we have learned a lot from them, although we see a significant problem, the price. The price for these products ranges from 300$ for the most basic version to 3000$ for more advanced, and according to WHO, about 90% of the visually impaired live in a low-income setting. Therefore, we want to make a cheaper and available product for a much wider audience, not only the top of the economic chain.

An additional important project in the same category is, SoincGuide. SonicGuide is a device used as a head mount and has a camera that takes pictures and analyses the pictures based on an algorithm to find any unusual objects. The device signals the user by sending alarms to a connected earbud. This is a great product, but it has its flaws. A major flaw that we can see is neck pain after wearing it for a long time. Since the device uses a camera to take pictures all the time, it is very battery-consuming and, therefore, must be charged a lot. This project is good for us because we can learn the disadvantage of having a camera and what type of signal we can give use on our project. Since we will be implementing an app, the user will be able to wear earbuds (11).

As mentioned before, InnoMake is one of the most significant products in this field and has been done by the Austrian company *Tec-Innovation*. The collaboration of Tec-Innovation and the Austria's Graz University of Technology leads to the development of an intelligent shoe with the brand name "InnoMake" that tells you where to go and how to avoid obstacles in your path. The smart shoe is built with distance sensors, a vibration unit, an ultra-bright LED, a processing unit, etc. The InnoMake shoe uses ultrasound sensors to alert the impaired whenever they face any obstacle. According to Ashwini Sakharkar (2020), the shoe can detect potential obstacles up to four meters in front of the user (12).

As mentioned above, SafeWalk is supposed to be affordable compared to the competitors. However, not only is it supposed to be cheaper, but it is also a better product since it has two sensors that sense objects, both on the ground and in the air. The two sensors have their own place where it signals the user. The sensor facing straight forward signals the user through the app, and the sensor that is aimed up in the air signals the user through a buzzer and vibrator that is placed in the device. The sensors are connected to pins on the Arduino, making it possible to determine where the signal should be sent. Using both the app and shoe makes it easy for the user to distinguish if the object is on the ground or in the air. This is a significant solution to the problem mentioned above by Kim Paulk, and the white cane does not detect objects higher than the waist. SafeWalk will detect up to head level.

2.2 Bluetooth-connection and App

Bluetooth is a technology used for exchanging data between two devices; they can either be mobile or fixed. This technology is now a standard for wireless connection to substitute the hardwired connection. Since the Bluetooth connection is very energy effective, this is very common in portable devices such as headphones, speakers, phones, etc.

To this date, there are hundreds if not thousands of Arduino programs that have connected an app via Bluetooth. One problem discovered during the research process was that most projects link the program to an already existing app on the google play store. In our project, we wanted to program our own app, so after much searching, we finally found a project that had an app that was used to turn on an LED. Although this is not really what we needed our app to do, we found much inspiration on how we would set up the Bluetooth connection and how the Arduino and our app would send and receive signals from each other.

We found the project "*Creating Bluetooth Android App to Control Arduino Board."* This project started from the beginning by describing to the reader how the app will work when it is completed. After this, there was a step-by-step list with all the necessary steps we would have to do to create a working app with a stable Bluetooth connection. Then, code snips were provided that showed essential parts of the project and described what each piece did. The main reason we got much inspiration from this article was that it described very well for the reader how to program a Bluetooth connection between Arduino and Android (13).

A very educational project is "Arduino Bluetooth controlling system for Home Automation using Arduino & Bluetooth." As described in the title, this project uses an Arduino and a Bluetooth module to connect the Arduino with a smartphone. The good thing about this tutorial was that it included more details on how to send information from Arduino to Android. It also included a circuit diagram that was very helpful for when we wired our Bluetooth module. In addition to just programming, this article also included information about Bluetooth and how it works (14).

We had a hard time finding a project where the data is sent from the Arduino to the app; most projects that we found send a signal from the app to the Arduino. Although after much research, we found a project where data is sent from the Arduino to the app. This project is called "How to Create Android App for Arduino Sensor Monitoring over Bluetooth." (15).

These projects above were crucial for us to be able to complete this app. Although we could not make any improvements compared to those in the articles, we still managed to program our own app instead of using an already available for download.

3. Theory

The following section will explain the technical parts of this project and how they are used. It will mainly be focused on the Bluetooth module and how ultrasonic sensors work.

3.1 Ultrasonic Sensor

The ultrasonic sensor is one of the most important components in this project. We had a variety of sensors to choose from, but we decided that this one would be the best since it can detect objects in a wider area than, for example, a photoelectric sensor. Another important feature of the ultrasonic sensor is that it does not matter what color the object is, nor if it is dark outside. Since there are two parts of the sensor, one sends the 40KHz soundwave and one that receives it. If there is an object, the emitted soundwave will reflect on the object and bounce back to the receiver. The sensor measures the distance to an object by calculating the time from when the waves were sent until they were received. Then, we use *Equation 1* to calculate the actual distance in centimeters.

$$Distance = \frac{\frac{Duration}{2}}{29.1} \qquad (1)$$

We use duration divided by two since the distance to the object is only half of the time compared to the recorded time. Then we divide this by 29.1, and the reason for this number is because when we divide one by the speed of sound, we get 29.1 microseconds, which is how long it takes sound to travel 1 cm.

Piezoelectric ceramics are used inside the ultrasonic sensor, specifically inside the transmitter and receiver. Looking at *Figure 1*, we can see that the ceramics are at the sensor's front. Piezoelectric ceramics is a material that converts a mechanical effect into an electrical signal and vice versa. This is why it is used in the ultrasonic sensor because when the soundwave hits the ceramic, it will then transform the "wave" into a signal sent to the Arduino (16).

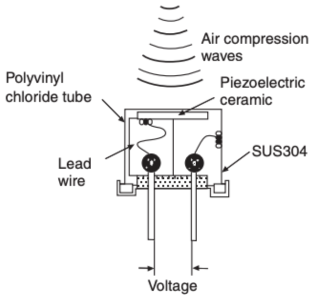


Figure 1. Ultrasonic receiver

3.2 Bluetooth

Every Bluetooth module has its own Bluetooth circuit that communicates wirelessly with other Bluetooth devices. By sending and receiving data through radio waves in the ISM bands at 2.404 GHz to 2.48 GHz, the devices build their own personal network. As soon as two Bluetooth circuits are close enough, the exchange will begin. This exchange is used to create a connection between the two modules.

Bluetooth networks, also called *piconets,* adapt the master/slave model to control where and when devices can send data. With this master/slave model, a single device can be connected to a total of seven different (slave) devices. However, a slave device in the piconet can only be connected to a single master. *Figure 2* below represents how the master can be connected to many devices but a slave only to one.

The master establishes communication through the piconet. It can send and request data from any of its slaves at any given time. Although, the slaves can only be allowed to receive and transmit data to the master, not other slaves in the piconet.

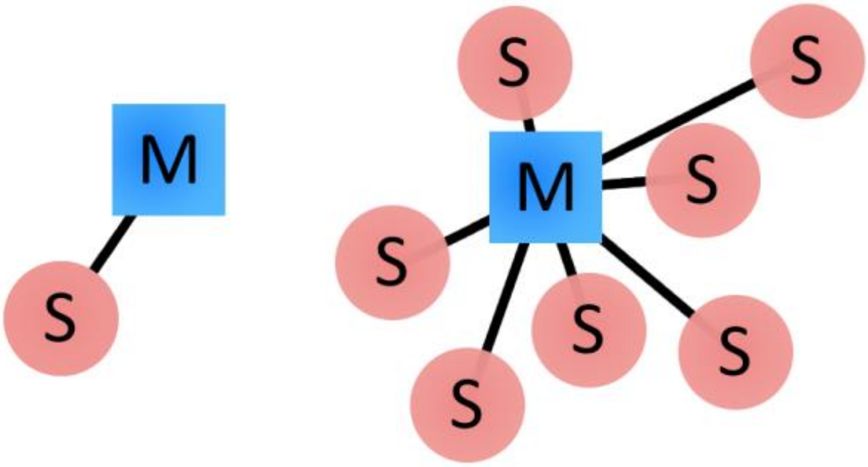


Figure 2. Bluetooth Slave & Master

To create a Bluetooth connection, a multi-step process must be completed, and this process includes three progressive states. The steps are *inquiry*, *paging*, and *connection*.

1. **Inquiry -** if it is the first time two devices are discovering each other, one of them is required to run an inquiry to be able to discover the other device. One of the devices sends the inquiry request, and any other device that is "listening" for a request will respond with its address, name, and other information.

2. **Paging -** This is the process of establishing the connection between two devices. Before the connection can be initiated, both devices need to know each other's addresses.

3. **Connection -** When a device has completed the two steps above, it enters the connection state. When a device is connected, it can either be put into low power sleeping mode or participate actively.

    - **Active mode -** is the standard connected mode; the device is actively receiving or transmitting data.
    - **Sniff mode -** A power-saving mode, the device is less active. It will sleep and only listen for transmissions at a given interval.
    - **Hold mode -** Hold mode is a temporary, power-saving mode. This is a loop where the device is put to sleep for a given time and then returns to active mode.
    - **Park mode -** This is where the device is in the deepest sleep. The master commands a slave to "park" and can only again be activated by the master.

When two devices have been connected one time, they will always be able to bond as soon as they come in a range of each other. In everyday life, you do not have to do the whole pairing process; it just connects. The same is for wireless headphones. They connect to each other without pairing again since they share a secret key that was created during the initial pairing process. There are two ways to transfer the data, synchronized and unsynchronized. Synchronized is used for audio transmission, while unsynchronized is used for other data transmission.

4. Materials and Methods

This chapter describes the workflow of this project, what components are involved in building the prototype, and the wiring diagram. It includes a Flowchart that describes the workflow of the system to illustrate the process from detecting objects to signaling the user. Furthermore, this chapter explains the implementation of the app. This subsection includes the App interface, the app's workflow, and the interaction between the Arduino microcontroller and the Android app to establish a stable connection between the hardware and the software.

4.1 Methodology

SafeWalk consists of two main parts. A hardware prototype with the different components interacting together and an application to provide signals to the user with the help of vibrations in the unit. The vibration will be generated from the vibrator motor or the smartphone.

Figure 3 below shows the process from studying to coding and testing. The literature, studies, and research are included in the background session above. To combine knowledge effectively, we started by reading through relevant and similar research and projects in this field to help us build a solid and strong background to implement the prototype. As important as that was ensuring that all group members understood what to do. Thus, continuous discussion of the material and the notes taken has been a main part.

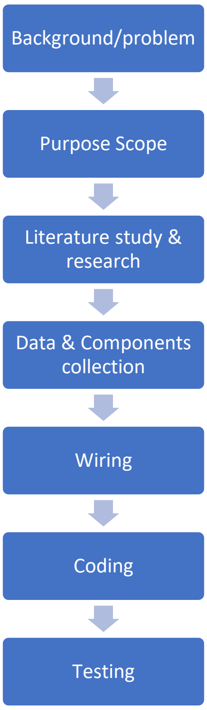


Figure 3. The process model of the study & implementation

4.2 Components

- **Arduino Uno**: is a microcontroller board consisting of 14 digital I/O pins, of which six can be used as PWM[2], see *Figure 4* below. Arduino Uno contains all the specifications which satisfy our requirements. Arduino is an open-source hardware and software, making it easy to write code and upload it to the board. Nevertheless, it runs on Windows, Mac OS, and Linux. Arduino gives users the ability to create a derivative of the board or a new product based on the Arduino's design. Therefore, we choose to write the source code in Arduino IDE and use the Arduino Uno as our main microcontroller.

Figure 4. Arduino Uno

- **Buzzer**: is a device that generates audio signals. There are many types of buzzers, just like magnetic, electromagnetic, mechanical, etc. But we chose a piezoelectric buzzer which is seen below in *Figure* 5. The reason why we chose this buzzer was because of its low power consumption and simple application. A piezoelectric buzzer consists of a piezoelectric element, which is activated by the Arduino Uno microcontroller to produce a beeping signal. The buzzer will be installed on the shoe and connected to the Ultrasonic sensor.

Figure 5. Piezoelectric buzzer

- **Ultrasonic sensor**: is a device to measure the obstacle distance. Ultrasonic sensors contain two units, a transmitter, and a receiver. It sends inaudible ultrasonic sound waves in the medium to detect the obstacles. The waves flow through the air, and if they

---

[2] Pulse-width modulation

come into contact with an obstacle, they reflect according to their angle of incidence (17). See *Figure 6* below.

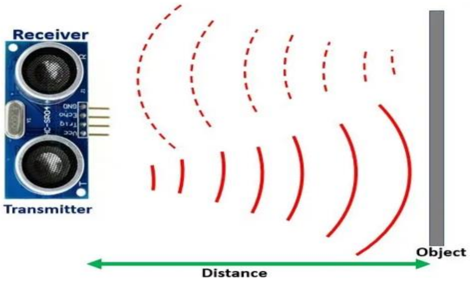


Figure 6. Ultrasonic detection mechanism

With the help of these waves, we can determine the distance between the object and the sensor. Typical ultrasonic sensors have four pins.

- Trigger pin: is an input pin used to start the distance measurement.
- Echo pin: represents the pulse output pin. It sends a pulse as an output. By measuring the width of the pulse of the echo signal, we can calculate how long it takes for the sound wave to travel between the sensor and the confronting object. Thus, depending on the time it takes, we can determine the distance between the sensor and the object.
- VCC and GND are used to power the sensor.

Mathematical calculation of distance using Ultrasonic sensor:

- Sound waves sent by the sensor travel at a speed of 340 m/s, corresponding to 29.1 µs/cm.
- Distance = (speed of sound * time)/2. We divide by two because the sound needs to travel back and forth.

- **Vibrating motor (mini)**: As the name indicates, a vibrating motor is a device used to generate vibration. It is widely used in many products such as cell phones and pagers. It comes in different sizes. The purpose of using the vibrator motor in our project is to alert the user if they have hearing problems or if they are in a noisy environment were the buzzer cannot be adequately heard. Like the buzzer, the vibrator motor will be installed on the shoe. Similarly, the buzzer and the vibrator motor will be installed on the shoe. *Figure 7* below shows a coin vibrator.

Figure 7. Coin vibrator

- **Bluetooth chip:** The Bluetooth module is used to create a wireless connection between the Arduino UNO and a smartphone. In this project, we use the HC-06 module, which is shown in *Figure 8*. This module draws 40mA while establishing a connection and only 8mA while running. This makes the module a very good option for our project since we will be running the device on a battery. The HC-06 module has four pins, two pins are used to establish the connection, and two pins are used for power (18).

Figure 8. HC-06 module

The different pins of a Bluetooth chip are:

- VCC pin is used for power.
- GND is the ground pin.
- TX is the transmission pin that transmits data.
- RX pin is used to receive data

- **Battery pack:** The battery we will be using is a six-cell 1200mA Nickle-Metall Hydride (NiMh) battery. With an output of 7.4V, this battery is more than enough to power the device. *Figure 9* below is a picture of the battery.

Figure 9. Battery pack

Battery consumption is a big concern in the SafeWalk project. Therefore, it is essential that we calculate how much battery it is supposed to consume. However, we know that the consumption of:

- Arduino UNO microcontroller in the range of 45–50 mA
- The buzzer is less than 10 mA
- Vibrating motor less than 80mA
- The Bluetooth module is about 8 mA

Considering the previous data, calculating the worst-case scenario, where all the components are ON all the time, a total of 148 mA will be drawn each hour. Using a battery with 1200 mAH, the device will be able to go 8.1 hours without having to be recharged. Although, since there will not be obstacles in the air at most times, a calculation where the vibrating motor is active 5% of the time is done. This gives the result of 73 mA each hour. Using the same battery, the device can go 16.4 hours without having to be recharged.

These are the main components needed to develop the prototype, and after calculating the price of all the components, the total price for the components, excluding a box, is 55 dollars. However, since we are implementing SafeWalk as our own project, we are responsible for providing/buying the components ourselves or collaborating with our university, Halmstad University. Fortunately, they had most of the components we needed to start our project.

4.3 Wiring HW & Block Diagram

The block diagram in *Figure 10* shows the working principle of the SafeWalk project. The device will be installed in the user's shoe. The process starts when the Ultrasonic sensor is activated. It will regularly send sound waves to detect obstacles in its path and in a range specified in the program code. The sensor will interact with the microcontroller to activate the buzzer and vibrator motor when the receiver unit in the Ultrasonic sensor finds the obstacle and measures the distance to it. Then, the buzzer and vibrator are activated, and the user will be notified (19). The left part of *Figure 10* below shows how the components are connected. This is the primitive form of the prototype, as it will later be packed into a small box to be placed in the user's shoe.

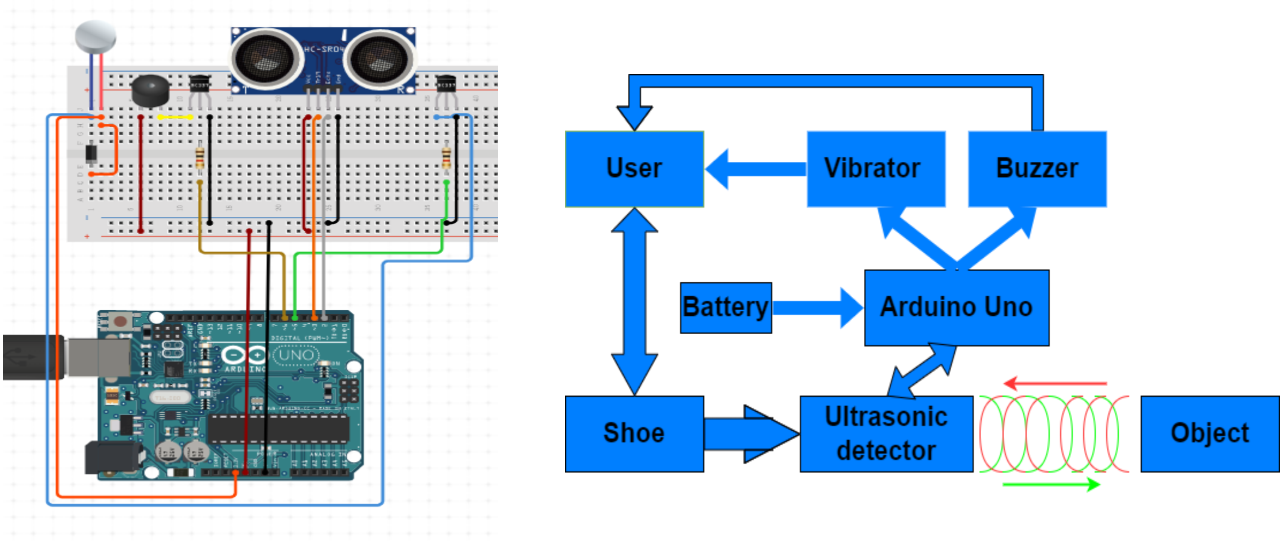


Figure 10. Wiring & Block diagram

To create communication between the devices, a Bluetooth module is implemented. The Bluetooth module has four pins, as mentioned in the components list. The VCC pin is connected to the 5V on the Arduino, and the GND pin is connected to the GND. Then there are the two data pins, the user chooses these pins, and we wired the TX pin to the RX pin on Arduino. The RX pin was wired to the TX pin. The wiring for the Bluetooth module is shown below in *Figure 11*.

The TR and RX pins on the Arduino board are used for serial communication between the Arduino board and a computer or similar devices. The Arduino UNO communicates on digital pins 0 (RX) and 1 (TX), except for the USB port. The problem with these two pins and using them for our Bluetooth module is that if we have the RX and TX pins connected to anything while uploading a program, it will fail to be uploaded. To fix this, we cut the power to the Bluetooth module while uploading the program, so the pins were "free."

For Arduino to be able to send data to the Android app, the *Serial* port was used. The serial port communicates through the RX and TX pins, and by using the *serial()* library, we could write and read from the serial monitor on the Arduino.

In the Android Studio environment, we used a message object to read the data from the Arduino. This message is the same value that the ultrasonic sensor displays on the Arduino serial monitor. We make the phone vibrate with a different amount of "force" to signal the user, depending on how far away the object is. The phone will vibrate at the lowest preset frequency if the object is far away (200 cm). Then, the closer the sensor (shoe) comes to the object, the stronger the phone will vibrate.

We had many problems implementing the app and how we would signal the user. The original plan was to vibrate at different speeds depending on the object's distance, just like a car parking sensor. Instead of setting up a bunch of intervals, we inverted the ultrasonic sensor's value sent to the serial monitor and then used this in the vibration function.

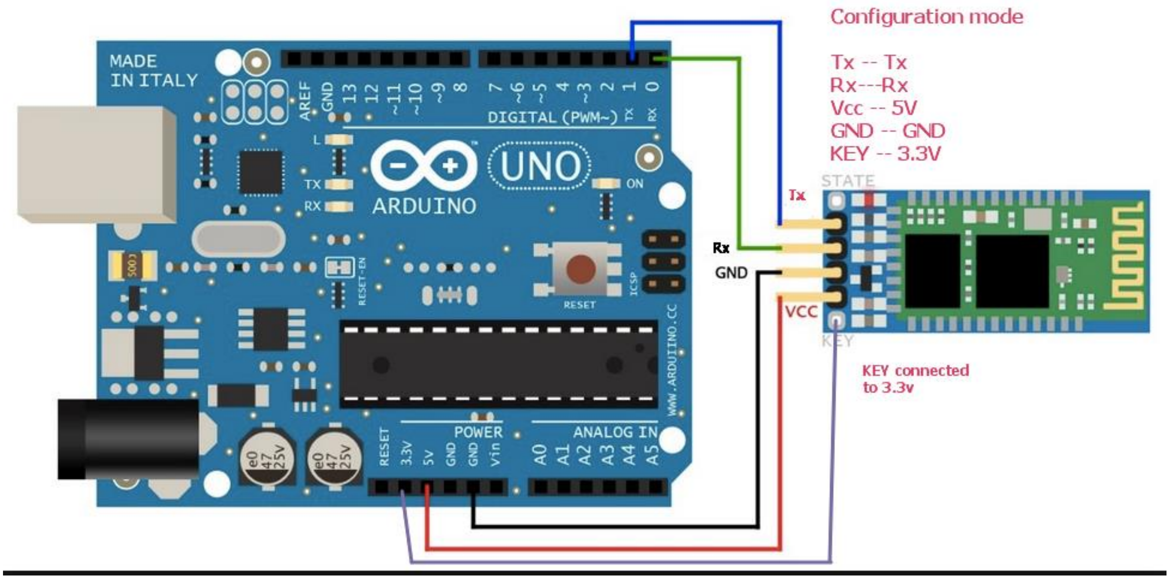


Figure 11. Bluetooth module wiring

4.4 Flowchart Of the Program

Figure 12 below explains the main workflow of the program. The process starts with the Arduino sending a LOW to HIGH signal on the Trig pin of the Ultrasonic sensor. Afterward, the Ultrasonic sensor transmitter unit sends sound waves to detect if there is any object in its path. When the transmitted waves collide with any obstacle, they reflect back to the Ultrasonic sensor, where they are received in the receiver unit.

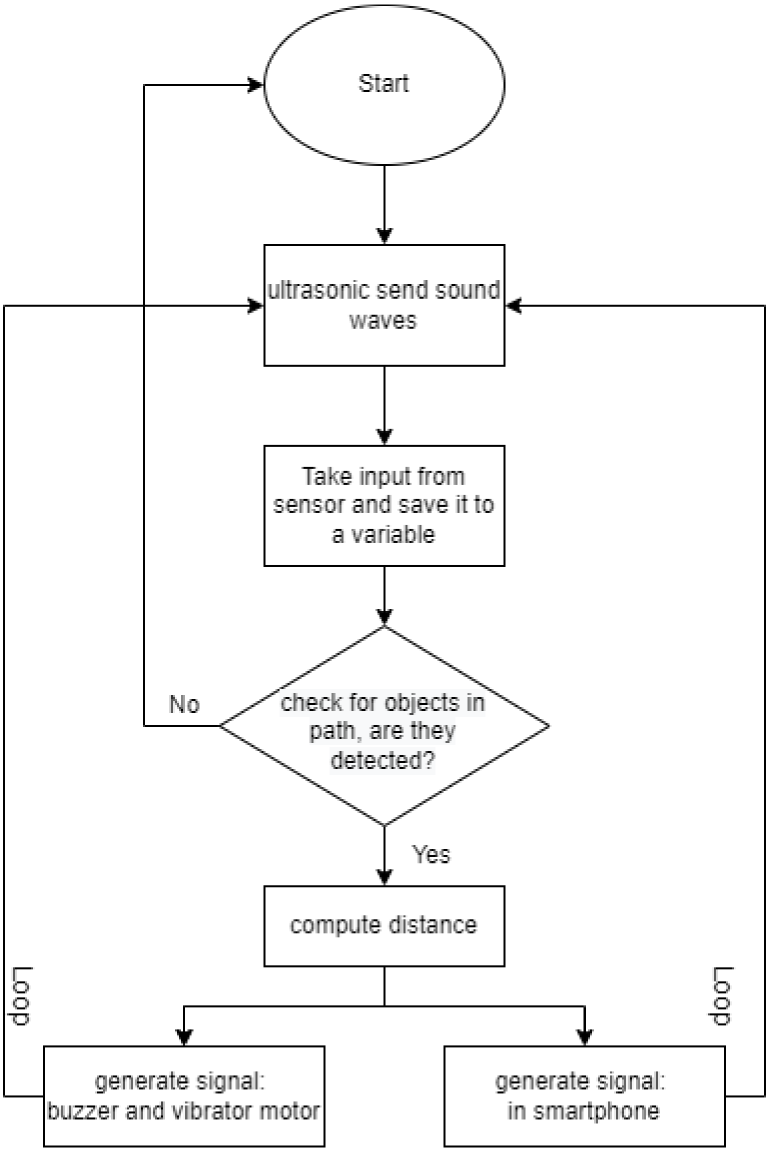


Figure 12. Flowchart of the program

The Echo pin in the Ultrasonic sensor, which represents an output pin, receives an electrical response from the sensor. This response illustrates the time it takes for the wave to travel forth and back from the obstacle. Mathematically this could be represented by *Equation 1* that we illustrated above, where duration is the Echo output, divided by two because we are only interested in the one-way distance (20). Furthermore, the 29.1 comes from the Arduino *datasheet*, which gives us the distance in cm instead of meters.

4.5 Implementation Of the App

The purpose of implementing the app is to notify the visually impaired people through vibration signals generated from the cell phone device instead of signals generated from the vibrator motor installed on the user's shoe, which means that the user will receive better, more accurate, and more sensitive signals compared to the signals received from the vibrator motor. Therefore, this tool is useful when the user exists in an inappropriate environment, where signals are not easily sensed. The software part of the SafeWalk project was developed using Java in Android Studio. The reason behind picking Java is the fact that it works for native as well as cross-platform apps, meaning that the app will work on a specific operating system, in our case Android. Besides, Android Studio is built on the Java language, meaning that there are plenty of Java libraries and frameworks that could be used to your aid.

However, to start implementing the app and after connecting the HC06 Bluetooth module device, we need to create a Bluetooth connection between the Arduino and the HC06 and likewise for the HC06 and the Android app. *Figure* 13 below shows the Bluetooth connection that needs to be created between the Arduino microcontroller and the Android device.

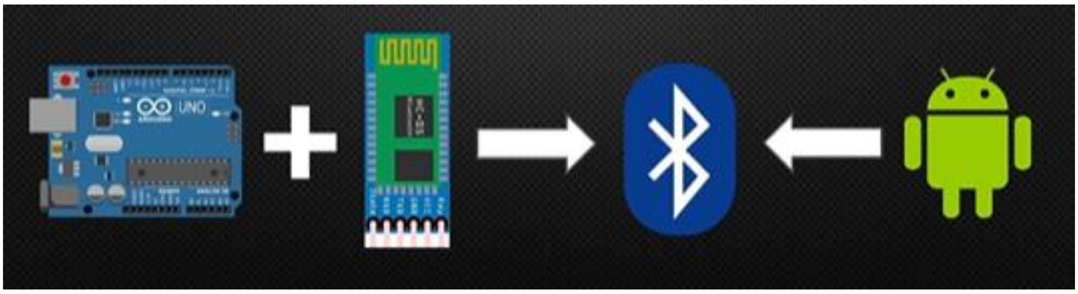


Figure 13. Connection between devices

To create a Bluetooth connection on the Android app, we first need to:

- Initialize the Bluetooth device on the Android phone.

- Get the MAC address from the device that we are connected to.

- Initiate a connection using the MAC address we obtained previously.

- Create a thread that manages the state of the connection, e.g., if the connection failed or succeeded. Moreover, this thread will manage data exchange. In other words, the data transmitting and receiving between the two devices.

- Finally, we need to create another thread to read the incoming transmitted data.

After establishing the Bluetooth connection, we need to create the app itself. It will be user-friendly, consisting of only one activity page. This page will include a button to search and connect to the Bluetooth device. The button is located at the top right corner of the main page, making it easy for the users to find. After clicking on the "connect" button, a list of available Bluetooth devices will appear. However, to make the application more visually impaired friendly, we programmed the application to always list the Bluetooth with the HC-06 at the top of the list. *Figure 14* below shows the location of the Bluetooth connect button.

Figure 14. Bluetooth Connection Button

To examine whether the connection is ready to start sending and receiving data between the hardware devices and the software application, it is preferable to implement a more straightforward application to check the connectivity. Thus, we choose to implement a simple app to turn on and off the LED. The idea behind this app is that when a button is pushed, it will send data to the Arduino microcontroller and make the LED turn on. In such a way, we confirm that the Bluetooth connection is ready and waiting.

After establishing a stable connection between the devices, the detection process begins. The ultrasonic sensor is activated, and it starts transmitting sound waves and receives sound reflected waves from an object. These signals help in measuring the distance to the object, and instead of notifying the user with the vibrator motor, the smartphone starts to vibrate.

5. Results and discussion

This section will describe how we tested the different parts of our project. Starting with the tests of the components and testing how accurate the data from the sensors are. The following section includes tests about the app and if there is a delay from the data captured until it is signaling the user by the phone. In addition, subsection 5.2 includes the problems that occurred while testing the app and other problems that we thought of. Under each problem, we include a possible solution.

5.1 Testing

As mentioned above, this section will include the tests we have done on the software and hardware. We have been testing the app since the day we started programming it. The first thing we had to test was to ensure that we were able to connect the smartphone to the Bluetooth module. To know if the module is connected to anything, there is a little LED that flashes during pairing mode, but when it is connected, the LED lights continuously.

Firstly, the components to be used should be fully working. Thus, we had to test every single device on its own to ensure we could go further and connect them. To test the components, we wrote an Arduino program using C++. The first test was to ensure that the Ultrasonic sensor did measure the distance correctly. To know that the distance and formula for the distance were correct, we used a ruler and placed the sensor at the beginning and then held an object at different distances to see if the output was accurate. *Figure 15* below shows how we set up the testing for the sensor.

Moreover, we needed to test the vibrator motor and piezo buzzer. The method to test these two components are the same. We connected the positive end to the 5V pin and the negative to the GND. By doing this, we should receive a constant vibration or buzzing sound, which we got.

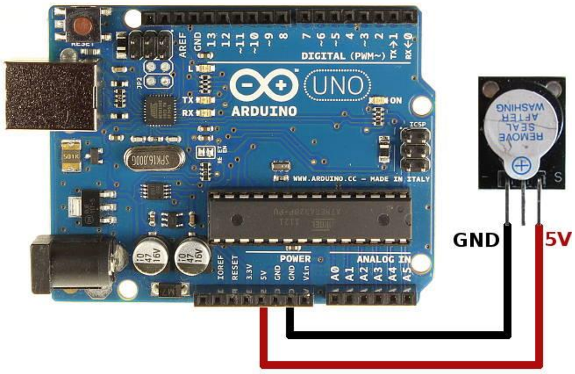


Figure 15. Testing method of Buzzer and Vibrator moto

After we connected the devices, we could test the entire prototype. A good test to start with was to test how accurate the measured data is. By transmitting the sound waves, we could calculate the distance from the obstacle and convert it to cm. This data was compared with the real distance measured by us. When we knew that the data was accurate, we could test the system in real-time. Meaning that the user can finally rely on the prototype and use it. The real-time test is a matter of how safe the system is; can it be trusted? Finally, a practical communication test between all devices to ensure that there is no delay when the object is detected and tone generated. *Figure 16* shows the setup for the test, and *Figure 17*, shows the output, which is the distance between the sensor and the box on the Serial Monitor.

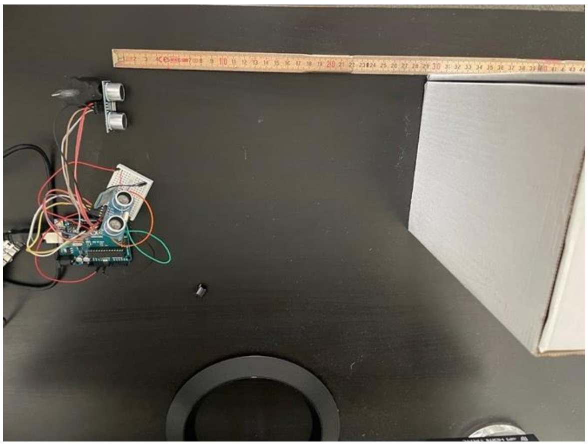
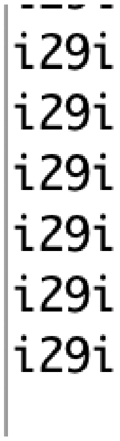


Figure 16. Testing of Ultrasonic sensor



```
i29i
i29i
i29i
i29i
i29i
i29i
```

Figure 17. Output of the Ultrasonic sensor test

The next step was to see if we were able to send values from the Arduino IDE to the Android Studio. To do this, we used the built-in Serial Monitor on Arduino IDE, and there we could see all the sensor data coming from both the top-mounted ultrasonic sensor and the one that faces straightforwardly. At this point, we knew that it would be possible to read the data into the Android Studio, so we then used a built-in function to retrieve the values from the Serial Monitor.

We had one huge problem that we were clueless about how to solve. When we send the data to the Serial Monitor, it is sent as a string. The original plan was to compare the string to parameters in Android Studio, although we were never able to make the program go into the if() statement. This was because when we send a string from the Arduino IDE, it sends extra information, not just the value from the ultrasonic sensor. We studied for many days how to fix this. We tried to see if there was a way to send an integer from Arduino IDE, but the only way we could find was to print on the Serial Monitor. Then the Android Studio would read from that same monitor. So the solution to this problem was to add the letter "i" before and after the sensor value. If the ultrasonic sensor gives us the value 180, the program will set the string to "i180i". Then we send this to the Serial Monitor, and in the Arduino Studio, we use a function called "split," which retrieves the value between the two i's. In *Figure 17* above, we can see the i's that surround the data that the sensor is emitting.

When we finally got the value into the Android Studio, we wanted to insert this into the vibrating function. However, we quickly realized that the vibrating was stronger the further away the objects were, which makes sense since the distance is longer. To fix this, we used the straight-line equation to invert the value and set the maximum distance that the sensor would measure.

In this stage, we had at least a Bluetooth connection that worked, and an app that displayed the value for the users. A phone that vibrates linearly to the distance, and a buzzer and vibrator on the shoe that signals the user for objects up in the air. To test the app and see the delay, we used the Serial Monitor on the Arduino IDE and the value that is displayed on the phone. Unfortunately, we experienced a delay of about 0.5 s. We tried to tweak the code as much as possible, but we believe that the delay is because the Arduino might send it with a bit of delay. As well as, the Bluetooth module might receive it with a delay and not instantly. There is, of course, a problem with this, but the solution for this problem is out of time and scope of this project.

We completed a test to see how long the device could run without being charged. As mentioned, we have calculated the time for the components. However, a power draw test with the code running is mandatory. We connected a multimeter to the positive side of the battery and a positive output pin on the Arduino. This gave us an accurate power draw of 0.4 amps per hour. With the help of this information, we can estimate the time that the device needs to be fully discharged. In our case, it was around 3 hours. This is something that we will need to improve in further work. Ways to do this are to use a larger battery, use components that are more energy-efficient and optimize the code.

A final test was executed with the finished prototype. The prototype was placed on top of the sweeper, about 3 cm above the ground, simulating it being attached to a shoe. The app was connected, and the testing began. The test was done in two different conditions, first in daylight with the sun shining through the windows, and secondly in the middle of the night when it was completely dark. The device worked just as well during both of the tests. It never missed an object, even if it was small like a candle. Therefore, a conclusion can be made that lightning does not matter at all for ultrasonic sensors, and neither does the size of an object. As long as it is in front of the sensor, it will be detected. The sensors were very good at sensing objects in the path; not once was an object or wall missed. However, the app's delay is a slight problem since the walking pace had to be slower than regular. This was so we had time to wait for the updated vibration in the phone. However, if we can eliminate this delay issue with further work, we do believe that this device can exchange the white cane.

Another problem we discovered while watching the Serial Monitor was that the sensors sometimes send the wrong data. We put the forward-facing sensor on a desk, held an A4 paper

200 cm away, and then moved forward. When we then looked at the monitor, the values were 99% of the time accurate, but 1% were totally wrong. We did not feel anything in the phone on this, but if we used the intervals as the original plan was, there would be lags and unsafe for the user.

5.2 Problems and solutions

Improvement is always a part of our project. Therefore, we always seek to find an optimal solution. During the testing phase, we discovered some problems that could be solved as well as problems that we knew we would face during the implementing process.

1) There is a delay from the time that the sensor finds and sends the signal to the vibration in the phone.

When we first send the signal from the sensor to the board so that the buzzer informs the user, there is not much delay. We tried this by hovering the hand in front of the sensor, and then we got a response very fast. However, when we send the signal to the phone, we have a bit of a delay. After comparing the buzzer and the phone, we can conclude that the delay comes from Bluetooth or the app. We do not have any delay function in the code for the app, so the problem is probably that the Bluetooth module that we use is not fast enough. A solution to this would be to use a better Bluetooth chip or perhaps change how we send and retrieve the value from the sensor. Nevertheless, since we did not have unlimited time and resources, this code and Bluetooth module was what we had to use.

2) The phone and shoe will vibrate continuously when standing in a cue.

Our initial plan was to code the program so that when the sensor sends a different value compared to the one before, the phone and shoe will vibrate. Then if the value is the same, it would not. As mentioned before, the problem with this concept is that the sensor is not good enough. The sensor can first send a different value than what it "should," but it could also send two to three different values for the same distance. Let us say that an object is 50 cm away, and then the sensor would send the values 49, 50, and 51. So by coding, we were not able to solve the problem. We then thought about using an accelerometer. Although the app took much longer to program than planned, we did not have enough time to implement an accelerometer. But if we continue with the project, we will try to implement the accelerometer.

3) The device is not waterproof.

Our box is not waterproof. If we were to continue with this project, an easy solution is to use a 3D printer to print a case with the exact dimensions. Inside the case, we would carve a path for a rubber seal so that when the lid snaps into place, it would be waterproof. Figure *18* shows a red seal that is inserted into the lid.

Figure 18. Arduino waterproof box

4) The sensors are not 100% correct and sometimes send a value with a certain margin of error.

The reason for that depends on the properties of the Ultrasonic sensor. Ultrasonic sensors have low resolution and slow refresh rate, making them not suitable for detection of fast-moving objects. In addition, they are unable to measure the distance of objects that have extreme textures or surface. The solution to this problem is probably to study more about the different kinds of sensors on the market. Find a more accurate sensor and, most importantly, a sensor that does not give a wrong value at some times.

5) The sensors do not detect low objects.

After putting the components in the box and trying to walk with it, we discovered that it was not as good as we had hoped. The sensor does not sense if a root is pushing up the concrete because the level difference is too small. Since the sensor is on top of the shoe and it is parallel with the ground, it does not detect objects lower than the sensor. If we were to angle the sensor down, the phone would constantly vibrate since it is always sensing an "object." A possible solution to this problem is to have the sensor in front of the shoe and lower to the ground. However, this would have to be tested to see if it actually works to walk with so it does not break.

5.3 Results

Starting with the good results. The sensors work; they can sense if there is an object in the path and then sends the signal to the phone. The phone's connection with the Bluetooth module is very strong, and the phone has never disconnected. This is very important since the user must have the device connected when they are walking. When the sensor that senses objects in the air finds one, the delay in notifying the person is very small. This is very good because the person has much time to slow down the pace and then either walk to the sides or hover their hands in the air to find the object.

The two bad results that we noticed during the testing were that there is a delay from the moment the signal on the ground is sent to when the phone starts to vibrate. The second bad result is that the sensors sometimes send the wrong value. Therefore we do not have a solution to the sensors turning off in the ques.

Below are pictures of the app in different states and a picture of the device that will be attached to a shoe. *Figure 19*, as shown below, is the home screen of the app. In the top right corner of the picture, there is a "CONNECT" button. If pressed, a list of available devices will appear, as seen in the next picture. *Figure 20* is the screen after the "CONNECT" button has been pressed, and there is the list of available devices to connect to.

Figure 19. App page activity to connect to Bluetooth

Figure 20. App page Activity to find Bluetooth device

After a device has been picked from the list, the user will come to a loading screen, as seen in *Figure 21*. If the device fails to connect, a message appears at the top of the screen under "Bluetooth Connection" the message is shown in *Figure 22*. Although if the device successfully connects, the message displays what device the app is connected to. The numbers, as seen in *Figure 23* and *Figure 24*, are the strength that the device is vibrating with.

Figure 21. Loading screen

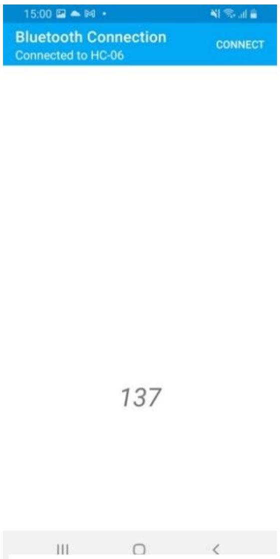
Device fails to connect

Figure 22. Connection failed

Connected to HC-06

0

Figure 23. Connection succussed



Connected to HC-06

137

Figure 24. Vibration strength

Figure 25 below shows the box, and it is made from a plastic box that is modified so that the components fit well. Since this is only a prototype, the box is fairly large, 10x11 cm and 5 cm tall. The back end of the device is open; the reason for this is so we have access to the battery. In the future, our goal is to develop this prototype into a product that will work without any problems such as delay. To be able to turn this into a product, we wish to partner up with a company and perhaps even get a patent on the device.

Figure 25. Prototype model

To measure the accuracy of the device, a simple test was done. *Table 1* below shows the data that was achieved during this test. The column to the left is the distance between the object and the sensor, the middle column is the distance that the sensor measured, and the right column is the delay from when the sensor was moved to the new distance and when the vibration in the phone was changed. By completing this test, we see that the prototype works as it should. The distance is almost perfect, and it has a margin of error of about 1 cm, which is very little and negligible due to that the user will not feel much difference at all in the vibrations. The delay was measured with a timer, so it is not completely accurate, although the average delay is 0.32 seconds.

Table 1. Measured data

Actual distance in CM	**Distance output Serial Monitor** in CM	**Delay in sec**
180	179	0.4
150	148	0.3
120	121	0.2
90	89	0.5
70	69	0.2
50	51	0.3

To determine the reasons behind the inaccurate data, we developed a set of test scenarios. The system has been tested on objects with different colors and different levels of transparency to see the performance of the system in such conditions. This was not affecting the performance of the ultrasonic sensor as they reflect sound from objects, so the color and the transparency have no effect on the sensor's reading.

Another test scenario to check the inaccurate data was to test the sensor's performance in dark environments. However, the ultrasonic sensor detects objects by reflecting sound waves. Thus, the dark environment has no effect on the detecting performance of the sensor. Different from proximity sensors that use light or cameras, the ultrasonic sensor has a good performance in dark environments.

However, when the system is tested for sensing soft materials, the reading data from the ultrasonic sensor gives more and more inaccurate data. This depends on the characteristics of the soft material. Objects covered in a soft fabric absorb more sound waves than other materials. Therefore, it is harder for the sensors to detect the obstacle.

Having a small target object affects the distance measuring ability of the ultrasonic sensor. Small objects do not reflect the sensor's signal sufficiently, meaning that the distance measured can give incorrect data.

5.4 Affordability and costs

One of the most important specifications of this thesis is to build a system that would provide the same functionality that other systems provide, yet with cheaper components and a price that is fair for almost everyone. It was mentioned before that in order to complete this project, Halmstad university provided us with almost all the needed parts. However, *Table 2* below shows the prices on Amazon.com for the hardware components in detail, assuming that the user owns an Android device.

Table 2. Price for the components

Component	**Price**
Battery (2100mA)	10$
Ultrasonic sensor	3$ x 2 = 6$
Arduino Uno	25$
Buzzer	2$
Vibrating coin	2$
Bluetooth module	10$
Total cost	55$

As shown in *Table 2* above, we can see that the price for the components in total does not exceed 60 dollars. In comparison to similar tools and projects, as mentioned in the background section, we can see that SafeWalk can be defined as a low-priced system. In addition to the table above, we have the engineering costs. We estimated that we worked 640 hours on the prototype, although when we want mass produce, it will not be very time-consuming. Therefore, we will add about 100$ to the price, so the total price for customers will then be rounded to 155$.

6. Conclusion

To recapitulate, the main objective of this project was to implement an application to help visually impaired people in their daily life. Today's system is fully functional and works as supposed, although it could be better if more accurate sensors were integrated with the system. However, SafeWalk was first implemented to provide the visually impaired with a tool that can help them in their daily routine.

The project is divided into three main stages. The first stage is where we create a prototype, including the hardware components. In this stage, we implement a prototype consisting of Arduino code, a vibrator motor, and a buzzer to help the visually impaired detect the obstacles in their path. The second stage was establishing a good Bluetooth connection between the involved devices. The Bluetooth module works here as a middle-man service. It transmits and receives data from the desired device. Finally, the third stage consists of combining these parts together into an Android app. Summing up all these parts together produces a prototype that can signal the visually impaired by either vibration generated from a smartphone device or by signals coming from the vibrator motor and piezo buzzer.

The application is very user-friendly and runs on all Android systems for free. It could be improved by adding other extensions and functionalities, such as the very nowadays known topic called "real-time 3D-mapping systems", which is a system that helps blind people to navigate with the help of google maps. However, these systems are very costly in comparison with SafeWalk. As previously mentioned, InnoMake sells such tools for more than 3000$, while the SafeWalk system is less costly.

Compared to other projects in this field, SafeWalk has the lead. There are two main reasons for this, and the first one is because the device senses object both on the ground but also in the air. The second one is the app that has been implemented to vibrate, this gives the user a stronger vibration, and it is easier to feel where the object is. If it is on the ground, the phone will vibrate, but if it is in the air, the shoe will vibrate instead.

The results achieved so far are very satisfying in following our project plan. SafeWalk is not ready to go to production, but as it stands today, we have a fully working prototype as we aimed. Improvements are our biggest concern, such as fixing the unwanted delays, adding other extensions and functions, covering edge cases, e.g., walking up a stair, and even adding talking tools to guide the users via headsets and maybe some other design issues. However, as time is nothing we have control over, it satisfies us with what we have achieved today, which is what we aimed for till this point in time.

7. UtExpo

The device will be at the UtExpo with a phone so that people who visit can see the device in action. The initial plan is to have the device on a table so people can hover their hand in front of it and feel the phone vibrate and hear the buzzer make a noise. Although if someone is interested in trying the device on the shoe, that will be possible.

8. References

1. Portch R. Quora. [Online].; 2017 [cited 2022 March 5. Available from: https://www.quora.com/What-are-the-problems-that-the-visually-impaired-face-with-the-white-cane.

2. Kurniawan RMaS. Watch Your Head, Mind Your Step: Mobility-Related Accidents Experienced by People with Visual Impairment. Santa Cruz: University of California, Department of Computer Engineering; 2010.

3. Abbas Riazi FRRYaFB. Outdoor difficulties experienced by a group of visually impaired Iranian people. Tehran: School of Medicine, Department of Ophthalmology; 2016.

4. Strong P. The history of the white cane..

5. Luan V. Nguyen HMLJS,TV. A Smart Shoe for building a real-time 3D map. Reno: University of Nevada at Reno, Department of Computer Science and Engineering; 2016.

6. Ashwani Kumar SSSSRVK. An Object Detection Technique For Blind People in Real-Time Using Deep Neural Network. Hyderbadad: Vardhaman college of Engineering, Department of CSE; 2019.

7. Senem Kursun Bahadir VKFK. Wearable obstacle detection system fully integrated to textile structures for visually impaired people. Istanbul: University of Lille North of France, Textile Engineering Department ; 2012.

8. Teja Chava ATSALSVR. IoT based Smart Shoe for the Blind. Andhra Pradesh: Koneru Lakshmaiah Education Foundation, Department of Computer Science and Engineering; 2021.

9. WHO. [Online].; 2021 [cited 2022 march 10. Available from: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

10. Aline Darc Piculo dos Santos FOMMJCARGRFES. Are electronic white canes better than traditional canes? A comparative study with blind and blindfolded participants. Long Paper. Bauru: Sao Paulo State University, Department of Design; 2020.

11. Larisa Dunai Dunai ILLITFBS. Obstacle detectors for visually impaired people. Universitat Politècnica de València, Research Center in Graphic Technology; 2014.

12. B.Chandran SMA. Low-cost ultrasonic-based shoe for visually impaired people. In Smart Healthcare for Disease Diagnosis and Prevention. Shillong; 2020. p. 103-114.
13. Blog D. Medium. [Online].; 2020 [cited 2022 April 2. Available from: https://medium.com/swlh/create-custom-android-app-to-control-arduino-board-using-bluetooth-ff878e998aa8.

14. Fawad. programmingdigest. [Online].; 2019 [cited 2022 April 02. Available from: https://programmingdigest.com/arduino-bluetooth-controlling-system-for-home-automation-using-arduino-bluetooth/#Final_Design_of_Home_Automation_Android_Bluetooth_App.

15. Fahad E. Electroniclinic. [Online].; 2019 [cited 2022 April 02. Available from: https://www.electroniclinic.com/how-to-create-android-app-for-arduino-sensor-monitoring-over-bluetooth/#Watch_Video_Tutorial.

16. Omoron. The technical explanation for ultrasonic sensors..

17. microcontrollerslab. [Online]. [cited 2022 April 20. Available from: https://microcontrollerslab.com/ultrasonic-hc-sr04-sensor-arduino-tutorial/#Measure_HC-SR04_Echo_Pulse_Time_with_Arduino.

18. Guangzhou HC Information Technology Co. L. Product Data Sheet..

19. MR.KUMARA B A MHMMSGMCG. INTEGRATED SMART SHOE FOR BLIND PEOPLE. TUMAKURU: SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY, DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING; 2016.

20. Dhruvin Patel HPHPNP. Electronicsforu. [Online].; 2017 [cited 2022 April 20. Available from: https://www.electronicsforu.com/electronics-projects/smart-stick-using-arduino-ultrasonic.

hh.se