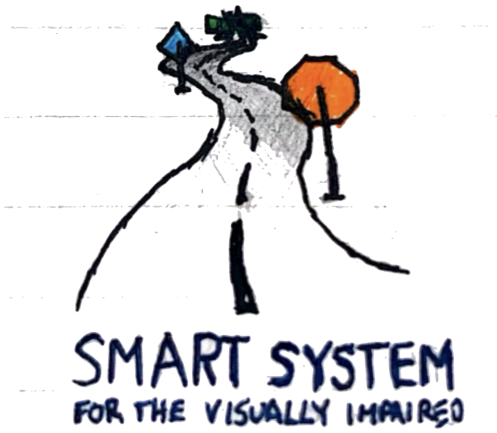
Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

Smart System for Visually Impaired



by

Team 29
Smart Bears

Team Members

Lukas Chin **lchin10@bu.edu**
Jake Lee **jaketlee@bu.edu**
Shamir Legaspi **slegaspi@bu.edu**
David Li **dav@bu.edu**
Jason Li **jli3469@bu.edu**

Submitted: 04/18/2025

Smart System for Visually Impaired

Table of Contents

Executive Summary

Smart System for Visually Impaired
Team 29

In the United States, there are over 20 million people who are visually impaired, with about 1 million of those with blindness. Every day these visually impaired individuals have trouble navigating their surroundings with many needing to use white canes or guide dogs for assistance. While white canes and guide dogs provide great aid for everyday navigation, our Smart System for Visually Impaired offers even greater enhanced mobility through wearable technology that combines obstacle detection, real time object recognition, live haptic and auditory feedback, and speech control. Smart System for Visually Impaired uses cutting edge computer vision techniques to detect everyday hazards, and it listens to the user for quick and automatic speech input. Our Smart System for Visually Impaired combines a Smart Cap and an Intelligent Wrist-Wearable wirelessly to provide the user with exceptional environmental awareness and navigation confidence.

1 Introduction

Smart System for Visually Impaired was created to address the difficulties users may encounter with existing aids for those with visual impairments. White canes and guide dogs provide visually impaired individuals a great deal of autonomy, but may be difficult to use to the fullest extent in unfamiliar environments. By providing a holistic overview of the environment and highlighting key safety and risk features, the Smart System for Visually Impaired can work in tandem with existing aids to grant further autonomy.

The main function of the project is to transform visual information into auditory and haptic information so that it would be consumable for visually impaired users. The choice of using auditory and haptic media guided many of the decisions and constraints of the design from the user interface to the latency requirements.

Overall, the project combines *IoT*, *computer vision*, and *LLM prompt engineering* into a user-friendly interface that helps users learn about their surroundings while also making them more visible to others.

The IoT aspect of the system allows for the smart cap and wrist wearable to seamlessly interact. A single button press from the user allows a pipeline of actions to be set off across both devices involved. Part of this pipeline includes capturing images and processing them through computer vision. Using *OpenCV* and its built-in *Stereo Block Mapping*, the system is able to obtain relative depth information. With raw images and relative depth mapping, the next step in the pipeline is the LLM prompting. The emphasis on natural language in large language models makes it possible to extract the most relevant information from the inputs given where defining relevance algorithmically may be difficult.

Given a stable connection, the processing portion of the system’s workflow has been demonstrated to take under 3 seconds on average from button press to the start of audio and haptic information being transmitted to the user. Because it takes much longer to output and consume audio and haptic information compared to visual information, reducing the processing time provides diminishing returns. This is also a key reason why the system is a supplement to existing aids and not a replacement. There is simply too much information to be conveyed through the media of choice.

The resulting system is easily equipped and beginner-friendly. The complex processing involved in the system is hidden from the user behind a simple button push. The combination of haptic and audio information provides the user with quick, simple information and also slower, more in-depth information.

2 System Overview and Installation

2.1 Overview block diagram

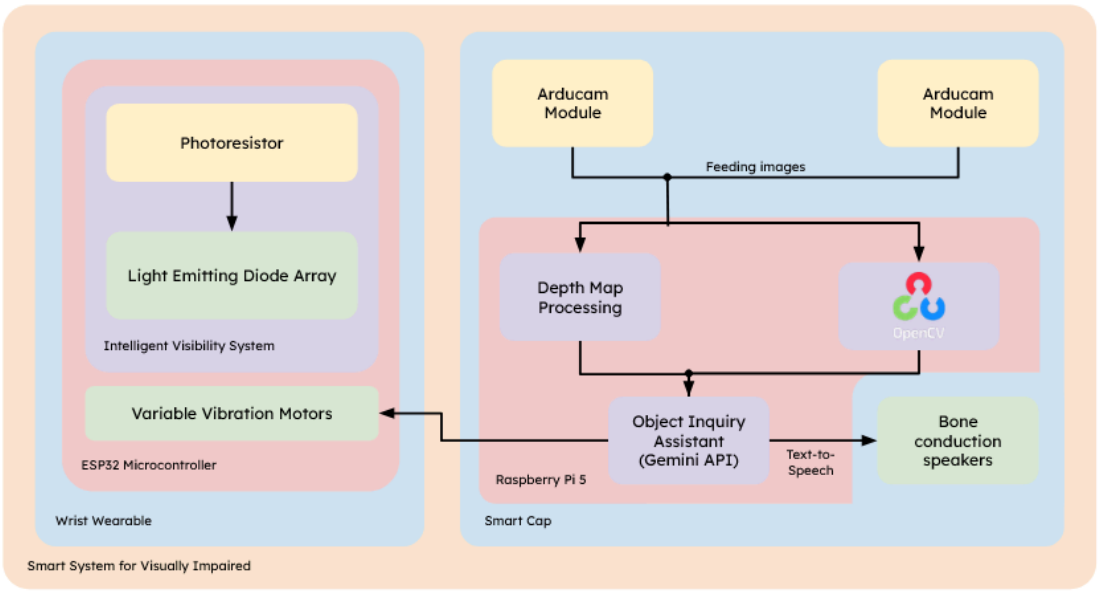


Figure 1. Flow of Data and Processing

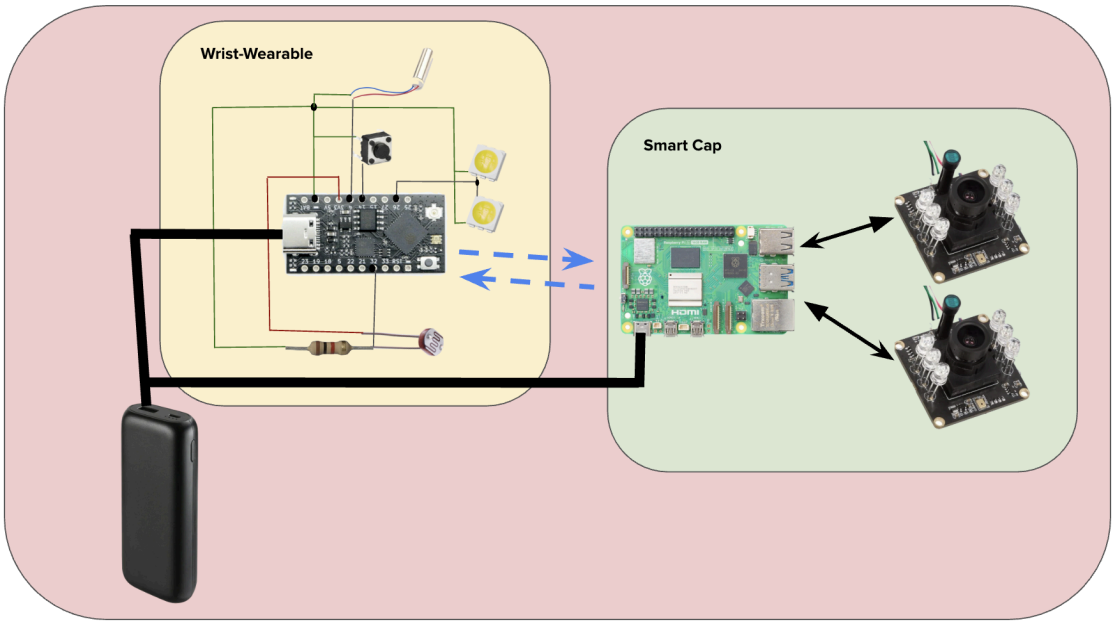
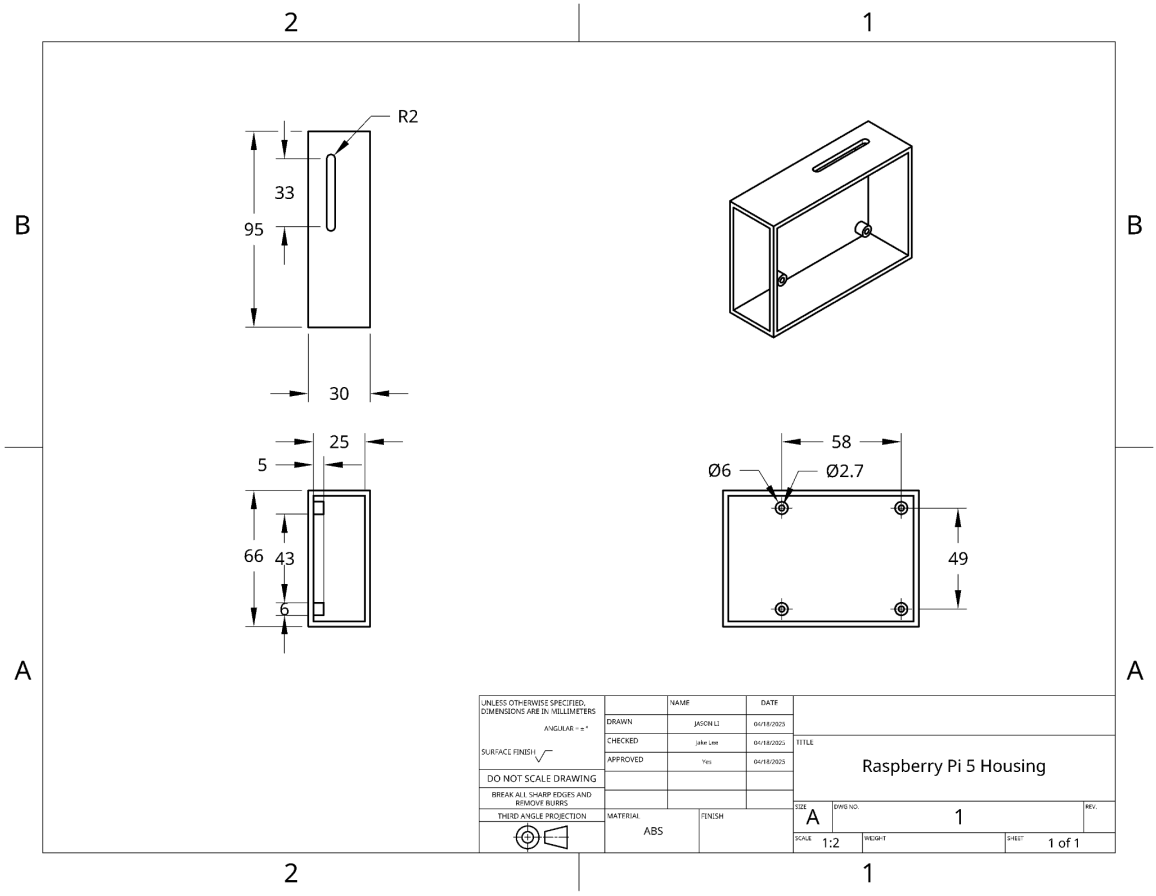


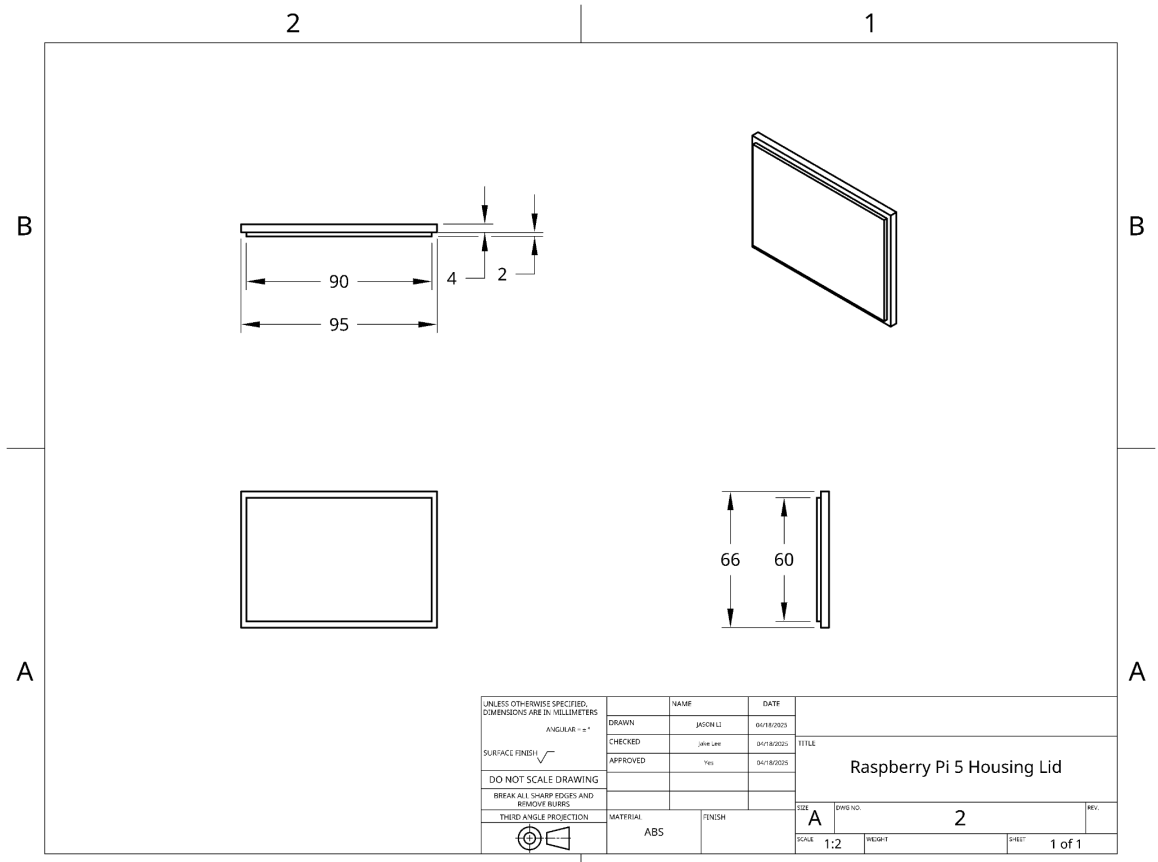
Figure 2. Smart Cap and Wrist Wearable Block Diagram

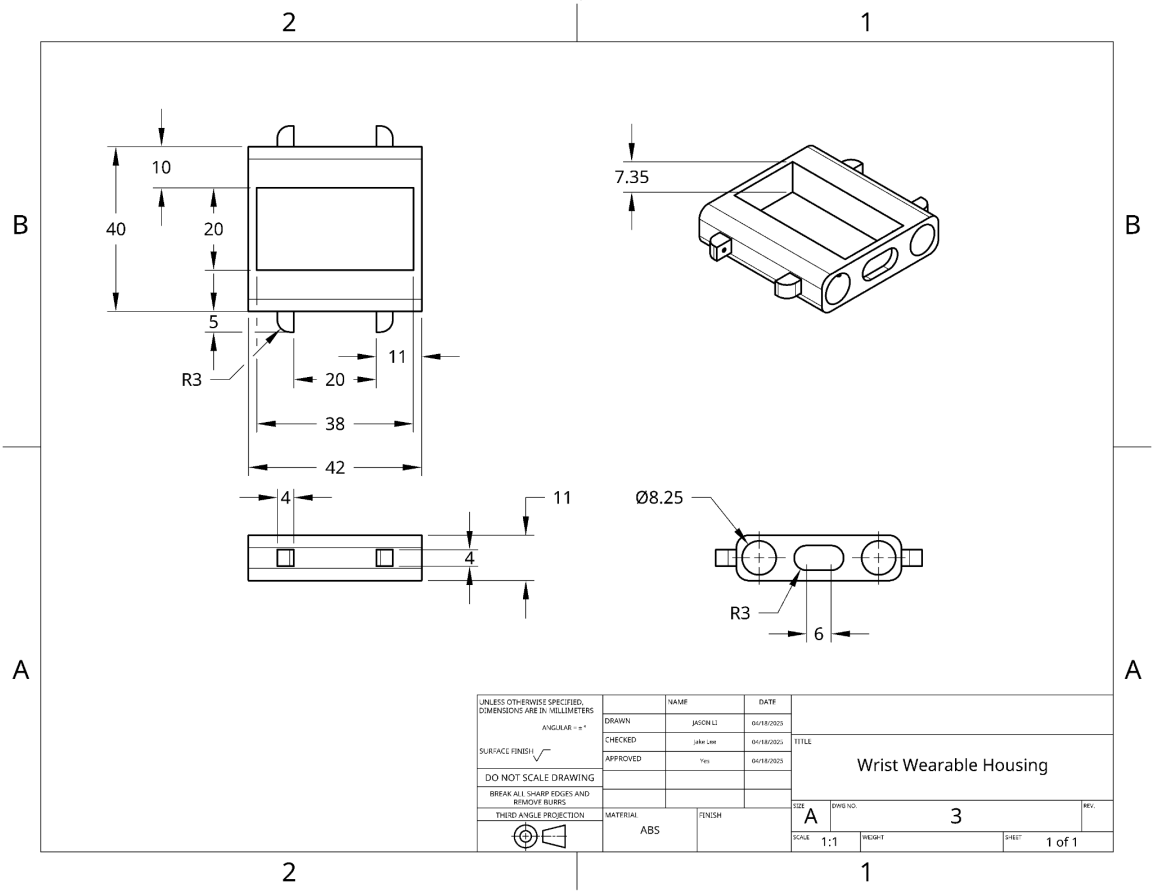
2.2 User interface.

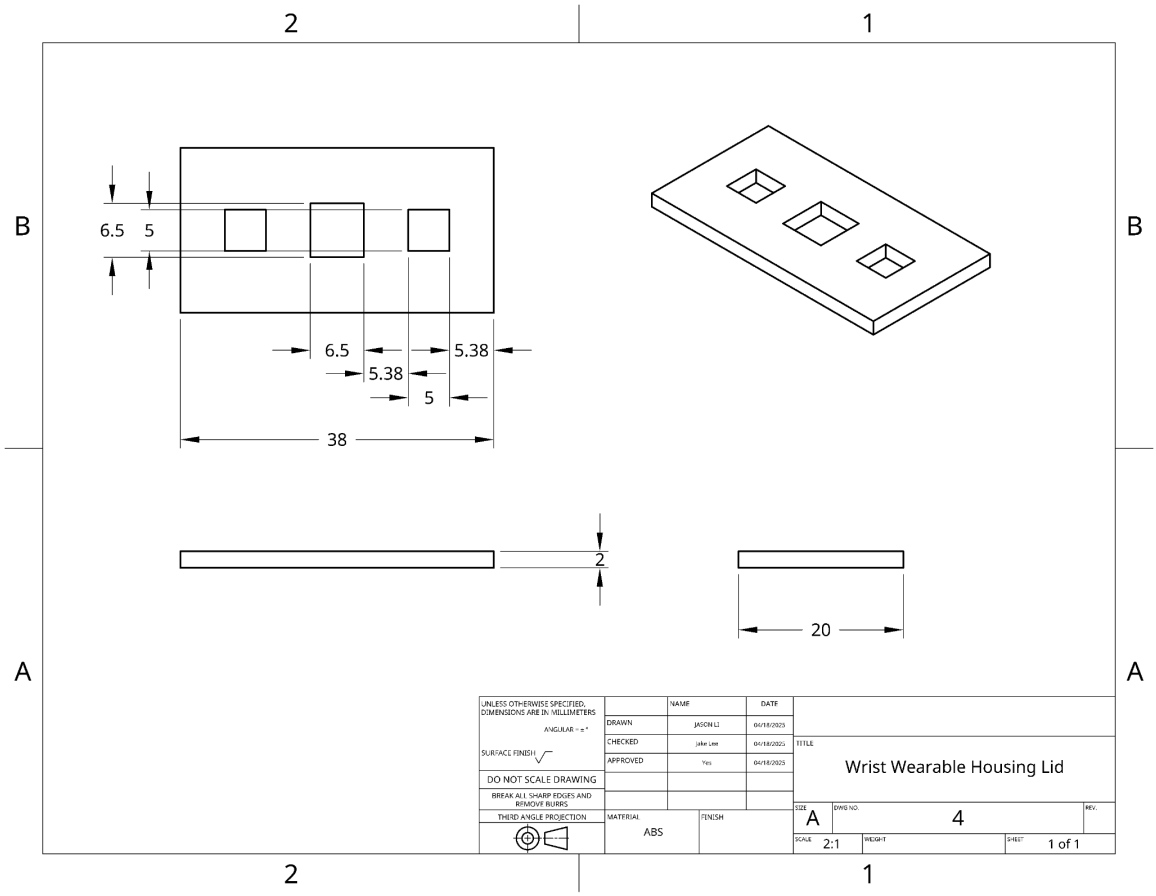
The user interacts primarily through the Wrist-Wearable, which features a central push button used to trigger environmental queries. Feedback is returned via two modalities: a bone conduction headset for spoken descriptions, and vibration motors embedded in the wrist module to signal proximity-based urgency. A light sensor regulates the IVS: in low-light conditions, white LEDs on the wrist activate automatically, improving visibility without causing glare.

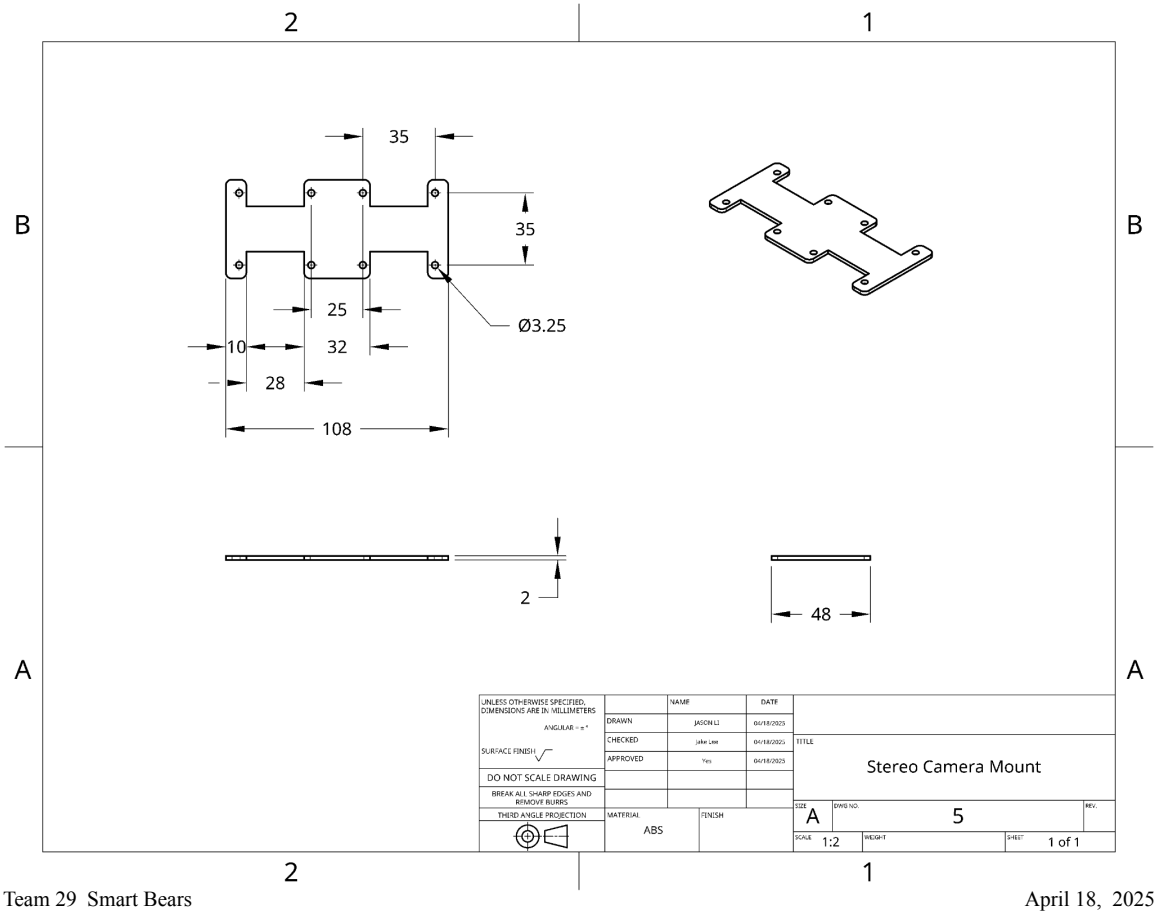
2.3 Physical drawings.




2
1
B
A
90
95
4
2
66
60
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS
ANGULAR = ± °
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
NAME
DATE
DRAWN
JASON LI
04/18/2025
CHECKED
Jake Lee
04/18/2025
APPROVED
Yes
04/18/2025
MATERIAL
ABS
FINISH
TITLE
Raspberry Pi 5 Housing Lid
SIZE
A
DWG NO.
2
REV.
SCALE
1:2
WEIGHT
SHEET
1 of 1


2
1
B
A
10
40
20
5
R3
11
20
38
42
7.35
4
11
4
Ø8.25
R3
6
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS
ANGULAR = ± °
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
NAME
DATE
DRAWN
JASON LI
04/18/2025
CHECKED
Jake Lee
04/18/2025
APPROVED
Yes
04/18/2025
MATERIAL
ABS
FINISH
TITLE
Wrist Wearable Housing
SIZE
A
DWG NO.
3
REV.
SCALE
1:1
WEIGHT
SHEET
1 of 1


2
1
B
A
6.5
5
6.5
5.38
5.38
5
38
2
20
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS
ANGULAR = ± °
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
NAME
DATE
DRAWN
JASON LI
04/18/2025
CHECKED
Jake Lee
04/18/2025
APPROVED
Yes
04/18/2025
MATERIAL
ABS
FINISH
TITLE
Wrist Wearable Housing Lid
SIZE
A
DWG NO.
4
REV.
SCALE
2:1
WEIGHT
SHEET
1 of 1


2
1
B
A
35
35
25
Ø3.25
10
32
28
108
2
48
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS
ANGULAR = ± °
SURFACE FINISH
DO NOT SCALE DRAWING
BREAK ALL SHARP EDGES AND REMOVE BURRS
THIRD ANGLE PROJECTION
NAME
DATE
DRAWN
JASON LI
04/18/2025
CHECKED
Jake Lee
04/18/2025
APPROVED
Yes
04/18/2025
MATERIAL
ABS
FINISH
TITLE
Stereo Camera Mount
SIZE
A
DWG NO.
5
REV.
SCALE
1:2
WEIGHT
SHEET
1 of 1

2.4 Installation, setup, and support

Hardware Setup:

1. Mount the stereo cameras on the Smart Cap and connect to the Raspberry Pi's USB ports.
2. Secure the Raspberry Pi in a weather-protected case and power it using a 5V portable battery.
3. Strap the Wrist-Wearable to the user's wrist, ensuring the photoresistor is unobstructed and the push button is easily accessible.
4. Connect both the ESP32 and Raspberry Pi to the same WiFi network.
5. Pair the bone conduction headset to the Raspberry Pi via Bluetooth.

Software Setup:

1. SSH into the Raspberry Pi or use a connected monitor and keyboard.
2. Navigate to the project root directory on the Pi.
3. Run the following scripts in order:
    - pi.py: Calls submodules to conduct all involved processes. Waits for a request from the UDP socket to begin image processing workflow. Captures stereo images and uses pregenerated camera calibration parameters to generate a depth map. Sends the visual processing output to an LLM and outputs the response through bluetooth speakers and initiates a buzzing signal to the wrist-wearable.
4. Ensure the ESP32 is flashed with the wrist-wearable firmware, and has open UDP ports.
    - esp.py: Handles the UDP communication for the buzzer and LEDs
5. Press the Wrist-Wearable button to initiate the query-response loop. Confirm system activation by verifying auditory and haptic feedback.

For technical support, users should review setup logs, ensure UDP port accessibility between devices, confirm Bluetooth pairing, and consult debug outputs in each script's terminal window.

3 Operation of the Project

3.1 Operating Mode 1: Normal Operation

During normal operation, the user points their head in a desired direction and presses the wrist button. The cameras capture synchronized images, which are processed by the Raspberry Pi to generate depth maps using OpenCV. These images and analysis are sent to an LLM through the Gemini API, which generates descriptive speech delivered through the bone conduction headset. If nearby obstacles are identified, the Wrist-Wearable vibrates accordingly. The IVS will activate the LED array in dim environments to assist with visibility.

3.2 Operating Mode 2: Abnormal Operations

Due to the system running via Bluetooth and WiFi, there are occurrences where the communication between the Smart Cap and the Wrist-Wearable is lost. In these cases, the system may temporarily fail to transmit signals from the wrist button to the Raspberry Pi or may drop the response signal returning from the Smart Cap. This can result in unregistered user input or a lack of audio and haptic feedback. To recover from such a state, users should first ensure that both devices are still connected to the same WiFi network. If needed, the user can restart the ESP32 microcontroller or reboot the Raspberry Pi to reinitialize the UDP socket connection.

In the case that the system involves failure in Bluetooth communication with the bone conduction headset. If the headset does not deliver speech output following an object inquiry, users should check the Bluetooth pairing status and recharge the headset if the battery is low. Re-pairing the headset through the Raspberry Pi’s Bluetooth settings typically resolves this issue.

Currently there are no hardware reset buttons, so user intervention to manually ensure that the system is connected is required. In the event of persistent issues, users are advised to go back to the setup and ensure everything is running properly.

3.3 Safety Issues

With any wearable device there are safety concerns. Since we have two stereo mounted cameras connected to our Raspberry Pi via USB there may be choking hazards if the wires get tangled. Additionally, if the cameras overheat it may cause discomfort to the user’s forehead. One of our design choices for the Raspberry Pi was to move it away from the user's head into our hip pack. This would reduce the risk of overheating from the microcontroller and take away weight from the head mount also reducing the risk of neck strain.

As for the wrist wearable, the main implications are overheating and wrist discomfort. The intensity of the vibration of the haptic feedback may be too

overwhelming for some users. Also, it could reduce stability and mobility of the hand due to the rapid vibrations. The intelligent visibility system poses a risk as it could cause discomfort to the users' eyes and people in the surroundings. The wrist strap also brings up another point of concern as the elastic nylon could rub against the users' skin and lead to rash, as well as if the strap is too tight could be a point of discomfort.

In addition, the suggestions from the smart system should be heeded with the user's discretion. Our system gives descriptions and advice, though the correct course of action may change between when the system makes a decision and when the user takes action. At any point, the user's greater judgement should be taken when making any navigational decisions.

4 Technical Background

The foundation of the Smart System lies in its fusion of stereo vision, real-time object detection, and natural language understanding. The stereo depth sensing leverages two camera inputs to simulate human binocular vision. The disparity between the images is calculated using OpenCV’s block matching algorithm to produce a gradient-based heatmap that conveys object distance in visual form.

Many parameters had to be tuned to improve the accuracy of the disparity mapping. Stereo-based depth mapping relies on block-matching algorithms that compare pixel blocks between left and right images to establish correspondences and calculate disparity. The primary challenge occurs in regions with low texture variation, where the algorithm struggles to find unique matches, resulting in sparse depth information that often outlines objects well but leaves their interiors empty. This happens because the StereoBM algorithm employs a texture threshold parameter which filters out homogeneous regions where establishing reliable correspondences is difficult. At object boundaries, additional complications arise from what's known as "speckle noise" - artifacts created when matching windows capture both foreground and background elements simultaneously, causing local regions of inconsistent disparity values

The technical solution involves careful tuning of several interconnected parameters. The texture threshold directly controls how aggressively low-texture regions are filtered, with higher values creating more gaps but reducing noise. Block size determines the area used for matching; larger blocks improve stability in uniform regions but sacrifice detail. Speckle filtering parameters (speckleWindowSize and speckleRange) help eliminate noise by defining connected component constraints - the window size determines how large a region must be to avoid being filtered out, while the range establishes how much disparity variation is acceptable within that region. Additional parameters like uniqueness ratio, pre-filter size, and pre-filter cap also affect the algorithm's ability to balance between noise reduction and information preservation. Effectively addressing depth mapping gaps requires finding the optimal balance between these parameters to ensure sufficient texture sensitivity while maintaining robustness against noise and false matches.

For object recognition, the system relies on the multimodal input capabilities of the Gemini API. The raw images are passed along with depth information and optional speech input to an LLM via a lightweight Python client which manages Gemini API

interactions. To ensure that the LLM responses align with what is considered useful, thorough prompt engineering was conducted. To minimize the processing and token costs for the Gemini API calls, the images taken are sized down before being passed. Gemini separates images into 512 x 512 pixel chunks, so reducing the image size to something that can fit into a 1 x 2 grid of 512 x 512 pixel chunks maintains a balance between detail and cost. Performing this size reduction after performing depth mapping allows for the depth mapping to be very accurate.

A system prompt is passed along with every call to the API. This prompt highlights to the LLM what it's role is and what its response format should be. In addition, the system prompt also tells the LLM how to interpret the depth map passed to it. It also dictates what the LLM should consider as most relevant when responding. We determined that distance, object type, proximity to the user's path, and surrounding context were all necessary for determining what should be included in the response, and the system prompt highlights these findings to the LLM.

After receiving the response, text-to-speech feedback is generated using system-level utilities like pyttsx3 or custom audio synthesis pipelines, which are then streamed via Bluetooth to the headset.

Communication between the Wrist-Wearable and the Smart Cap is established using UDP sockets for minimal latency and maximum reliability in a wireless setting. All modules are designed to run concurrently, ensuring smooth end-to-end interactions even under constrained computational resources.

The Wrist-Wearable makes use of GPIO control and PWM to control the input of the button and photoresistor and the output of the LEDs and the buzzer. The photoresistor is paired with its matching 10k Ohm resistor along an analog pin, so that the voltage drop can be read to understand the brightness of the surroundings.

5 Relevant Engineering Standards

The Smart System for Visually Impaired incorporates various engineering standards to ensure safety, reliability, and accessibility. These standards guide the development of both hardware and software components, providing a framework that supports the system's functionality while meeting industry requirements.

The wearable components of our system adhere to IEEE 360-2022, which outlines requirements for wearable consumer electronic devices. This standard ensures our devices meet basic safety, comfort, and wearability criteria through clear technical guidelines and test methods. To address general electrical safety, we follow IEC/UL 62368-1, which applies to audio/video and information technology equipment. This standard supports the safety of our head-mounted and wrist-worn electronic components during both normal operation and fault conditions.

While our system uses a standard commercial-grade portable battery to power the wearable and head-mounted components, we still followed best practices and relevant standards to ensure safety. We follow UL 1642 and IEC 62133-2, which define requirements for lithium batteries in consumer electronics. These standards cover aspects such as overcurrent protection, thermal regulation, and failure containment.

Additionally, our team adhered to course safety policies for lithium battery usage as outlined by the course faculty. While our battery posed significantly less risk than raw lithium cells, we observed the following:

- Battery terminals were not exposed; no alligator clips were used.
- No charging occurred without pre-approved charging setups.
- Commercial battery packs were only charged using manufacturer-provided chargers.
- Batteries were never left unattended during charging.

Our system employs a UDP-based communication protocol over WiFi, enabling real-time data exchange between the TinyPico ESP32 in the wrist wearable and the Raspberry Pi 5 in the head-mounted unit. UDP allows for low-latency performance, which is critical for time-sensitive haptic feedback. Our stereoscopic vision algorithm adheres to established methodologies for generating depth maps. Both implementations are based on current industry best practices in computer vision and embedded AI.

To ensure reliable and secure communication when sending requests to the Gemini API, our system follows established API development and integration best practices. This includes implementing rate limiting, handling timeouts and retries, and validating input/output

payloads to prevent malformed requests. Authentication is securely managed using appropriate tokens, and all requests are made over HTTPS to ensure data confidentiality and integrity. These practices ensure robust integration and minimize the risk of communication failures or security vulnerabilities during real-time operation. While the current prototype does not retain any user data, our architecture is built with potential future data logging in mind. In such cases, the system is designed to comply with appropriate data protection regulations and privacy best practices to safeguard sensitive information.

Designed specifically for visually impaired users, our system aligns with the Assistive Technology Act and other accessibility guidelines to ensure usability and inclusion. We considered research-based best practices for mobility aids in urban environments during the design of our wearable devices. Interface elements such as button input, haptic feedback, and audio cues follow established accessibility standards. Haptic feedback is designed with distinct vibration patterns based on tactile interface standards to enhance user differentiation and interaction without requiring vision.

6 Cost Breakdown

Project Costs for Production of Beta Version (Next Unit after Prototype)				
Item	Quantity	Description	Unit Cost	Extended Cost
1	1	Raspberry Pi 5 w/ 16 GB of RAM	$159.99	$159.99
2	1	TinyPICO ESP32	$21.95	$21.95
3	1	Bone Conduction Headphones	$17.88	$17.88
4	1	DC Coreless Vibration Motor	$8.99	$8.99
5	2	Arducam 1080P USB Camera	$34.99	$69.98
6	1	100 count 5mm x 5mm LED Lights	$4.99	$4.99
7	1	Braided Nylon Watch Band	$11.99	$11.99
8	1	10 Count 10k Ohm Photoresistor	$7.90	$7.90
9	1	100 Count Push Buttons	$5.39	$5.39
10	1	Power Supply (5V, 4.5 A)	$26.98	$26.98
11	1	3d Printing Filament	$3.00	$3.00
12	1	Fanny Pack Holster	$8.95	$8.95
13	1	Jumper Wires Pack	$2.00	$2.00
			Beta Version-Total Cost	$349.99

The main expense for the beta version of the product is the Raspberry Pi 5 w/ 16 GB of RAM, accounting for over 45% of the total cost. The need for such a powerful device can be attributed to the need for quick onboard computer vision processing. However, there is still a need to make an external API call within the workflow. When scaling to production, it may make sense to downgrade the onboard hardware in favor of hosting an external cloud solution for processing.

All software and API calls used were free, open source, or provided free services with rate limiting. This limits the cost of the product to just the hardware components. In production, these costs will be further reduced due to the ability to buy in bulk.

7 Appendices

7.1 Appendix A - Specifications

Core Components

- Dual 2MP Arducam 1080P Day/Night Vision USB Cameras (stereoscopic vision for depth mapping)
- TinyPico ESP32 Microcontroller Board
- Raspberry Pi 5 (16GB RAM)
- Bone Conduction Bluetooth Powered Headset (Bluetooth 5.3, open ear construction)
- Wrist-Wearable Module with:
    - 7x25mm Micro Vibrating Motor (1–6V, 8000–20000 RPM)
    - 2x 5x5mm White LED Diodes (6000K)
    - 6x6mm White Push Button
    - LDR Photoresistor (10kΩ)
- INIU Portable Charger (10000 mAh)
- Router (for WiFi communication)

Battery Life

- Bone conduction Bluetooth headset: 6–10 hours of use per charge, 2-hour charging time1
- Raspberry Pi and TinyPico: 10 hours battery life estimated

Connectivity

- Bluetooth 5.3 (headset)
- WiFi (UDP communication between ESP32 and Raspberry Pi 5)
- Potential for LTE connectivity in future iterations

Performance

- Wrist-Wearable to Smart Cap signal transmission: ≥95% success rate
- Dual-camera snapshot initiation: <5 seconds from signal
- Depth mapping image processing: <1 second
- Haptic feedback system: immediate response upon signal receipt
- Speech engine response accuracy: 85–90% in quiet environments
- Intelligent Visibility System (LEDs in low light): >75% activation rate

User Interaction

- Wrist-wearable button triggers image capture and analysis
- Auditory feedback via bone conduction headset
- Haptic alerts for immediate obstacles

Physical Dimensions

- Micro vibrating motor: 7x25mm
- LED diodes: 5x5mm
- Push button: 6x6mm
- Cameras: Standard USB 2MP module size 32x32mm each
- Wrist-wearable: Adjustable strap (unisex expandable strap)

Expandability

- Future LTE upgrade possible for Raspberry Pi 5 to remove WiFi dependency

7.2 Appendix B – Team Information

Smart Bears is a dynamic team of five computer engineers—Lukas Chin, Jake T. Lee, Shamir Legaspi, David Li, and Jason Li—who first connected during their freshman year at Boston University. Their shared passion for technology and innovation quickly brought them together, leading to collaborations on a variety of personal and group projects throughout their undergraduate journey. Each member brought unique interests within computer engineering, ranging from mobile and cloud computing to artificial intelligence, data science, and hardware systems. This diversity of expertise enabled them to approach problems from multiple perspectives and develop creative, robust solutions.

Throughout their time at BU, the Smart Bears immersed themselves in a rigorous curriculum that combined foundational engineering theory with hands-on technical experience. The Department of Electrical & Computer Engineering emphasized both individual and teamwork skill-building, exposing students to real-world challenges and fostering a culture of interdisciplinary research and innovation. The team participated in research labs, internships, and university-sponsored competitions, further sharpening their technical and communication skills.

The culmination of their undergraduate experience was the Senior Design Capstone, where they leveraged everything they had learned to design and prototype an ambitious, innovative product. This project required them to collaborate closely, navigating technical challenges, managing project timelines, and engaging with real-world clients—experiences that mirrored professional engineering environments and solidified their teamwork. Their collective efforts not only showcased their technical acumen but also highlighted their ability to translate classroom knowledge into impactful engineering solutions.

Following graduation, Lukas will be continuing work on his mobile app startup. Jake will be working at a startup. Shamir will be working at an AI-based startup. David will be working at a chip-based startup. Jason Li will be working at Deloitte as a consultant.