eyeAssist

By

Annamika Dua [ad8]
Sahil Kamesh [skamesh2]
Veda Menon [vedanta2]

Final Report for ECE 445, Senior Design

TA: Xihang Wu

May 5, 2021

Project No. 40

Abstract

Life for the visually impaired can be very difficult at times, and without outside help it can be hard for them to live their daily lives. Our assistive eye-glasses aim to tackle two of the main issues that people who are visually imapired go through. The glasses provide users with the ability to read books in real-time, as well as navigate their surroundings safely. In our report, we discuss in detail the design of our product, each of its components, as well as how they are integrated with each other to build our glasses.

Contents

1. Introduction

1.1. Background and Objectives

People with visual impairments represent a unique demographic who face challenges that may not be widely understood by the general population. According to the World Health Organization, there are an estimated 285 million people across the globe who are visually impaired. Out of those 285 million, around 39 million are completely blind [1].

Mobility is an extremely crucial part of our lives; we depend on the ability to effectively navigate through any environment we are in every single day. The visually impaired often have trouble navigating through their environment without some kind of assistance. Additionally, reading any kind of text, whether that be a book or an important document, can also be burdensome. They may also have difficulty accessing audiobooks online, as those books must be pre-recorded before being sold to the public. This can be extremely frustrating and serves as a significant limitation in their lives. Developing tools to cater to these difficulties, especially through technology, can provide this demographic with a new range of abilities that may have been previously inaccessible.

Our solution is wearable multi-purpose "smart" glasses. These glasses aim to tackle both issues of mobility and reading, enhancing the quality of life of those with visual impairments. This product was designed to help the visually impaired navigate their home with ease as well as give them the ability to read text in real-time.

Ultrasonic sensors mounted on the glasses are used to detect any obstacles close to the user within a certain field of view. We will then provide feedback to the user regarding the direction and distance of the obstacles. This design improves upon traditional navigation canes by eliminating the need for constant physical effort on the user's part. The glasses will also have a built-in camera that can detect and capture any text the user is looking at using OCR, and read it aloud to them with a text-to-speech converter.

1.2. Visual Aids

Figure 1: Visual aid for obstacle detection capabilities - objects ahead of the user are reported while they are in motion

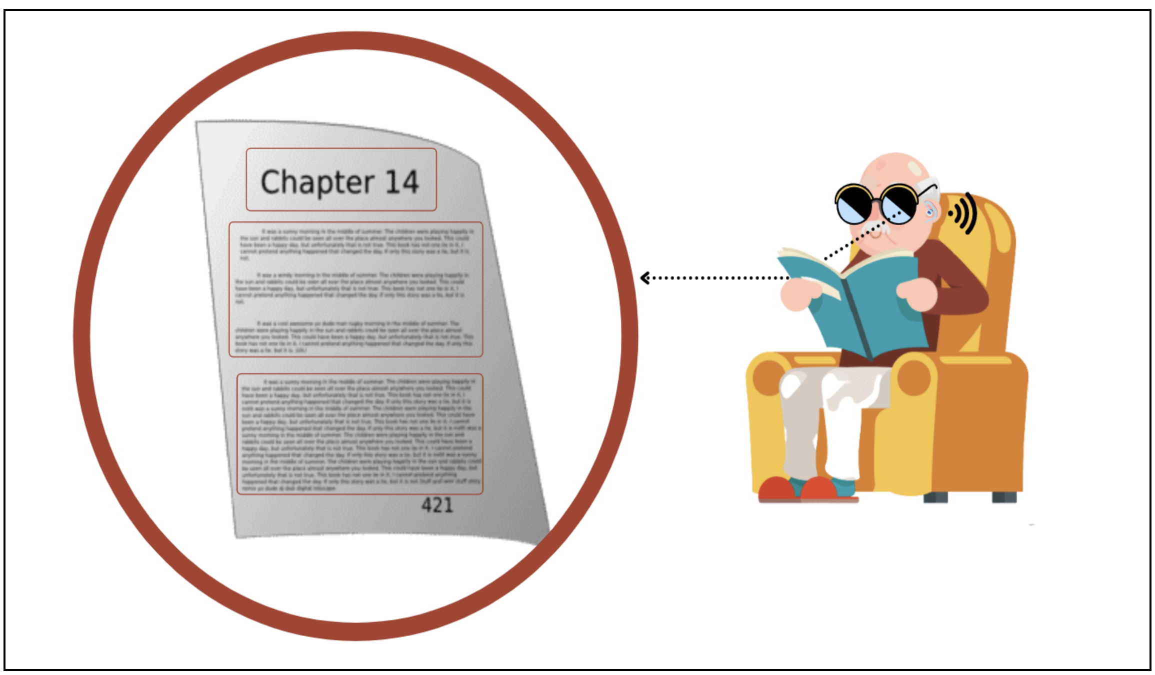

Figure 2: Visual aid for reading capabilities - a snapshot of text is converted to audio in real time for the user

1.3. High-Level Requirements

- The device should be able to detect and read an image of clear, unobstructed text within 30 cm (1 foot) of the user with an accuracy of at least 85%, within 3 minutes.
- The device should be able to detect obstacles at least 1.5 feet high in front of a mobile user at a distance of 1 meter and a field of view of 60 degrees, within an error range of 2 inches.
- The device should be able to operate without recharging for at least 4 hours.

2. Design

2.1. Block Diagram

Our design consists of four main modules: power module, reading module, obstacle detection module, and a user interface module. The power module ensures the steady operation of the design, and powers all of the subsystems, meeting the high-level requirement of operation without recharging for up to four hours. The reading module is used to capture text with the camera, accurately extract it with computer vision, and perform the text-to-speech conversion, meeting the high-level requirement of reading text with an 85% accuracy. The obstacle detection module periodically signals the ultrasonic sensors to send out ultrasonic pulses. The sensors then collect obstacle data and send it to the microcontroller, which processes this data and accordingly communicates with the audio system in the user interface module. The user interface module contains the push button, which the user operates to choose which module to activate, and the audio system, which notifies the user of any obstacles around them. All of this can further be seen below in Figure 3.

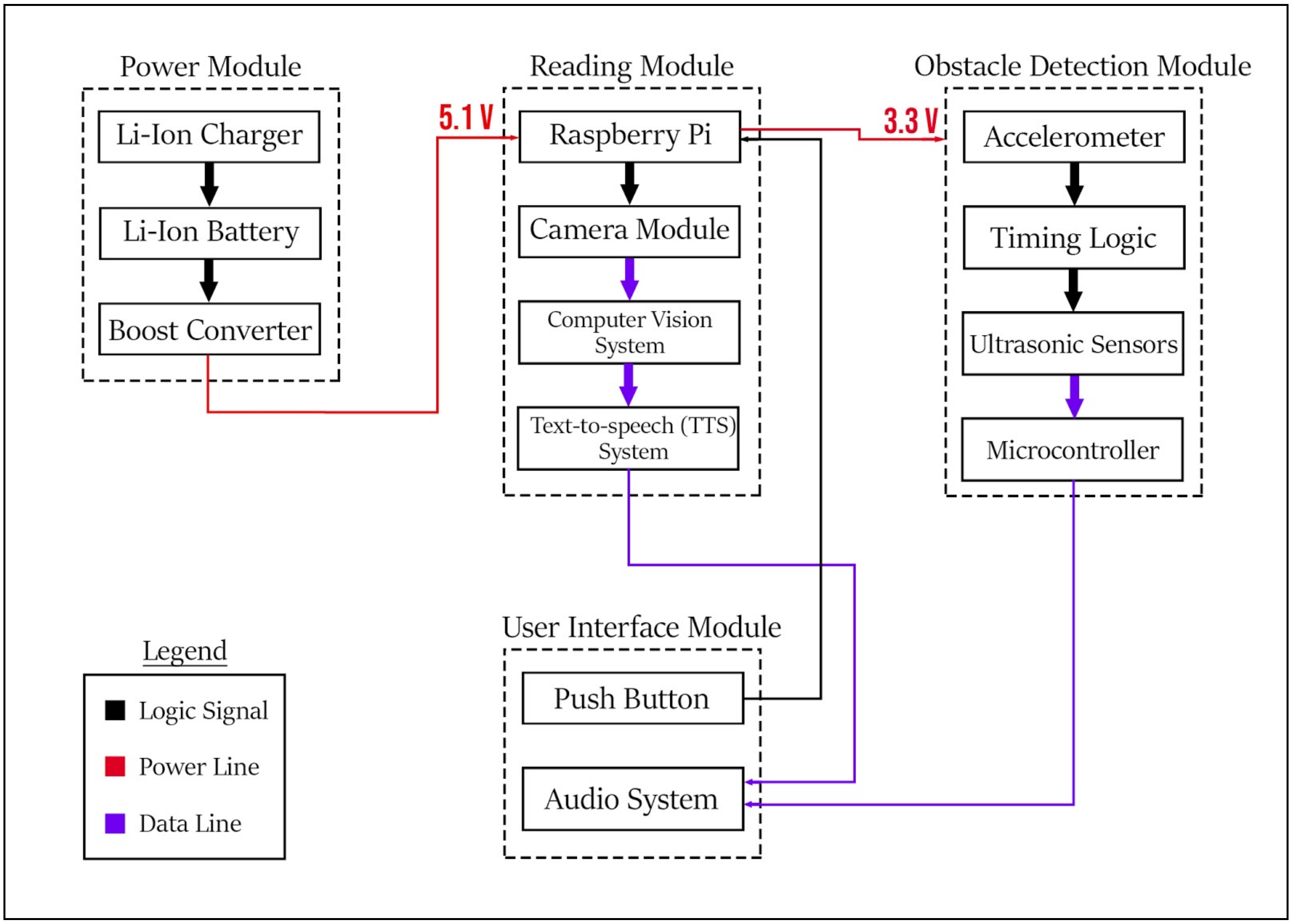

Figure 3: System block diagram

2.2. Physical Design

The project consists of several modules attached to a pair of glasses as shown in Figure 4. The first module is the obstacle detection module, which consists of two pairs of ultrasonic sensors that can be seen mounted on top of the glasses. The second module is the reading module, characterized by the camera mounted on one side of the glasses frame. All the other components that are needed for operation including the Raspberry Pi, the battery, and the power booster are placed in the box attached to the glasses as seen in Figure 4.

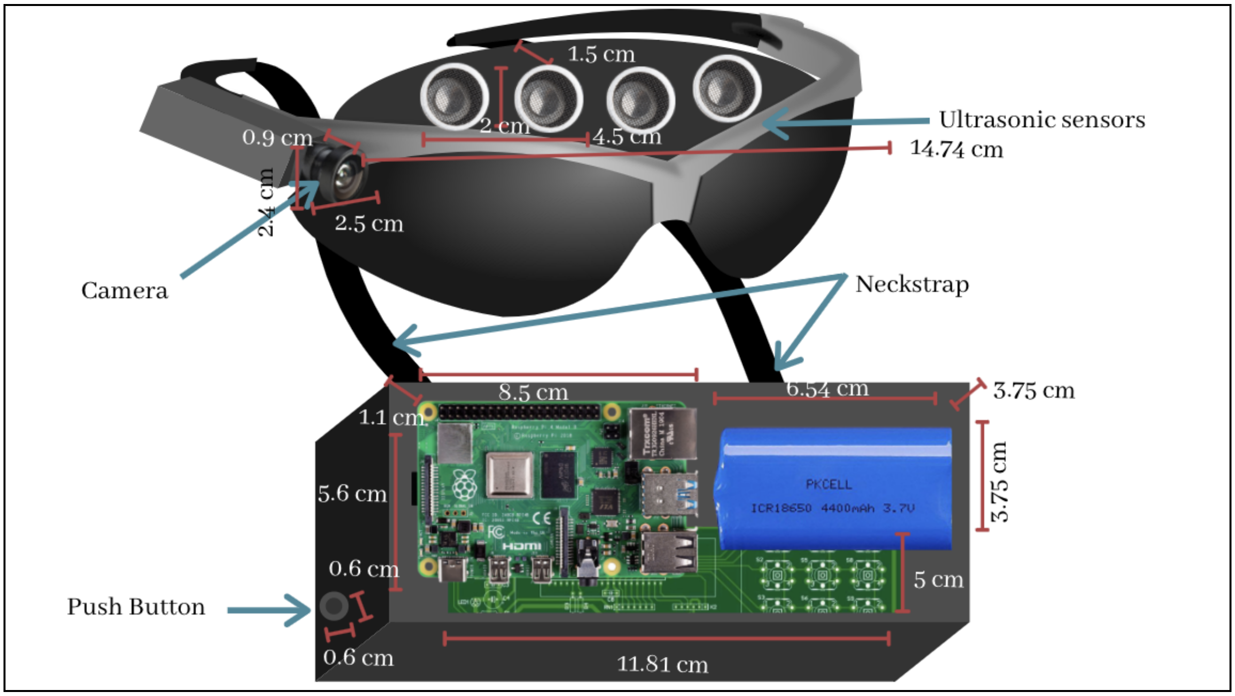

Figure 4: Physical design of eyeAssist glasses

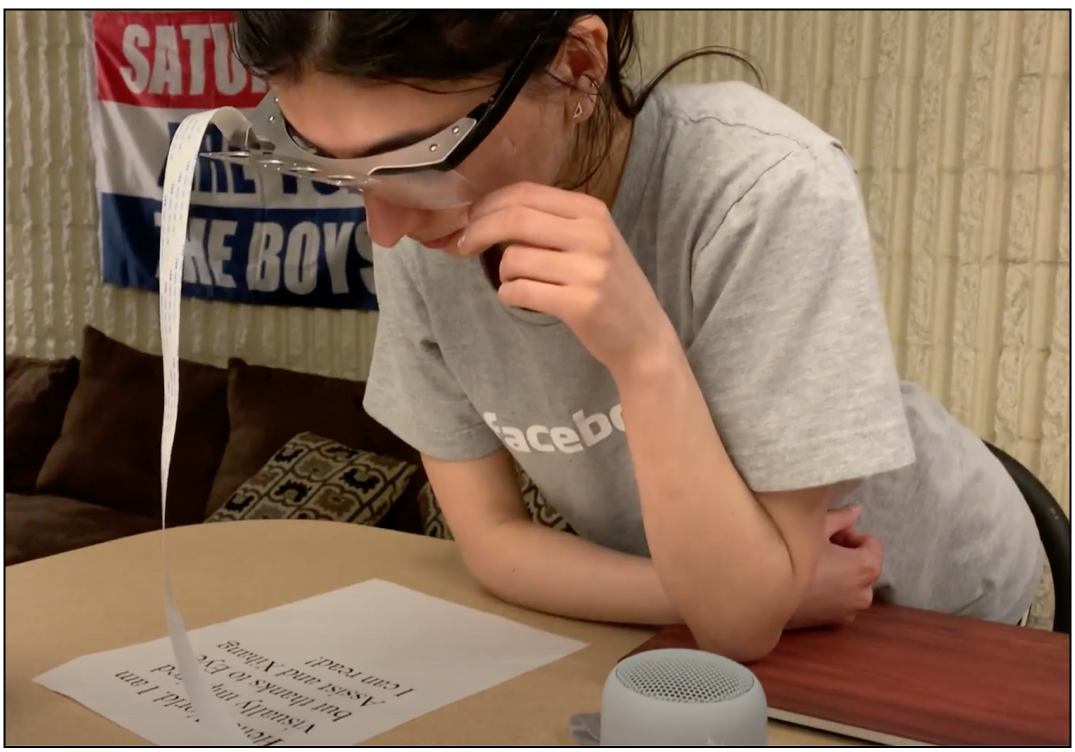

Figure 5: The physical design of our product in action

2.3. Flowchart

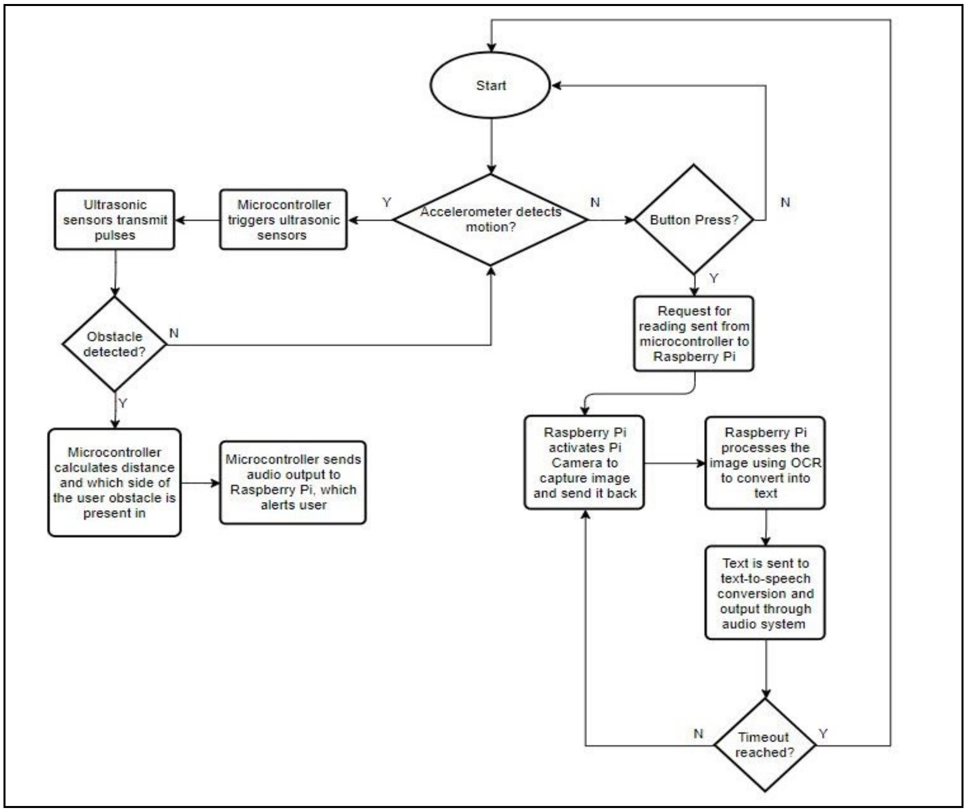

Figure 6: Flowchart of the design, depicting the device operation logic

2.4. Schematic Diagram

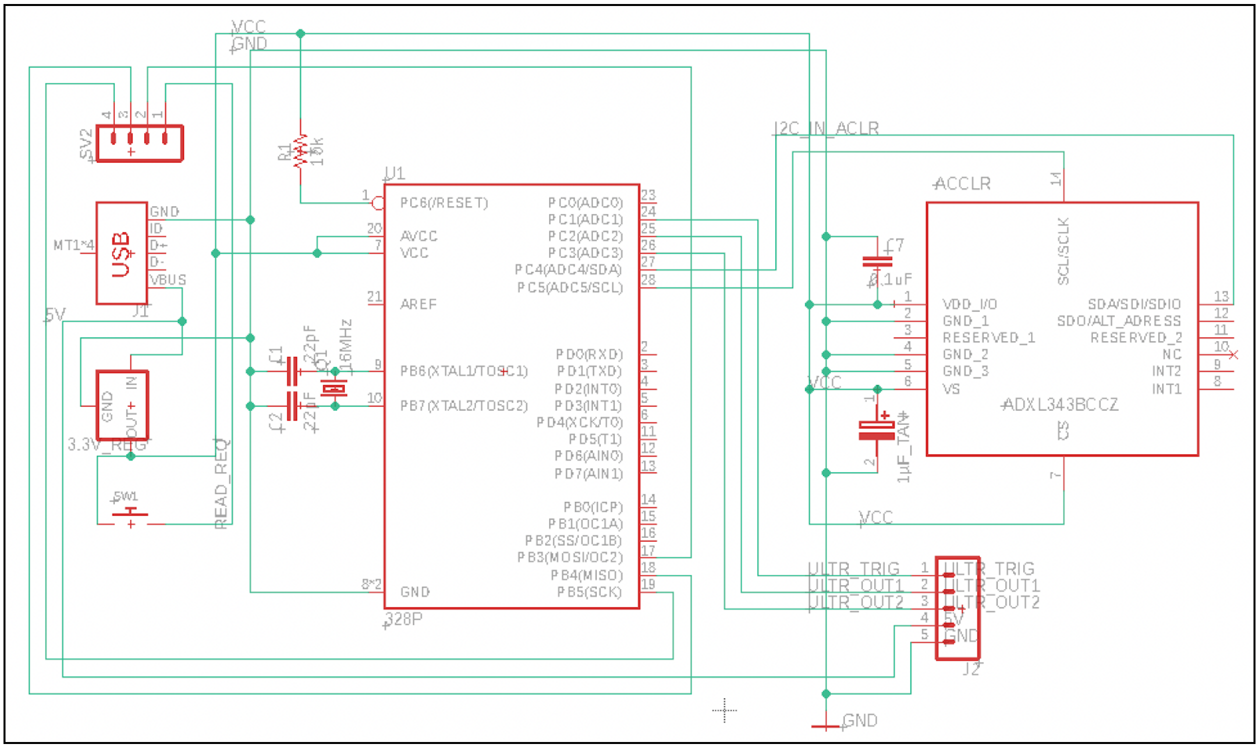

Figure 7: Circuit schematic used in the obstacle detection module

2.5. Power Module

This module allows for consistent and safe power distribution throughout the components as well as within the system. Specifically, it is used to power the Raspberry Pi and the obstacle detection module. To achieve this, we use a 3.7V Lithium Ion battery in order to power the Raspberry Pi, along with a power booster to get the voltage up to the required 5.2V. We also use the USB port on the Raspberry Pi to power the PCB with the required 3.3V.

2.5.1 Lithium-ion Battery

The Lithium-ion battery supplies power to the reading module, obstacle detection module, and all internal components. This 3.7V 4400 mAh lithium-ion battery is used to ensure the product can operate for up to four hours before recharging and allows for increased mobility for the user.

2.5.2 Lithium-ion Charger

The lithium-ion charger provides a convenient method for the user to recharge the batteries via micro USB.

2.5.3 Power Booster

The power booster converts the 3.7V battery output to 5.2V DC to power the Raspberry Pi.

2.6. Reading Module

This module detects and processes text the user is looking at using OCR, and then performs text-to-speech conversion on it. High quality images of the text are captured through our camera and are then preprocessed using OpenCV libraries (including grayscale, thresholding, skew correction, cropping, and page segmentation) as shown in Figure 8. Next we use Google's Tesseract computer vision engine to process the image and convert it to text. This text is then used in the text-to-speech conversion component, and is then sent to the user feedback module to be read aloud to the user. These components together ensure that the text is read with at least 85% accuracy.

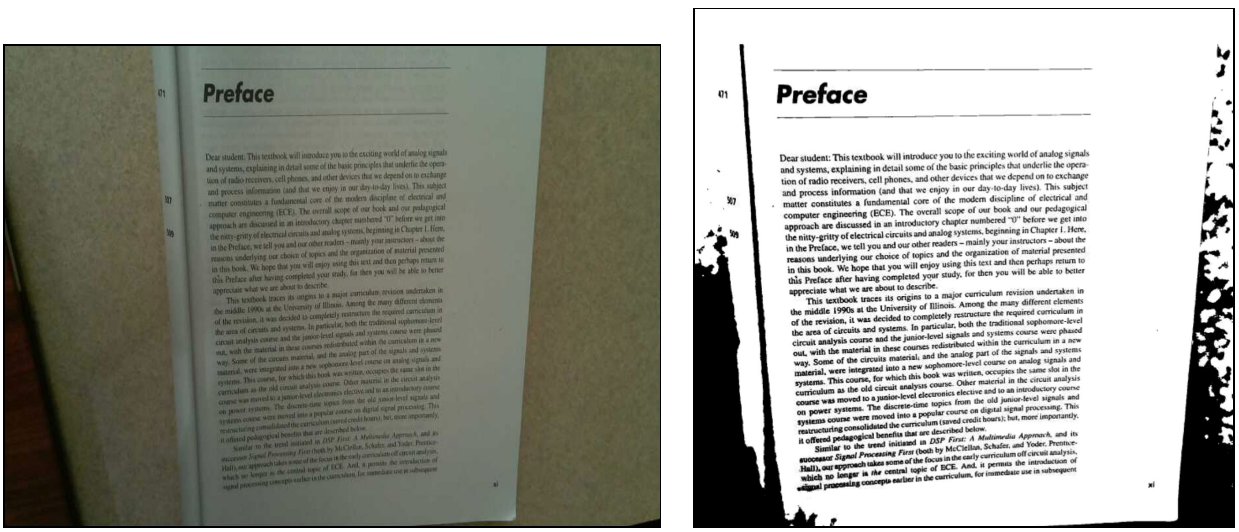

Figure 8: Images captured by Arducam, before and after preprocessing

2.6.1 Camera

The Arducam's 8-megapixel sensor captures real-time images of text placed in front of the glasses, which is then sent to the OCR system for processing.

2.6.2 Computer Vision System

This system runs OCR software, Google Tesseract, on the image captured by the camera to convert the preprocessed images into text, which is then sent to the text-to-speech system.

2.6.3 Text-to-Speech System

This system processes the characters imputed through the computer vision system, and outputs the audio of the text through the user interface module.

2.6.4 Raspberry Pi

The Raspberry Pi processes the images taken by the camera using OCR. It implements the text-to-speech system using audio output, and also processes the data from the microcontroller for audio output.

2.7. Obstacle Detection Module

This module is used to determine whether there is an obstacle in the way of the user's path. If the module detects any obstructions, it will also alert the user using voice feedback through the audio system, so that they can safely avoid the obstacle. This module would allow the user to detect obstacles within a 1 meter range and an angle of 50-60 degrees, within an error range of 2 inches.

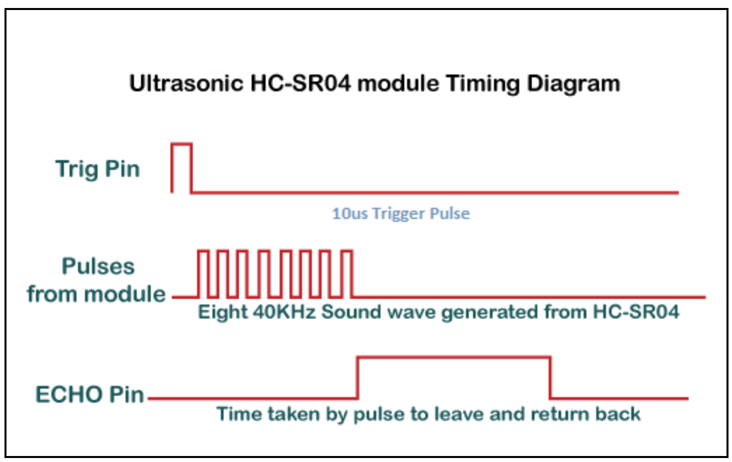

Figure 9: Timing Diagram of HC-SR04 ultrasonic sensor[3]

2.7.1 Microcontroller

The microcontroller ATMEGA328P is used to trigger the ultrasonic sensors by generating a high level signal of at least 10 µS as shown in Figure 9. The microcontroller then receives data from the ultrasonic sensors and calculates the distance at which the obstacle detected was located using the following equation (1). The calculated distances are then sent to the Raspberry Pi to be outputted through the audio system.

$$Distance = \frac{Speed * Time}{2} \tag{1}$$

2.7.2 Ultrasonic Sensors

Two ultrasonic sensors are used to detect objects in the path of the user, by transmitting eight 40 kHz ultrasonic pulses and checking if any signals are received back. Data related to the time it takes for a signal to be received as well as the sensor that picked up the signal is then sent to the microcontroller to determine the distance and direction of the obstacle.

2.7.3 Accelerometer

The accelerometer, though not used in our final product, was intended to serve as a trigger for obstacle detection by determining whether a user is in motion. Given that the user is in motion, the device would begin to detect objects around the user.

2.8. User Interface Module

This module determines how the user interacts with the physical device. The push button will be used for triggering the text-to-speech feature, and the audio system will be used to activate voice feedback. The audio will be transmitted through the headphone jack on the Raspberry Pi.

2.8.1 Push Button

The push button activates the reading module (short press) or the obstacle detection module (press over four seconds) depending on the length of the button press. This ensures that the device only takes images of text and sends them to be processed or detect objects only when required by the user.

2.8.2 Audio System

The audio system relays text-to-speech translation to the user, as well as feedback and alerts related to obstacle detection.

3. Design Verification

3.1. Power Module

Provided the lithium-ion battery we are using and the necessity for maximum mobility of the user, we ensured that our product can operate for up to four hours without recharging by running the reading and obstacle detection module for more than four hours without it running out of charge.

We also verified the power module requirements by measuring the power consumption of our module. The current draw of the Raspberry Pi while it was running our reading module and capturing pictures using our camera was 600mA. The current draw of our obstacle detection module from the Pi's USB port was ~100mA, giving a total of 700mA. Since our battery pack has a capacity of 4400mAh, this means our project should be able to run for over 6 hours with our power supply.

$$Hours = \frac{Capacity\ in\ mAh}{Current\ consumption} = \frac{4400}{700} = 6.28 \tag{2}$$

3.2. Reading Module

The primary requirement for our reading module was to achieve an 85% accuracy of text recognition and translation under 3 minutes. Without any pre-processing our accuracy averaged around 60%. We then improved upon this accuracy significantly after applying pre-processing techniques to the image captured by our camera. We first converted the image to grayscale to remove distractions by filtering background colors, after which we performed binary thresholding to further segment the image into foreground and background pixels. Finally we corrected for significant image skew and cropped the image to minimize the detection of stray text on other pages that may have been partially captured in the frame. For fine tuning we set specific page segmentation parameters as part of Google Tesseract to facilitate the detection of blocks of text in paragraph format.

After preprocessing the image, we verified our accuracy by testing our OCR on various reading material, fonts, font sizes and text alignments. We ran our OCR on the most popular fonts used - Times New Roman, Arial, Helvetica, Calibri, Garamond, and Verdana as well as on font sizes ranging from 10 to 60 pt font. Based on our results we found that the accuracy of these common font types remained consistent however the OCR, as we expected decreased in accuracy as the font size decreased as shown in Figure 10. Hence, for fonts larger than 10 pt we were able to achieve greater than an 85% accuracy.

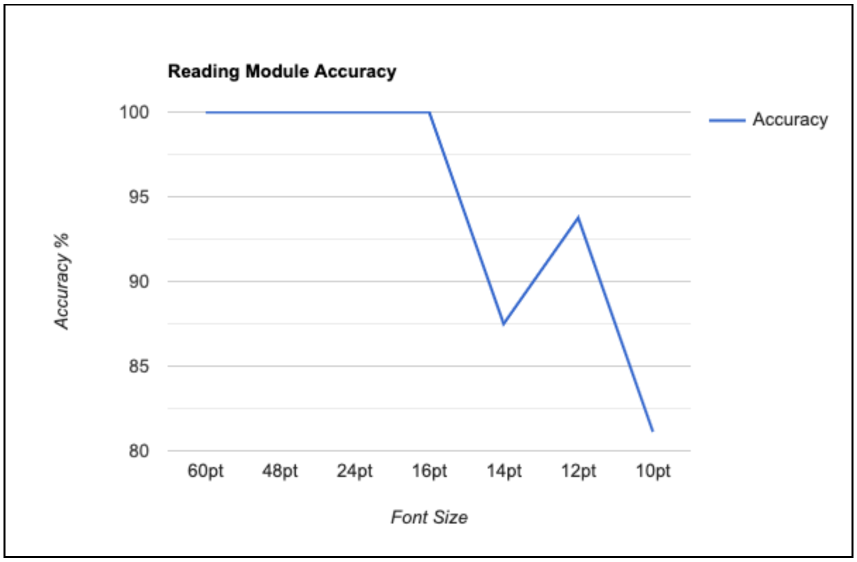

Figure 10: Graph showing the accuracy of reading module for various font sizes

Another interesting finding was that center aligned text performed about 15% worse than left aligned text on average in terms of accuracy. We also noted the times it took to read these various texts and found that the left-aligned Times New Roman font was detected nearly 5 seconds faster than the other fonts we tested as shown in Figure 11. On average we calculated short blocks of text took an average of 16 seconds and longer pages text around 300 words ranged from 35-45 seconds, satisfying our requirements.

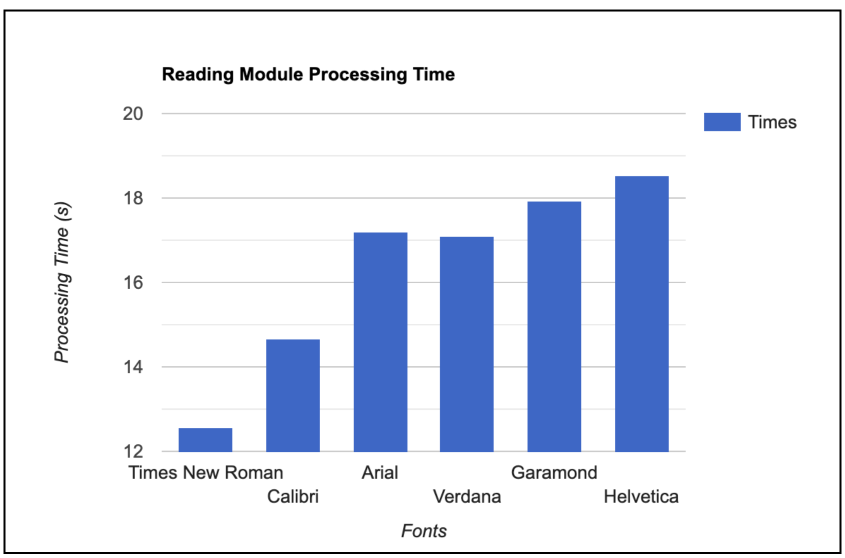

Figure 11: Graph showing the processing time of reading module on various font types

3.3. Obstacle Detection Module

Based on the HC-SR04 datasheet [4] and our tolerance analysis, we set the requirement that our obstacle detection module detect objects within 2 meters and a combined 50-60 degree field of view. In order to verify this, we took a protractor and compass to measure and draw out an angle of 50 degrees which we extended out using measuring tape. We then positioned objects including chairs and tables between the 45-50 degree mark which our sensors were able to detect. We also placed these objects at various distances away from our sensors and measured the minimum height at which the object must be to be detected. As depicted in Figure 12, the further away an object was located from the sensors the taller it would have to be in order to be detected.

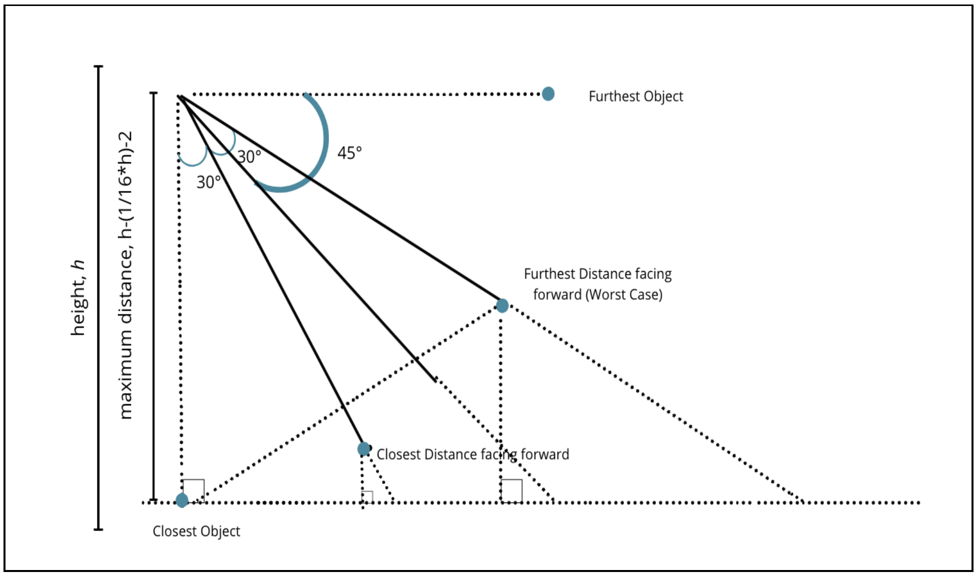

Figure 12: Measurements involving best (closest), average, and worst case distances and heights

In our tolerance analysis, we calculated best, worst, and average case scenarios as reference points we could use in our testing as shown in Figure 12 and Table 1.

	Best	**Worst**	**Average**
Height	0.16 ft (2 inches)	2 ft	1.5 ft
Distance	0 ft	4.6 ft	2.6 ft

Table 1: Calculations involving best (closest), average (provided user is directly facing forward), and worst case distances and heights

As shown in Figure 13, our best and worst case scenario were on par with our calculations, but our average case scenario was even better than expected, with an object about 2.6 feet away needing to have a minimum height of 1 foot, thereby satisfying our field of view requirements. Another requirement we had for this module was the use of an accelerometer to trigger the module only when the user was in motion. Unfortunately, due to delivery delays with our initial PCBs and soldering issues we were unable to implement our accelerometer with this module.

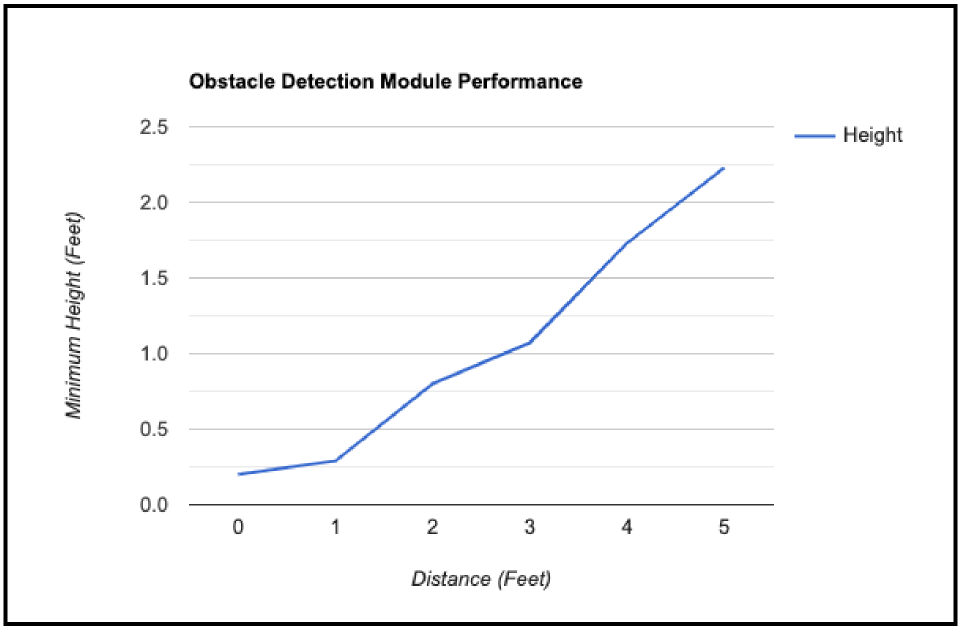

Figure 13: Graph showing obstacle detection module performance based on distance away from user

3.4. User Interface Module

The verification for the user interface module consists of ensuring that the reading module begins upon a button press of up to 3 seconds, as well as clear and audible audio is outputted from our headphone jack. In order to notify the user that the camera has turned on and is about to take an image, the message “beginning reading module” is read aloud to the user. While verifying this, our startup message was outputted and image processing began after a button press of less than 3 seconds, therefore satisfying both requirements.

4. Costs

4.1. Parts

Total: $165.49

Part	Manufacturer	Retail Cost ($)	Quantity Purchased	Total Cost ($)
Li-ion Charger	Adafruit	$5.95	1	$5.95
Li-on Battery	Adafruit	$19.95	1	$19.95
Boost Converter	Adafruit	$19.95	1	$19.95
Raspberry Pi 4 B, 4GB	Raspberry Pi	$55.00	1	$55.00
Camera	Arducam	$9.99	1	$9.99
Ultrasonic Sensors	Sparkfun	$3.95	2	$7.90
Microcontroller	Mouser Electronics	$2.29	1	$2.29
Accelerometer	Adafruit	$5.95	1	$5.95
Push Button	Sparkfun	$0.55	1	$0.55
Capacitor Set	Amazon	$8.70	1	$8.70
Voltage Regulator	Amazon	$5.89	5	$5.89
USB Connector	Amazon	$5.79	6	$5.79
Crystal Oscillator	Amazon	$6.59	1	$6.59
Ribbon Extender	Amazon	$10.99	1	$10.99

Table 2: Costs of each part

4.2. Labor

Total: $45,000

Name	Hourly Rate	Hours Invested	Total

Annamika	$40	150	$15,000
Sahil	$40	150	$15,000
Veda	$40	150	$15,000

Table 3: Costs of labor

5. Conclusion

5.1 Accomplishments

We believe that our project was successful in terms of providing the user with the ability to read a piece of text, as well as notifying them of nearby obstacles. We were able to achieve at least 85 to 95% accuracy in almost all of our trials for our reading module. Furthermore, we were able to clearly relay the text to the user through our audio system, using either wired headphones or through a bluetooth speaker. Finally, we were also able to detect obstacles near the user within a distance of 2 meters. Depending on which sensor got a signal returned, we were able to determine on which side of the user the obstacle was on. This way, the user could be notified not only of how far the obstacle was, but also where it was located relative to their position.

5.2 Uncertainties

The main uncertainties of our project stemmed from our PCB and system integration. While we were able to get our PCB working during testing, soldering issues and the damage of our microcontroller, a key component of the obstacle detection module, just before our final demonstration did not make it possible for us to integrate our PCB for the demonstration. As a result of this, our final product was not as compact as our originally envisioned physical design, making mobility more difficult. In addition to this we realized the positioning of our camera, which was on the right side of our glasses, was not ideal as the user would need to position text they wanted to read, to the right of them instead of directly in front of them. This process would have been a lot simpler if our camera was instead located right in the center of our glasses.

5.3 Ethical Considerations

The ethics behind our product design and the safe usage of our final product are of great importance to us. Throughout the course of our project we were determined to uphold and apply IEEE's Code of Ethics. Most importantly, as outlined in IEEE's fifth Code of Ethics, it is our utmost responsibility to be transparent and realistic in terms of the capabilities of our product [2]. As explained in our design verifications, we can affirm that our product does satisfy our high level requirements. However, we urge our users with severe visual impairments to use other assistive navigational devices such as a white cane in conjunction with our solution to further ensure their safety. We also honor the privacy of our users and collect no other data besides text related data through our camera.

5.4 Future Work

We feel that there is a lot of opportunity for future implementations of our product. One additional feature we could include is a supplementary lighting source. While testing our product, we realized that lighting and shadows greatly impacted the results of our reading module. With better lighting, our reading module tended to be very accurate, while with dimmer lighting or shadows, the accuracy tended to be lower. Attaching a small light source next to the camera could be a potential solution to this, allowing for clearer images without shadows and hence even more accurate reading.

Another potential addition to our project could be making our components more compact. Currently, we have a separate compartment that houses our battery, PCB, and Raspberry Pi. Ideally we would like these components to be a part of the glasses themselves while also keeping in mind the comfort of the user and weight of the components on the user's head.

Finally, we would also like to create a mobile application for our users to make the experience more user friendly and convenient. Instead of having to press a push button on the external compartment, they would simply be able to press a button in our application on their phone, which would then activate either the reading module or obstacle detection module.

References

[1] World Health Organization, Global Data on Visual Impairment, [Online], Available: https://www.who.int/blindness/publications/globaldata/en/
[2] ieee.org, IEEE Code of Ethics, [Online], Available: https://www.ieee.org/about/corporate/governance/p7-8.html
[3] ElectronicWings, Ultrasonic Module HC-SR04, [Online], Available: https://www.electronicwings.com/sensors-modules/ultrasonic-module-hc-sr04
[4] Sparkfun, HC-SR04 Ultrasonic Sensor Datasheet, [Online], Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
[5] Towards Data Science, Google's Tesseract OCR:, [Online], Available: https://towardsdatascience.com/googles-tesseract-ocr-how-good-is-it-on-documents-d71d4bf764

Appendix A: Requirements and Verification Tables

Requirements and verifications for each module are shown below.

Requirements	Verification	Verification Status
Power Module 1. Design must operate for 4+ hours without charging.	**Power Module** 1. Verification process for Item 1: a. Fully charge the battery and connect it to the Raspberry Pi through the boost converter. b. Connect the USB output of the Raspberry Pi to the power input of the PCB. c. Start the OCR computation and the obstacle detection module, and record the time in the lab notebook. d. Wait for the battery to die out and record the time again. e. Compare the maximum duration to the requirement and present the result as a numeric value in the final report.	Yes
3.7V DC Li-Ion Battery 1. Battery must output voltage in the range of 3.7V +/- 0.2V at 1A.	**3.7V DC Li-Ion Battery** 1. Verification process for Item 1: a. Fully charge the battery with the Li-ion charger. a. Test the output of the battery using an oscilloscope. b. Record the voltage obtained in the lab notebook. c. Compare the recorded voltage to the required range and present it as a numerical value in the final report.	Yes
Raspberry Pi 1. Raspberry Pi must output 3.3V +/- 0.2 V% to the obstacle detection module for a current load up to 100mA.	**Raspberry Pi** 1. Verification process for Item 1: a. Fully charge the battery and connect it to the Raspberry Pi through the boost converter. b. Connect the USB output of the Raspberry Pi to the power input of the PCB. c. Use a voltmeter to test the output of the voltage regulator of the obstacle detection module. d. Record the voltages obtained by the voltmeter in the lab notebook. e. Compare the recorded voltages to the required ranges and	Yes

	present the results as numerical values in the final report.	
Reading Module 1. The reading module must detect unobstructed text within a distance of <= 30 cm (1 foot).	**Reading Module** 1. Verification process for Item 1: a. Connect the power supply to the reading module. b. Place a page of printed text within 30 cm directly ahead of the camera. c. Plug in earphones into the audio jack of the Raspberry Pi and press the push button to activate the OCR. d. Record in the lab notebook whether the module detected text and output some audio attempting to read it out. e. Present the result as a True/False (boolean) check in the final report.	Yes
Reading Module 1. The reading module must read out unobstructed text within a distance of 30 cm (1 foot) with an accuracy of >= 85%.	**Reading Module** 1. Verification process for Item 1: a. Connect the power supply to the reading module. b. Place a page of printed text of 100 words within 30 cm directly ahead of the camera. c. Plug in earphones into the audio jack of the Raspberry Pi and press the push button to activate the OCR. d. Record in the lab notebook the original text, and the text read out by the algorithm. e. Compare the original and final text by calculating how many words were accurately read out and present the result as a percentage in the final report.	Yes
Reading Module 1. The reading module must be able to process a page of text and send the result to the audio system in <= 3 minutes.	**Reading Module** 1. Verification process for Item 1: a. Connect the power supply to the reading module. b. Place a page of a book within 30 cm of the camera. c. Plug in earphones into the audio jack of the Raspberry Pi, record the time in the lab notebook, and press the push button to activate the OCR. d. Wait for the OCR to finish and record the time in the lab notebook again. e. Calculate the time taken for the algorithm to complete and present the result as a numerical value in the final report.	Yes
Obstacle Detection Module 1. Accelerometer must detect if	**Obstacle Detection Module** 1. Verification Process for Item 1 a. Connect accelerometer to microcontroller and power supply.	No

the user is in motion, i.e. if the linear acceleration is greater than 1 m/s^2, and start the obstacle detection process.	b. Begin walking with components. c. Measure change in acceleration to check if movement has been detected from a stationary position.	
Obstacle Detection Module 1. Ultrasonic sensors should detect objects within <= 2 meters and a combined 50 - 60 degree field of view.	**Obstacle Detection Module** 1. Verification Process for Item 1 a. Use a protractor and compass to measure out a 30 degree angle. b. Use measuring tape to measure out 2 meters. c. Place an object at one corner of measured angle and distance. d. Start moving to trigger the obstacle detection module. e. Move the object around the detection range of the sensors and observe the output from the audio system. f. Record the maximum distance and angles that the sensors detected in the lab notebook. g. Present the results in the final report as the maximum range measured.	Yes
User Interface Module 1. The push button must activate the reading module in <= 3s of being pressed.	**User Interface Module** 1. Verification Process for Item 1 a. Program the Raspberry Pi to output a log of the current time as it is starting OCR computation. b. Record the time in the lab notebook as you are pressing the button. c. Listen to the output from the audio jack and record the time output by the Raspberry Pi in the lab notebook. d. Compare the two, record the duration in the lab notebook, and present it as a numerical value in the final report.	Yes
User Interface Module 1. Audio outputted from the headphone jack must be clear and audible i.e. >= 30 dB in intensity.	**User Interface Module** 1. Verification Process for Item 1 a. Start the reading module and obstacle detection module. b. Ensure audio feedback from both is loud enough to hear by the user. c. Record the result in the lab notebook and the final report.	Yes