

# SMARTNAV CANE: A Sensor-Integrated Mobility Aid for Enhanced Navigation



## Department of Electrical and Computer Engineering

### Group 12 Authors:

*Evans  
Bakire*

*Yussef  
Abdelbary*

*Deon  
McCoy*

*Brennan  
Russell*

*Sharifah  
Alselahi*

*Computer  
Engineering*

*Computer  
Engineering*

*Electrical  
Engineering*

*Electrical  
Engineering*

*Computer  
Engineering*

### Reviewers:

*Dr. Mark Maddox  
Dr. Avra Kundu  
Dr. Justin Phelps*

*ECE  
ECE  
ECE*

*Lecturer  
Laboratory Manager  
Lecturer*

### Mentor:

*Dr. Chung Yong Chan*

# Table Of Contents

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <b>Chapter 1: Executive Summary.....</b>                        | <b>10</b> |
| <b>Chapter 2: Project Description.....</b>                      | <b>11</b> |
| 2.1 Project Motivation and Goals:.....                          | 11        |
| 2.2 Current Existing Projects.....                              | 12        |
| 2.3 Basic Goals (Essential Functionalities).....                | 14        |
| 2.4 Advanced Goals (Enhanced Usability & User Experience).....  | 14        |
| 2.5 Stretch Goals (Future-Oriented Features & Scalability)..... | 15        |
| 2.6 Objectives.....                                             | 15        |
| 2.4 Requirements and Specifications.....                        | 16        |
| 2.5 House of Quality Diagram.....                               | 18        |
| 2.6 Project Block Diagram/Prototype.....                        | 18        |
| 2.7 Hardware Design.....                                        | 19        |
| 2.7.1 Block Diagram.....                                        | 19        |
| 2.8 Software Design.....                                        | 20        |
| 2.8.1 Flow Chart.....                                           | 20        |
| <b>Chapter 3: Research.....</b>                                 | <b>21</b> |
| 3.1 Technology Comparison.....                                  | 21        |
| 3.1.1 MCU.....                                                  | 21        |
| 3.1.1.1 ESP 32.....                                             | 22        |
| 3.1.1.2 Raspberry PI 4 Model B.....                             | 23        |
| 3.1.1.3 Arduino.....                                            | 24        |
| 3.1.1.4 Final MCU Comparison.....                               | 25        |
| 3.1.2 Object Detection Technology.....                          | 26        |
| 3.1.2.1 Time-of-Flight (ToF).....                               | 26        |
| 3.1.2.2 Lidar.....                                              | 28        |
| 3.1.2.3 Ultrasonic.....                                         | 29        |
| 3.1.2.4 Photoelectric.....                                      | 30        |
| 3.1.2.5 mmWave.....                                             | 31        |
| 3.1.2.6 Infrared Distance.....                                  | 32        |
| 3.1.2.7 Sensor Technology Comparison.....                       | 33        |
| 3.1.3 Object Detection Sensor Component Selection.....          | 34        |
| 3.1.3.1 Analysis of ToF Sensors for Low-Level Obstacle.....     | 34        |

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 3.1.3.2 Analysis of Infrared Distance Sensors for High-Level Obstacle Detection..... | 36 |
| 3.1.4 Global Navigation Satellite System (GNSS) Module.....                          | 37 |
| 3.1.4.1 NEO-M9N-00B.....                                                             | 37 |
| 3.1.4.2 NEO-F10N-00B.....                                                            | 37 |
| 3.1.4.3 MAX-M10S.....                                                                | 38 |
| 3.1.4.4 L86-M33.....                                                                 | 38 |
| 3.1.4.5 ZED-F9P-05B.....                                                             | 39 |
| 3.1.4.6 NEO-6M.....                                                                  | 39 |
| 3.1.4.7 CD-PA1616S.....                                                              | 39 |
| 3.1.4.8 GNSS Modules Comparisons.....                                                | 40 |
| 3.1.5 Antenna.....                                                                   | 40 |
| 3.1.5.1 ANN-MB-00.....                                                               | 42 |
| 3.1.5.2 CGGBP254.....                                                                | 42 |
| 3.1.5.2 ANT-GNCP-CA188L165.....                                                      | 43 |
| 3.1.5.3 Adafruit GPS Antenna.....                                                    | 43 |
| 3.1.5.4 Antenna Comparisons.....                                                     | 43 |
| 3.1.6 Audio Output.....                                                              | 44 |
| 3.1.6.1 Traditional Dynamic Speakers.....                                            | 44 |
| 3.1.6.2 Piezoelectric Buzzers.....                                                   | 45 |
| 3.1.6.3 Electrostatic Speakers.....                                                  | 45 |
| 3.1.6.4 Audio Output Technology Comparison.....                                      | 46 |
| 3.1.7 Dynamic Audio Output Speakers.....                                             | 46 |
| 3.1.7.1 Adafruit Mini.....                                                           | 47 |
| 3.1.7.2 Dayton Audio CE Series.....                                                  | 47 |
| 3.1.7.3 PUI Audio.....                                                               | 48 |
| 3.1.6.4 Audio Output Speakers Comparisons.....                                       | 48 |
| 3.1.7 Audio Amplification.....                                                       | 49 |
| 3.1.8 Microphones.....                                                               | 49 |
| 3.1.8.1 Digital MEMS Microphones.....                                                | 50 |
| 3.1.8.2 Analog MEMS Microphones.....                                                 | 50 |
| 3.1.8.3 Electret Condenser Microphones (ECMs).....                                   | 50 |
| 3.1.8.4 Microphone Technology Comparisons.....                                       | 51 |
| 3.1.9 Digital MEMS Microphones.....                                                  | 51 |
| 3.1.9.1 INMP441.....                                                                 | 52 |
| 3.1.9.2 SPH0645LM4H-B.....                                                           | 52 |
| 3.1.9.3 ICS-43434.....                                                               | 53 |
| 3.1.9.4 Digital MEMS Microphones Comparison.....                                     | 53 |

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| 3.1.10 Voltage Regulators.....                                           | 54d |
| 3.1.10.1 Linear Voltage Regulators.....                                  | 55  |
| 3.1.10.2 Switching Voltage Regulators.....                               | 55  |
| 3.1.10.3 Charge Pump Voltage Regulators.....                             | 56  |
| 3.1.10.4 Voltage Regulators Comparison.....                              | 57  |
| 3.1.11 Linear Voltage Regulators.....                                    | 58  |
| 3.1.11.1 AMS1117-3.3.....                                                | 58  |
| 3.1.11.2 MIC5365-3.3.....                                                | 59  |
| 3.1.11.3 LM1117-3.3.....                                                 | 59  |
| 3.1.11.4 Linear Voltage Regulators Comparisons.....                      | 60  |
| 3.1.12 DC-DC Converter.....                                              | 61  |
| 3.1.12.1 DC-DC Boost Converter Technology.....                           | 62  |
| 3.1.12.2 Step-Up Voltage Regulator Technology.....                       | 62  |
| 3.1.12.3 Boost Technology Converter Comparison Table.....                | 63  |
| 3.1.13 Step-Up Voltage Regulators.....                                   | 63  |
| 3.1.13.1 Pololu 5V Step-Up Voltage Regulator.....                        | 63  |
| 3.1.13.2 TPS61030 Step-Up Voltage Regulator.....                         | 64  |
| 3.1.13.3 TPS61200 Boost Converter.....                                   | 64  |
| 3.1.13.4 Step-Up Voltage Regulators Comparison.....                      | 64  |
| 3.1.13 Power Source.....                                                 | 65  |
| 3.1.13.1 Lithium-Ion (Li-ion) vs. Lithium-Polymer (Li-Po) Batteries..... | 66  |
| 3.1.13.2 18650 Li-ion.....                                               | 67  |
| 3.1.13.3 Flat Li-ion Pouch Battery.....                                  | 68  |
| 3.1.13.4 Power Source Comparisons.....                                   | 68  |
| 3.1.14 Communication Protocol.....                                       | 70  |
| 3.1.14.1 UART.....                                                       | 70  |
| 3.1.14.2 I2C.....                                                        | 71  |
| 3.1.14.3 I2S.....                                                        | 71  |
| 3.1.14.4 USB.....                                                        | 72  |
| 3.2 Software Selection.....                                              | 72  |
| 3.2.1 Development Environment.....                                       | 73  |
| 3.2.1.1 Arduino IDE.....                                                 | 73  |
| 3.2.1.2 PlatformIO.....                                                  | 75  |
| 3.2.1.3 ESP-IDF.....                                                     | 75  |
| 3.2.1.4 Development Environment Comparison.....                          | 75  |
| 3.2.2 Programming Language.....                                          | 76  |
| 3.2.2.1 C/C++.....                                                       | 76  |

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| 3.2.2.2 Micropython.....                                             | 77        |
| 3.2.2.3 Rust.....                                                    | 77        |
| 3.2.2.4 Programming Language Comparison.....                         | 78        |
| 3.2.3 Navigation API Integration.....                                | 79        |
| 3.2.3.1 Google Maps API.....                                         | 79        |
| 3.2.3.2 OpenStreetMap APIs.....                                      | 80        |
| 3.2.3.3 Mapbox Directions API.....                                   | 80        |
| 3.2.3.4 Directions API Comparisons.....                              | 81        |
| 3.2.4 Embedded Operating Systems.....                                | 81        |
| 3.2.4.1 FreeRTOS.....                                                | 82        |
| 3.2.4.2 Zephyr OS.....                                               | 82        |
| 3.2.4.3 RIOT OS.....                                                 | 83        |
| 3.2.4.4 RTOS Comparison.....                                         | 83        |
| 3.2.5 Cloud-Based Speech-to-Text.....                                | 83        |
| 3.2.5.1 Google Speech-to-Text API.....                               | 84        |
| 3.2.5.2 OpenAI Whisper API.....                                      | 84        |
| 3.2.5.3 Azure Speech Service.....                                    | 85        |
| 3.2.5.4 Speech-to-Text Comparison.....                               | 85        |
| 3.2.6 Schematic and PCB Design Software.....                         | 86        |
| 3.2.6.1 KiCad.....                                                   | 86        |
| 3.2.6.2 Fusion 360.....                                              | 87        |
| 3.2.6.3 Eagle.....                                                   | 87        |
| 3.2.6.4 Design Software Comparison.....                              | 87        |
| <b>Chapter 4: Standards and Design Constraints.....</b>              | <b>88</b> |
| 4.1.1 Assistive Technology Standards.....                            | 89        |
| 4.1.2 Sensor Performance and Integration Standards.....              | 90        |
| 4.1.3 Manufacturing and Quality Control Standards.....               | 91        |
| 4.1.4 Electrical and Safety Standards.....                           | 92        |
| 4.1.5 Other Standards.....                                           | 93        |
| 4.2 Constraints.....                                                 | 93        |
| 4.2.1 Time.....                                                      | 93        |
| 4.2.2 Usability.....                                                 | 94        |
| 4.2.3 Safety.....                                                    | 95        |
| 4.2.4. Economic.....                                                 | 96        |
| 4.2.7 Technical Constraints.....                                     | 98        |
| <b>Chapter 5: Comparison of Chat GPT with Similar Platforms.....</b> | <b>99</b> |
| 5.1 The Role of AI in the development of our project.....            | 99        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| 5.2 ChatGPT.....                                                               | 99         |
| 5.3 Comparison of AI tools: (ChatGPT, Google Gemini, Microsoft Copilot).....   | 100        |
| 5.4 Pros of utilizing ChatGPT.....                                             | 101        |
| 5.5.1 Troubleshooting.....                                                     | 101        |
| 5.5.2 Technical Aspect.....                                                    | 102        |
| 5.5.3 Allocation of Different Resources.....                                   | 102        |
| 5.5.4 Refined Writing.....                                                     | 103        |
| 5.5.5 Quick Responses.....                                                     | 103        |
| 5.5.6 Personalized Learning and Accessibility for visually Impaired Users..... | 103        |
| 5.5.7 Cost-Effective and Scalable Solutions.....                               | 104        |
| 5.6 Negative aspect of using ChatGPT.....                                      | 104        |
| 5.6.1 Virtual Limitation.....                                                  | 105        |
| 5.6.2 Possible Incorrect Information.....                                      | 105        |
| 5.7 Case Studies.....                                                          | 106        |
| 2. LiDAR (Light Detection and Ranging).....                                    | 106        |
| 3. Infrared (IR) Sensors.....                                                  | 106        |
| 4. Computer Vision (AI-Based Cameras).....                                     | 107        |
| 5. mmWave Radar (Millimeter-Wave Radar).....                                   | 107        |
| Low-Power Components.....                                                      | 108        |
| Microcontrollers.....                                                          | 108        |
| Sensors.....                                                                   | 108        |
| Communication.....                                                             | 108        |
| Feedback/UI.....                                                               | 108        |
| Battery Life Strategies.....                                                   | 108        |
| <b>Chapter 6: Hardware Design.....</b>                                         | <b>110</b> |
| 6.1 GNSS Module.....                                                           | 110        |
| 6.2 Power Delivery.....                                                        | 111        |
| 6.3 USB to UART Bridge.....                                                    | 114        |
| 6.4 Button Interface.....                                                      | 114        |
| 6.7 Sensors.....                                                               | 116        |
| 6.8 MIC and Speaker Schematics.....                                            | 117        |
| 6.9 ESP 32 Wroom MCU.....                                                      | 119        |
| 6.10 Overall Schematic.....                                                    | 120        |
| <b>Chapter 7: Software Design.....</b>                                         | <b>121</b> |
| 7.1 Functionalities Use Case Diagram.....                                      | 122        |
| 7.1.2 Software Architecture Class Diagram.....                                 | 124        |
| 7.2 Navigation / Object Detection System Architecture and State Flow.....      | 125        |

|                                                                        |            |
|------------------------------------------------------------------------|------------|
| 7.2.1 Navigation System.....                                           | 126        |
| 7.2.2 Object Detection System.....                                     | 127        |
| 7.4 GPS Module Initialization, Data Processing and Feedback.....       | 128        |
| 7.5 Data Transfer.....                                                 | 129        |
| 7.5.1 I2S Data Transfer.....                                           | 129        |
| 7.5.2 I2C Communication.....                                           | 130        |
| 7.5.3 ADC Data Acquisition.....                                        | 130        |
| 7.5.4 UART Communication.....                                          | 130        |
| 7.6 Task Management Hierarchy and Interrupt-Driven Responsiveness..... | 131        |
| 7.6.1 Task Prioritization.....                                         | 131        |
| 7.6.2 Interrupt Service Routines.....                                  | 132        |
| 7.7 API Integration and Protocols.....                                 | 132        |
| <b>Chapter 8: Prototype Fabrication.....</b>                           | <b>133</b> |
| 8.1 Main PCB Board Design.....                                         | 134        |
| 8.2 Rechargeable TP4056 Module Peripheral.....                         | 136        |
| 8.3 NEO GPS Module BreakOut Board.....                                 | 138        |
| 8.4 TOF VL530X BreakOut Board.....                                     | 141        |
| <b>Chapter 9: System Testing.....</b>                                  | <b>142</b> |
| 9.1 Hardware Testing.....                                              | 142        |
| 9.1.1 ESP32-WROOM-32D.....                                             | 142        |
| 9.1.2 MIC5365-3.3 Linear Voltage Regulator.....                        | 143        |
| 9.1.2 TPS61200 Step-Up Voltage Regulator.....                          | 143        |
| 9.1.3 ToF VL53L0X Object Detection Sensor.....                         | 144        |
| 9.1.4 IR Sharp GP2Y0A710K0F Object Detection Sensor.....               | 144        |
| 9.1.4 NEO-6M GPS Module.....                                           | 145        |
| 9.1.5 PUI Audio 3W 4Ω Mini Loudspeaker.....                            | 145        |
| 9.1.6 INMP441 Microphone.....                                          | 145        |
| 9.1.7 CGGB254 Antenna.....                                             | 146        |
| 9.2 Software Testing.....                                              | 146        |
| 9.2.1 Speech to Text API.....                                          | 147        |
| 9.2.3 Google Maps API.....                                             | 148        |
| 9.2.4 ESP 32 WROOM Modules.....                                        | 149        |
| 9.3 Performance Evaluation.....                                        | 150        |
| 9.4 Plan for Senior Design II (SD2).....                               | 153        |
| 9.4.1 Sensor Housing Integration.....                                  | 154        |
| 9.4.2 PCB Manufacturing and Testing.....                               | 154        |
| <b>Chapter 10: Administrative Content.....</b>                         | <b>155</b> |

|                                       |            |
|---------------------------------------|------------|
| 10.1 Project Budget.....              | 155        |
| 10.2 Total Cost.....                  | 156        |
| 10.3 Bill of Materials.....           | 157        |
| 10.4 Milestones Schedule.....         | 159        |
| 10.5 Table of Work Distributions..... | 160        |
| <b>Chapter 11: Conclusion.....</b>    | <b>162</b> |
| <b>Appendix A.....</b>                | <b>163</b> |
| References.....                       | 163        |

## List of Tables

|                                                               |    |
|---------------------------------------------------------------|----|
| <b>Table 2.1</b> Components Specifications.....               | 7  |
| <b>Table 2.1</b> System Specifications.....                   | 8  |
| <b>Table 3.1.</b> MCU Comparisons Table.....                  | 16 |
| <b>Table 3.2.</b> Sensor Technology Comparison.....           | 24 |
| <b>Table 3.3.</b> Sensor ToF Comparison.....                  | 26 |
| <b>Table 3.4.</b> Sensor Infrared Comparison.....             | 27 |
| <b>Table 3.5.</b> GNSS Modules Comparisons.....               | 31 |
| <b>Table 3.6.</b> Antenna Comparisons.....                    | 35 |
| <b>Table 3.7.</b> Audio Technology Comparison.....            | 37 |
| <b>Table 3.8.</b> Audio output Comparisons.....               | 39 |
| <b>Table 3.9.</b> Microphone Technology Comparisons.....      | 42 |
| <b>Table 3.10.</b> Microphone Digital MEMS Comparisons.....   | 44 |
| <b>Table 3.11.</b> Voltage Regulators Comparison.....         | 48 |
| <b>Table 3.12.</b> Absolute Maximum Ratings Digikey.....      | 49 |
| <b>Table 3.13.</b> Linear Voltage Regulators Comparisons..... | 52 |
| <b>Table 3.14.</b> Boost Converter Tech Comparison Table..... | 54 |
| <b>Table 3.15.</b> Step-Up Voltage Regulator Comparison.....  | 56 |

|                    |                                                 |            |
|--------------------|-------------------------------------------------|------------|
| <b>Table 3.16.</b> | <b>Battery Tech Comparison.....</b>             | <b>57</b>  |
| <b>Table 3.17.</b> | <b>Component Current and Voltage Draw.....</b>  | <b>60</b>  |
| <b>Table 3.18.</b> | <b>Li-ion Battery Comparison Table.....</b>     | <b>60</b>  |
| <b>Table 3.19.</b> | <b>Development Environment Comparisons.....</b> | <b>67</b>  |
| <b>Table 3.20.</b> | <b>Programming Language Comparison.....</b>     | <b>70</b>  |
| <b>Table 3.21.</b> | <b>Navigation API Comparison.....</b>           | <b>72</b>  |
| <b>Table 3.22.</b> | <b>RTOS Comparison.....</b>                     | <b>74</b>  |
| <b>Table 3.23.</b> | <b>Speech-to-Text Comparison.....</b>           | <b>77</b>  |
| <b>Table 3.24.</b> | <b>Design Software Comparison.....</b>          | <b>79</b>  |
| <b>Table 10.1.</b> | <b>Project Budget.....</b>                      | <b>146</b> |
| <b>Table 10.2.</b> | <b>Project BOM.....</b>                         | <b>148</b> |
| <b>Table 10.3.</b> | <b>SD1 Milestones.....</b>                      | <b>150</b> |
| <b>Table 10.2.</b> | <b>SD2 Milestones.....</b>                      | <b>151</b> |

## List of Figures

|                                                                                                 |     |
|-------------------------------------------------------------------------------------------------|-----|
| <b>Figure 2.1.</b> House of Quality.....                                                        | 9   |
| <b>Figure 2.2.</b> SmartNav Design Concept.....                                                 | 10  |
| <b>Figure 2.3.</b> Hardware Block Diagram.....                                                  | 11  |
| <b>Figure 2.4.</b> Software Design Flowchart.....                                               | 12  |
| <b>Figure 3.1.</b> Principle of an Infrared Sensor.....                                         | 19  |
| <b>Figure 3.2.</b> Principle of a LIDAR Sensor.....                                             | 20  |
| <b>Figure 3.3.</b> Principle of an Ultrasonic Sensor.....                                       | 21  |
| <b>Figure 3.4.</b> Principle of a Photoelectric Sensor.....                                     | 22  |
| <b>Figure 6.1.</b> GPS Module with Backup Battery, EEPROM, and Voltage Regulator Schematic..... | 103 |
| <b>Figure 6.2.</b> 3.3V Regulator Schematic.....                                                | 104 |
| <b>Figure 6.3.</b> 5V Step-Up Voltage Regulator Schematic.....                                  | 104 |
| <b>Figure 6.4.</b> TP4056 with External Modules Schematic.....                                  | 105 |
| <b>Figure 6.5.</b> USB to UART Bridge with External Modules Schematic.....                      | 106 |
| <b>Figure 6.6.</b> Buttons and resistors for UI commands Schematic.....                         | 107 |
| <b>Figure 6.7.</b> Sharp GP2Y0A710K0F Sensor Pin Header Schematic.....                          | 108 |
| <b>Figure 6.8.</b> VL53L0X Time-of-Flight Sensor Schematic.....                                 | 108 |
| <b>Figure 6.9.</b> Pin header for the INMP441 Microphone Module.....                            | 109 |
| <b>Figure 6.10.</b> MAX98357A Amplifier Connected to Speaker Schematics.....                    | 110 |
| <b>Figure 6.11.</b> I2S ESP 32 Setup Example.....                                               | 111 |
| <b>Figure 6.12.</b> ESP 32 Wroom MCU Schematic.....                                             | 112 |
| <b>Figure 6.13.</b> Overall Project Schematic.....                                              | 113 |
| <b>Figure 7.1.</b> Use case diagram of SmartNav.....                                            | 114 |
| <b>Figure 7.2.</b> Software Architecture Class Diagram.....                                     | 116 |
| <b>Figure 7.3.</b> Navigation API Feedback System.....                                          | 118 |
| <b>Figure 7.4.</b> Object Detection Feedback System.....                                        | 119 |
| <b>Figure 7.5.</b> GPS data processing from GPS module.....                                     | 120 |
| <b>Figure 7.6.</b> Conveying API Integration.....                                               | 126 |
| <b>Figure 8.1</b> Main PCB Board KiCAD.....                                                     | 128 |
| <b>Figure 8.2</b> Main PCB Manufacturing Preview.....                                           | 129 |
| <b>Figure 8.3.</b> TP4056 Module PCB layout.....                                                | 130 |
| <b>Figure 8.4.</b> TP4056 Module Manufacturing Preview.....                                     | 131 |
| <b>Figure 8.5.</b> GPS module PCB.....                                                          | 132 |
| <b>Figure 8.6.</b> GPS Module PCB (Top Side).....                                               | 133 |
| <b>Figure 8.7.</b> GPS Module PCB (Bottom Side).....                                            | 133 |
| <b>Figure 8.8.</b> ToF Sensor PCB.....                                                          | 134 |

|                    |                                                       |     |
|--------------------|-------------------------------------------------------|-----|
| <b>Figure 9.1.</b> | Feedback Text from Speech to Text API Connection..... | 140 |
|--------------------|-------------------------------------------------------|-----|

## Chapter 1: Executive Summary

The SmartNav Cane is an ambitious project that seeks to revolutionize mobility tools for the blind by combining advanced sensing, intelligent power management and a user-centered interface into an integrated, portable system. The need for this feature was nurtured due to the ever-increasing demand for improving the situational awareness of a user, including reliable obstacle detection along with navigational assistance to ensure a safer, confident mobility of individuals in various environments that they find themselves in. A multi-disciplinary research project into sensor technology to allow the SmartNav Cane to closely map its environment lies at the core of the device. We are targeting a solution for spatial sensing, employing a dual-sensor setting where one sensor (Ka) is located on the upper body section of the cane while the other (Kb) is on its lower extremity. However HUMON has a layered sensing strategy that detects obstacles both near and far. This technique is very important in the realistic dynamic and complex environment where potential hazards must be detected in time to avoid accidents and increase user confidence. This is complemented with our emphasis on sustainable and efficient power management. The cane is designed to have a balance of energy density, longevity, and weight, so that it can run all day long and provide users consistent support throughout their daily activities, without needing to recharge frequently. We're exploring different power sources that we can yet tap, towards an optimal solution that supports prolonged operation while keeping the physical footprint compact and tractable. Such emphasis on energy efficiency not only helps raise awareness for the practicality of the device, but also for the maintenance it can help with through a minuscule tendency for needing to change batteries. User interaction and accessibility are also a major focus in SmartNav Cane.

The focus is on the delivery of feedback in a tactile and auditory fashion, in a way that is second hand to neural processing, important for navigation and the world around us. With a simple but effective user interface, ensuring the device remains accessible to the target users, so performance okay in the middle of conditions so difficult. The cane does that by being able to respond to the dense network of feedback that are both immediate and perceptible in nature. Our solution is modular by design so you have the flexibility and the ability to enhance your service later on. As more research comes out, the cane is upgradable without the need of a complete redesign. It enables the device to grow and evolve along with the develop of new features/requirements in the usage of assistive technology. This means we have a platform that is scalable and versatile so we can always keep the cane updated and relevant. To sum it up, SmartNav Cane has proven to be a milestone & leap to technology in assistive devices. Through a unique approach, integrating thorough research, creative innovation, and functionality specifically designed for users, the project strives to provide a mobility aid that not only caters to the needs of visually

impaired individuals today but also sets the stage for further improvements in the field. Challenging this convention and set to disrupt the16 mobility aid industry, through its integration of advanced sensing, efficient power management and intuitive feedback mechanisms, empowers users with the technology to help them navigate their world with increased safety, independence and confidence.

## **Chapter 2: Project Description**

### **2.1 Project Motivation and Goals:**

In today's modern world, technology plays a large role in many people's lives. We have smartphones that allow us instantaneous access to the internet, self-driving cars that navigate the roads with precision, and AI models that quickly provide solutions to any prompt you could come up with. Despite these advancements, mobility tools for the visually impaired such as white canes have seen little improvement. This inspired us with the SmartNav Cane project. The idea is to create a technologically advanced cane to help visually impaired individuals with their daily lives. We found immediate success amongst the group due to personal reasons - at least half of us have a family member with a visual impairment. By witnessing these challenges firsthand, we decided there is a need for an innovative solution that can help improve accessibility, safety, and independence for our family members and others. While a traditional walking cane is effective, it presents certain limitations such as dealing with busy environments or running into an object out of reach. The personal connection we have to this project drives us to design the SmartNav Cane!

Our goal for SmartNav is to combine innovative engineering principles with cutting-edge technology to help build an immersive product that follows industry standards. We plan on developing a walking cane that enhances the mobility, safety, and independence of individuals. We plan on doing this by integrating features such as obstacle detection sensors and GPS navigation all while providing real-time feedback as the product is being used. It will also come with rechargeable batteries for the convenience of the client as well as a more environmentally friendly approach. Ultimately, these improvements to a traditional walking cane will significantly improve the quality of life for individuals facing visual challenges by offering a reliable tool for everyday life!

Our first and most important goal is safety, ensuring safe and reliable navigation for visually challenged individuals. The system is designed to provide navigation assistance, enabling users to move securely between walkable destinations. The smart mobility cane is designed specifically for campus environments, ensuring reliable navigation across university pathways, buildings, and student areas for seamless daily use. In addition to safety, the project emphasizes

user-friendliness, we also maintain fundamental user-friendliness as well as operational stability which is also a key focus, ensuring the device remains dependable over extended periods of use. The system aims to provide an effective mobility aid that enhances independence and confidence for visually impaired users navigating on University campuses.

## **2.2 Current Existing Projects**

Visually impaired aids have seen significant improvement in modern times, with each device offering a specific function to solve obstruction awareness and navigation problems. Despite all new approaches, most gadgets in use today have considerable gaps in them, and hence, our work aims at filling in such gaps through additional revolutionary capabilities for ease in moving, obstruction awareness, and direction, and ease for a user in moving through them comfortably.

The WeWALK Smart Cane demonstrates the significance of community-driven design by involving users with disabilities directly in the development process. This ensures the device is tailored to real user needs, making it a great innovation in assistive technology. The WeWALK Smart Cane employs sensors to detect obstacles above/below chest level, providing users with vibrational and audio feedback to alert them of nearby hazards. Additionally, it integrates an Inertial Measurement Unit (IMU) composed of a gyroscope and accelerometer to monitor movement and provide basic navigational support. These features highlight the device's user-centric approach and effectiveness in addressing some mobility challenges faced by individuals with visual impairments.

However, while the WeWALK Smart Cane introduces meaningful innovations, it also has notable limitations. Its reliance on ultrasonic sensors restricts obstacle detection to a relatively narrow range, focusing only on chest-level hazards. This limitation leaves the user vulnerable to obstacles at varying heights or distances, such as overhead branches or low-lying objects. Furthermore, the device lacks location-based navigation capabilities, which would be a game-changer for users seeking greater independence and confidence in unfamiliar environments. While the IMU provides movement analysis, it falls short of offering comprehensive navigation support, such as route planning or real-time guidance.

Recognizing these limitations, our project seeks to improve upon the strengths of the WeWALK Smart Cane while addressing its limitations. We replace the ultrasonic sensors with enhanced sensors, which provide a higher level of precision and are capable of detecting obstacles at a greater range and across a wider variety of heights. This ensures that users are alerted to a broader spectrum of hazards, enhancing their safety and awareness of their surroundings. Furthermore, our design integrates GPS modules, to introduce

real-time, location-based navigation. This addition allows users to confidently plan routes with turn-by-turn directions, follow them accurately, and maintain a clear understanding of their current position within their environment.

By addressing these gaps, our project not only improves the user experience but also expands the capabilities of assistive mobility technology. The combination of ToF sensors and GPS-based navigation creates a device that is not only more precise in obstacle detection but also significantly more versatile in navigation support. These advancements are driven by a focus on user independence and safety, ensuring that individuals with visual impairments can navigate their environments with greater ease and confidence. In doing so, our project builds upon the innovative foundation laid by the WeWALK Smart Cane, pushing the boundaries of what assistive technology can achieve.

Other assistive devices, such as the Sunu Band, offer a more minimalist approach to obstacle detection. This wrist-worn device is lightweight, portable, and user-friendly, making it an attractive option for those seeking a compact solution. It uses haptic feedback to alert users to nearby obstacles, ensuring they remain aware of their immediate environment. Despite its affordability and ease of use, the Sunu Band is limited in functionality. Its obstacle detection range is relatively short, making it less effective in complex or large environments, and it lacks any form of navigation assistance. While its simplicity serves a purpose, our project seeks to address these limitations by offering a more comprehensive solution. Our device incorporates modular design principles, combining enhanced sensors for long-range obstacle detection with GPS-enabled navigation. This dual functionality ensures users can move freely and safely, whether navigating crowded areas or open spaces.

At the cutting edge of assistive technology are devices like the Biped Smart Glasses, which leverage artificial intelligence for advanced obstacle detection and provide audio feedback to guide users. These glasses represent a leap forward in precision and functionality, offering high-tech solutions for individuals with visual impairments. However, their high cost and the frequent need for battery recharging make them less practical for everyday use, particularly for individuals on a budget or those seeking long-lasting reliability. Our project draws inspiration from the precision and innovation of devices like the Biped Smart Glasses but focuses on affordability and practicality. By using enhanced sensors, we achieve similar levels of accuracy in obstacle detection, ensuring users are alerted to their surroundings with minimal delay. Additionally, our integration of GPS navigation provides a seamless travel experience, guiding users through routes while maintaining awareness of obstacles. This holistic approach makes our solution both cost-effective and user-friendly, addressing the financial and maintenance barriers often associated with high-end technologies.

What sets our project apart is its focus on combining the strengths of these existing technologies while addressing their weaknesses. We prioritize the

inclusion of advanced features, such as high-precision sensors and GPS navigation, to create a device that excels in both obstacle detection and route planning. Our design also emphasizes accessibility and practicality, ensuring that users from diverse backgrounds can benefit from its capabilities without being burdened by high costs or frequent maintenance. By learning from the limitations of devices like the WeWALK Smart Cane, the Sunu Band, and the Biped Smart Glasses, we aim to provide a solution that not only enhances mobility but also helps visually impaired individuals to navigate their surroundings with confidence and ease.

Through thoughtful integration of cutting-edge technology and a user-centered design, our project represents a meaningful step forward in the field of assistive technology. It is not merely an improvement over existing devices but a comprehensive restructure of what a mobility aid can achieve, combining the best aspects of current innovations with new features to meet the evolving needs of its users.

### **2.3 Basic Goals (Essential Functionalities)**

Essential requirements are set by the targets: they maintain fundamental user friendliness and operational stability.

- Object Detection:** The system should be capable of detecting obstacles ahead of the user up to a specified distance with high accuracy. Obstacles to be detected include, but are not limited to, pedestrians, benches, trees, and other common hazards typically encountered on Campus environments.
- Navigation Assistance:** The system will provide real-time location guidance. This will help users safely and efficiently reach their destinations in outdoor environments.
- Obstacle Avoidance:** The system detects obstacles with precise distance measurements and provides haptic feedback with audio warnings to alert the user. It actively monitors the environment to help prevent accidental collisions.
- Adjustable Height and Ergonomic Design:** The device features an adjustable height mechanism, allowing users to select a comfortable and safe height for optimal usability.
- User Interface (UI):** The smart cane is designed with a user-friendly interface tailored for individuals with visual impairments. It features a simplified physical interface with essential buttons for easy operation and a built-in audio system that delivers clear voice guidance, real-time navigation assistance, and system updates, such as battery status.

### **2.4 Advanced Goals (Enhanced Usability & User Experience)**

- Multi Alert System: The system should be capable of providing alerts through two means: audio notifications and vibration feedback. These alert systems will be triggered based on the proximity of detected obstacles and their potential impact on the user.

- Dynamic Route Recalculation: The system ensures seamless navigation by automatically recalculating optimal routes when users deviate from planned pathways.

- Low-Battery Alerts & Power Optimization: The system should be capable of providing alerts to ensure continuous functionality. The device would also feature a power-saving mode to extend battery life.

## **2.5 Stretch Goals (Future-Oriented Features & Scalability)**

These design goals aim to enhance the cane with advanced AI and emergency response systems while future-proofing the product.

- AI-Powered Object Recognition: The system would integrate Artificial Intelligence technology through a camera system to deliver real-time object recognition for detecting various objects. Using AI technology, the device delivers complete spatial awareness that leads to advanced detection of obstacles and improved understanding of the physical environment.

- Smartphone App Connectivity: The system would be capable of pairing with a mobile app, allowing users to access features like route planning and device settings. The app will also provide offline support for stored routes and basic guidance when connectivity is limited.

- Emergency SOS Feature: The system would attain an SOS button that enables users to generate emergency alerts and show their present location to their set emergency contacts.

- Smart Home & IoT Integration: The system would be capable of working with smart home technology, allowing users to interact with devices like smart doors, indoor navigation aids, and voice assistants for a more seamless experience.

## **2.6 Objectives**

### Obstacle Detection and Classification:

- Develop algorithms to process sensor data and identify obstacles based on distance, size, and motion within a 3–4 meter range, and provide appropriate feedback based on these factors.

- Ensure reliable detection of both static and dynamic objects to enhance user safety.

#### Navigation Assistance:

- Implement a system for real-time GPS-based route guidance with turn-by-turn instructions with guaranteed 5-meter direction accuracy.
- Implement dynamic rerouting algorithms to adaptively adjust the navigation path in response to real-time deviations, minimizing user intervention

#### Feedback System Integration:

- Design and implement a multi-modal feedback system that provides haptic vibrations and audio cues for both obstacle warnings and navigation instructions.
- Ensure feedback is context-sensitive, adapting to proximity and type of detected obstacles.

#### Power Monitoring and Optimization:

- Create a battery monitoring system that alerts users when the charge drops below 20%.
- Implement power-saving features, such as standby modes, to extend battery life.

#### Ergonomic and User-Friendly Design:

- Develop an adjustable height mechanism to accommodate different user preferences.
- Design a simple interface with intuitive tactile buttons for easy control of navigation and feedback features.

## **2.4 Requirements and Specifications**

***Table 2.1. Components Specifications.***

| Parameters                       | Description                                                      | Specification              |
|----------------------------------|------------------------------------------------------------------|----------------------------|
| <b>Object Detection Accuracy</b> | Percentage of objects correctly identified                       | $\geq 85\%$                |
| <b>GPS Location Accuracy</b>     | The GPS module delivers data points with exact positional values | $\leq 3$ meters deviation. |

|                                   |                                                              |                         |
|-----------------------------------|--------------------------------------------------------------|-------------------------|
| <b>Voice Recognition Accuracy</b> | Processing accuracy of system data from voice command inputs | $\geq 95\%$             |
| <b>Voice Interface Delay</b>      | Time from user voice command to system response              | $\leq 100$ milliseconds |

***Table 2.2 System Specifications.***

| <b>Parameters</b>                | <b>Description</b>                                                           | <b>Specification</b>    |
|----------------------------------|------------------------------------------------------------------------------|-------------------------|
| <b>System Weight</b>             | Weight of the entire system                                                  | $\leq 3$ lbs.           |
| <b>Feedback Response Time</b>    | Time elapsed from detection to alert transmission                            | $\leq 500$ milliseconds |
| <b>Navigation Accuracy</b>       | Distance from the final arrival point to the intended destination            | $\leq 5$ meters.        |
| <b>Power Consumption</b>         | System's power requirements                                                  | $\leq 5W$               |
| <b>Internet Connection Speed</b> | Users need dependable internet connections to interface with Google Maps API | $\geq 2$ Mbps           |
| <b>Cost</b>                      |                                                                              | $\leq 200\$$ .          |

The requirements of this project address two main areas of focus:

1. Enhancing the mobility and safety of visually impaired individuals through advanced object detection and navigation systems.
2. Developing a reliable and compact design that integrates various assistive technologies while maintaining the functionality of a traditional mobility cane.

To meet these requirements, the team must design a smart mobility cane that incorporates key functionalities. The object detection system must identify

obstacles, including stationary and dynamic ones, with high accuracy and provide timely feedback to the user through haptic and auditory alerts. The navigation system should be powered by a GPS module capable of delivering real-time location guidance with a high level of precision. Furthermore, the cane's design should accommodate users to ensure comfort and ergonomic usability.

The system must also include a user-friendly interface at the top of the cane to control settings and receive feedback. The interface will house the MCU, which manages sensor inputs and communication protocols, alongside the GPS module for navigation assistance. These functionalities aim to deliver a device that is lightweight, durable, and efficient in operation. The specific engineering requirements and specifications for this project have been detailed in the following section.

## 2.5 House of Quality Diagram

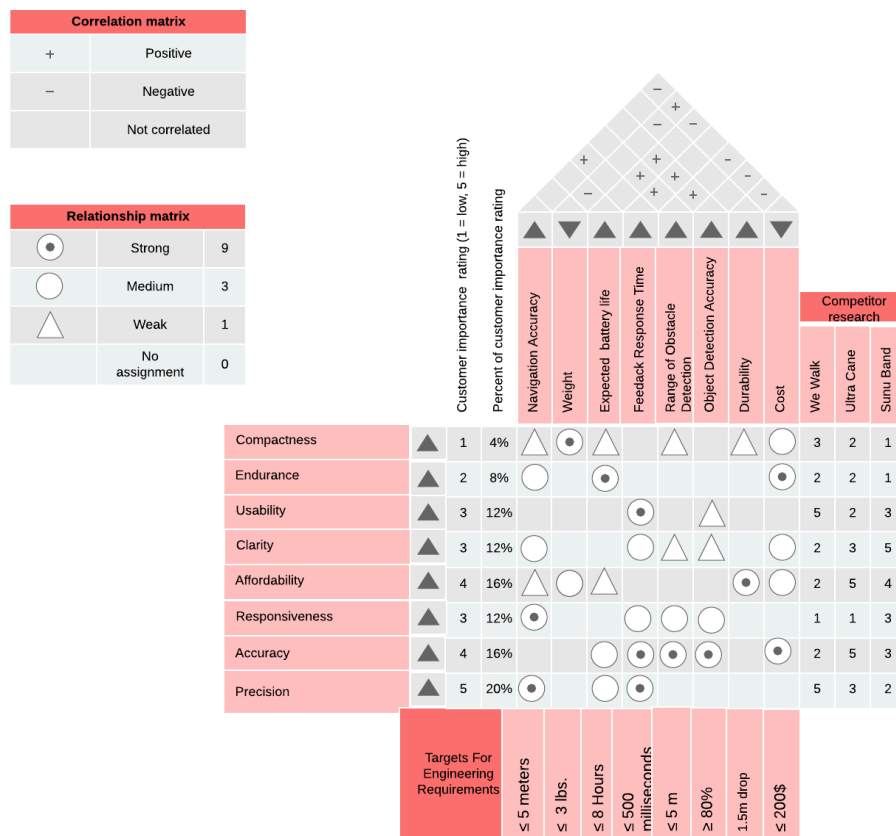


**Figure 2.1. House of Quality.**

## 2.6 Project Block Diagram/Prototype

Our prototype illustration highlights a well-engineered mobility cane that combines simplicity, functionality, and user comfort. The sensors, strategically located at the bottom tip of the cane, provide accurate real-time obstacle detection and avoidance. The adjustable height design ensures optimal comfort for users of varying needs, while the lightweight and compact build minimizes strain during prolonged use. At the top of the cane, we've conveniently placed the user interface, providing easy access to buttons and controls for enhanced usability. This area is connected to the housing enclosing key interactive components, such as the audio feedback system, the MCU, and the GPS module, ensuring smooth operation and compact integration. Retaining the familiar appearance of a traditional white cane, this design achieves a perfect balance of simplicity and advanced functionality for safe and efficient navigation.

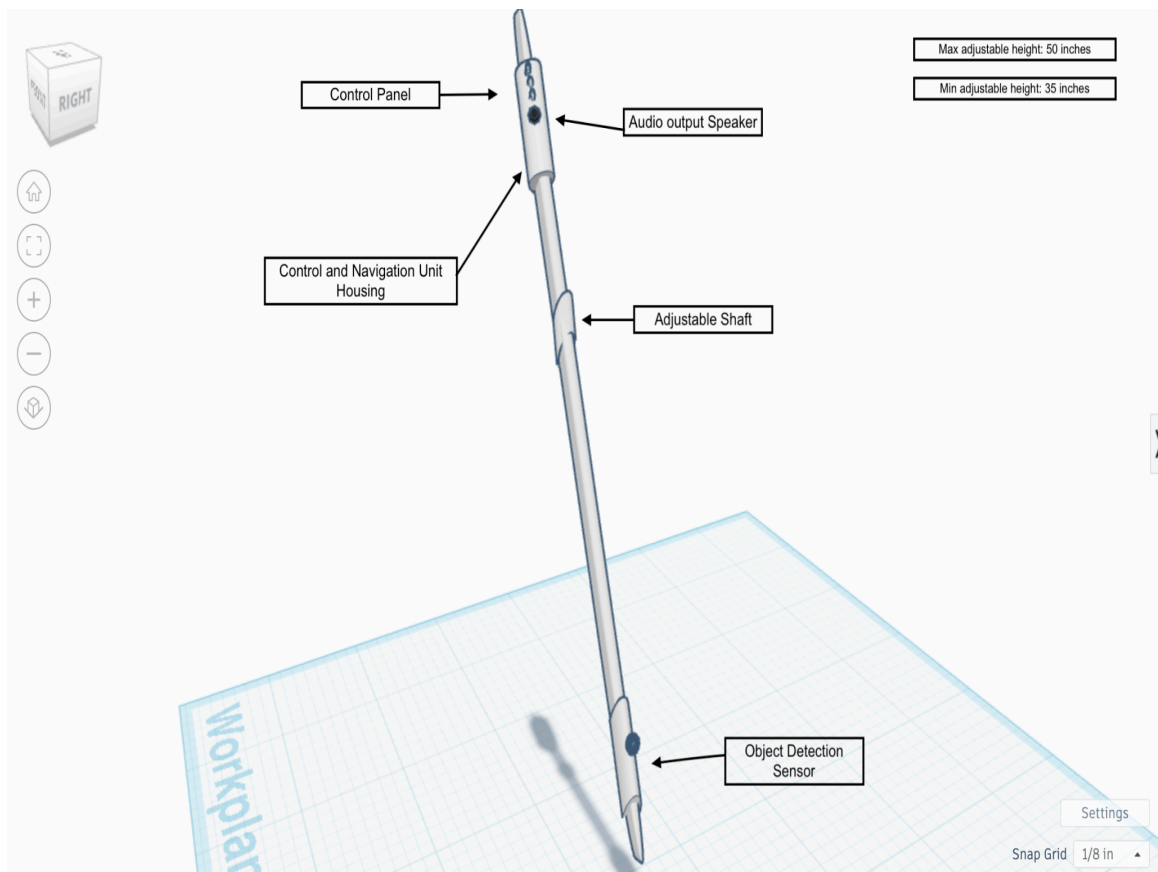


**Figure 2.2.** SmartNav Design Concept.

## 2.7 Hardware Design

### 2.7.1 Block Diagram

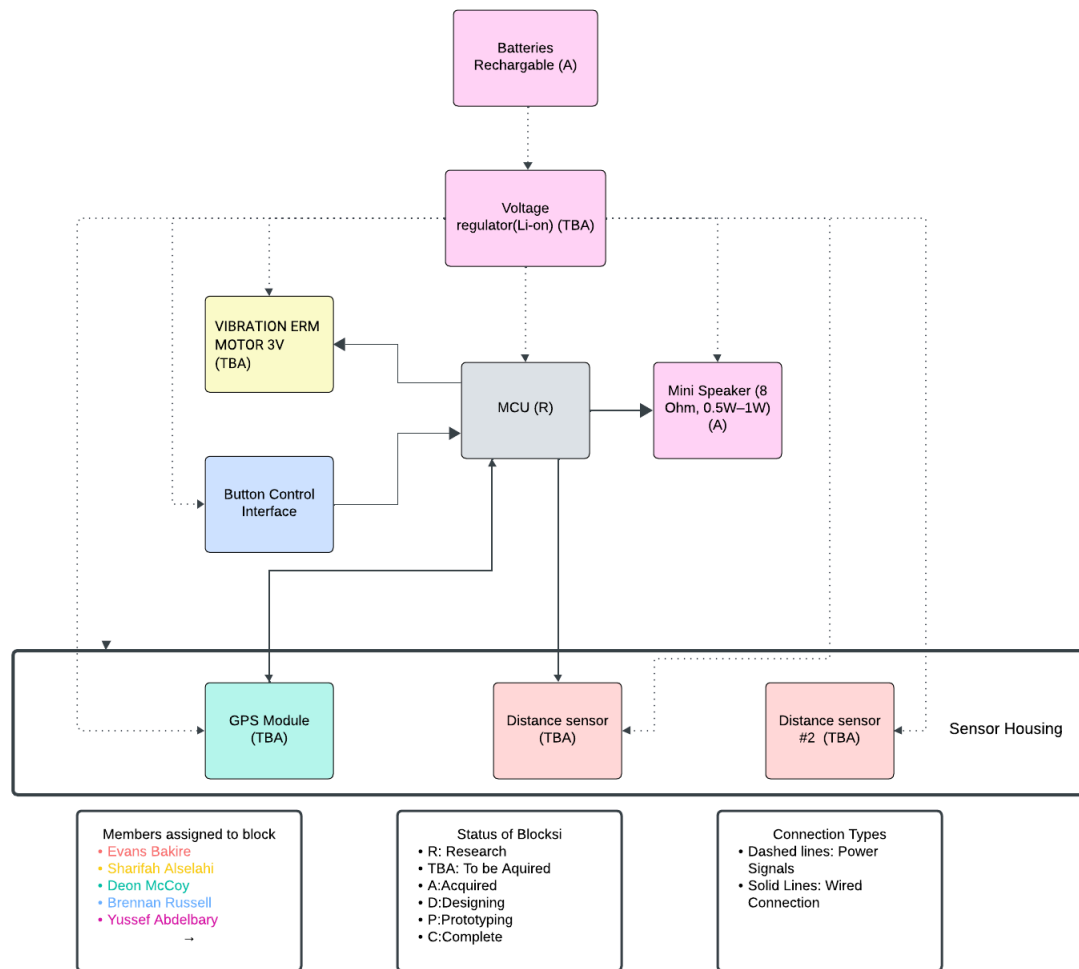


**Figure 2.3. Hardware Block Diagram.**

## 2.8 Software Design

### 2.8.1 Flow Chart



the functionality of our project. The ideal MCU must meet specific requirements to effectively support our project's objectives, which include handling API calls for real-time navigation using maps, integrating various sensor readings for obstacle detection, and maintaining seamless mcu connectivity for user interaction. As we aim to create a reliable and user-friendly device, the MCU should maintain adequate processing power to manage various tasks at the same time. This is important for our Smart Stick project as it will process incoming sensor data while maintaining a stable Wi-fi connection for real-time updates. It will also interact with various I/O pins and communicate with a mobile application.

Furthermore, the MCU should have an immense set of input/output pins to facilitate connections with various sensors and components, including object detection sensors for distance measurement, GPS module for location tracking, and possibly handle camera data and buttons or other interfaces for user input. This flexibility in design is essential for ensuring that our smartnav stick can adapt to different user needs and environments. Power efficiency is another significant consideration in our selection process, as the smart stick is intended for portable use. A low-power MCU will extend battery life, allowing users to rely on the device throughout the day without frequent recharging. This aspect is particularly important for enhancing the overall user experience, as we want to minimize inconveniences.

Moreover, the compatibility of the MCU with development environments and libraries is essential for rapid prototyping and implementation. This compatibility will allow us to leverage existing frameworks and resources, enabling easier integration of features such as GPS navigation and obstacle detection. An MCU with strong and extensive documentation will facilitate troubleshooting during the development process.

Finally, we must consider the overall cost-effectiveness of the chosen MCU, as our goal is to create an accessible solution for visually impaired individuals. Balancing performance, features, and cost will ensure that our smart stick remains affordable while maintaining reliability and efficiency. Ultimately, the right MCU will serve as the backbone of our project, enabling it to deliver a reliable and responsive user experience while fulfilling its intended purpose of assisting visually impaired individuals in navigating their surroundings safely and independently.

#### **3.1.1.1 ESP 32**

ESP32, specifically the ESP32-WROOM-32 is a very popular low-cost, power-efficient MCU. It tends to be a great option for various developers implementing complex projects and manages to preserve simplicity while utilizing this MCU. Developing robotics, wireless networks, and Internet of Things(IoT) devices is simpler when utilizing the ESP32 development boards. It is the successor to the ESP8266 and brings several improvements, making it an

excellent choice for projects that require wireless communication, efficient processing, and flexibility in design. It features a built-in microcontroller that comes pre-installed on development boards, allowing for quick setup and a reduction in the time required to get started with a project. The microcontroller is the powerful Xtensa dual-core 32-bit LX6 microcontroller, designed by Tensilica and optimized for efficient processing and low power consumption. This microcontroller architecture allows the ESP32 to handle multitasking efficiently, making it suitable for applications that require real-time data processing, sensor fusion, and wireless communication. This would be an immense trait for our SmartNav project. Another key feature of the ESP32 is its flexibility in programming, as it supports multiple development environments, including Arduino IDE, MicroPython, and Espressif's own ESP-IDF. The ESP32 MCU has clock speeds of up to 240 MHz, it provides immense computational power for handling API calls, sensor data aggregation, mobile connectivity and complex algorithms, without significant performance loss. Additionally, the ESP32 microcontroller integrates built-in Wi-Fi and Bluetooth capabilities, eliminating the need for external communication modules.

However, despite its strengths, the ESP32 does have one notable limitation that must be considered when selecting it for this Smart Stick project. One significant drawback is its limited SRAM of 512 KB, which may be insufficient for memory-intensive applications that require handling large data sets, complex computations, or advanced machine learning models. Despite this constraint, the ESP32 remains a great choice due to its affordability, ease of use, and complex data management.

#### **3.1.1.2 Raspberry Pi 4 Model B**

Due to its extremely low cost, small size, and maker-friendly design, the Raspberry Pi line of single-board computers has become incredibly popular. One of the most potent yet is the Raspberry Pi 4 Model B, which keeps the same low cost and small size as previous models while gaining a significant boost in processing power, improved video output, and peripheral connectivity. The Raspberry Pi 4 Model B stands out as a strong option for our project due to its balance of affordability, processing power, and connectivity. It features a 1.5-GHz quad-core Broadcom BCM2711B0 (Cortex-A72) processor and up to 4GB of LPDDR4 SDRAM, which makes it significantly more powerful than earlier models. This could be useful for our project that may require multitasking or handling real-time data. It also has USB 3.0 ports and built-in Wi-Fi and Bluetooth, which would allow us to interface with different peripherals. The Broadcom BCM2711 is a quad-core Cortex-A72 64-bit system on a chip that runs at 1.5 GHz. The 40-pin GPIO header keeps it compatible with existing Raspberry Pi accessories, meaning we would have access to a lot of expansion options. Software-wise, it runs Raspberry Pi OS formerly known as Raspbian, which supports programming environments like Python and C++, making it flexible for different types of applications. One constraint to consider is heat management,

the Pi 4 runs hotter than previous versions, so depending on our use case, we might need a cooling solution like a heatsink or fan as the SMARTSTICK should be able to be used in all conditions. This would lead to extra weight which is not desired in our project.

### **3.1.1.3 Arduino**

Moreover, for the applications with a demand for sufficient input-output expander, Arduino offers good versatility, ease of use and is supported by a very strong community. Arduino is an open-source hardware and software platform originally developed by students in Italy in 2003. They provide an affordable and flexible microcontroller that is widely used today. The Arduino family of boards including the Arduino Uno or Arduino Nano is packed with CPU cores and useful I/O interfaces that allow them to work with different sensors and actuators; hence they are the best candidates for our project. SmartNav, as it is intended to help mobility, requires both obstacle detection and navigation guidance, thus needing a system capable of handling sensor inputs and control outputs in real time. Arduino's C++ based programming environment is very user-friendly. They provide open-source with excellent libraries and example codes available for use and development, speeding up the coding process. So, most of the Arduino boards like Arduino Uno based on ATmega328P Microcontroller contains just 32 KB Flash memory, 2 KB SRAM, and 1 KB EEPROM running the clock speed of 16 MHz. Also, the 14 digital and 6 analog I/O pins on the board enable us to connect the necessary sensors for connecting with our environment (as in distance measurement, obstacle avoidance, etc.). We can use Arduino Wi-Fi modules e.g Arduino Wi-Fi Shield or Arduino MKR WiFi 1010. This ensures proper internet connectivity for accessing data remotely and for integrating GPS-based navigation features, both of which are important for our project. One limitation to keep in mind is processing power – if you plan to pursue AI-based navigation, Arduino is not up to the task of high computations. We may find it hard to integrate it for more advanced features later (e.g., machine learning-assisted obstacle recognition) and we might find ourselves refactoring the system or looking at alternative co-processors.

### **3.1.1.4 STM32F411CEU6**

Another possible contender is an STMicroelectronics microcontroller, the STM32F411CEU6. It provides padded processing capabilities with its Cortex-M4 core clocked at 100 MHz, which will allow it to handle real-time data like object detection, navigation, sensor data, etc. Its powerful 512 KB flash and 128 KB RAM is enough to save the firmware, navigation algorithms, and dynamically, sensor data. Since our cane will need more than one peripheral such as sensors and audio audio feedback, the STM32F411CEU6 can achieve this with the use of I2C, SPI and UART, which enables the communication to any device component. Powerwise, it has a few low power modes and is rated for 1.7 - 3.6 V operation, which will help us achieve our goal of long-lasting battery life. The downside of

this microcontroller is the lack of built-in wi-fi, so we would need to wire another module for wi-fi if needed.

### 3.1.1.4 Final MCU Comparison

**Table 3.1. MCU Comparisons Table.**

|                        | Raspberry PI                               | Arduino                             | ESP 32                           |
|------------------------|--------------------------------------------|-------------------------------------|----------------------------------|
| Price                  | \$ 50                                      | \$ 20                               | \$ 15                            |
| Processing Power       | 1.5 Ghz                                    | 16 Mhz                              | 240 Mhz                          |
| RAM                    | 4 GB                                       | 2 KB SRAM                           | 512 KB SRAM                      |
| Wireless Connectivity  | Yes                                        | Yes                                 | Yes                              |
| Bluetooth Connectivity | No                                         | Yes                                 | Yes                              |
| GPIO Pins              | 40-pin GPIO header                         | 14 digital I/O, 6 analog input pins | 36 Gpio pins                     |
| Weight                 | Heavier (may require cooling solutions)    | Lightweight                         | Lightweight                      |
| Expandability          | Excellent (USB 3.0, GPIO, and accessories) | Good (supports shields and modules) | Good (supports external modules) |
| Operating Voltage      | 5 V                                        | 5 V                                 | 3.3 V                            |
| Flash Memory           | Up to 1 TB with microSD card for storage   | 32 KB                               | 16 MB                            |
| Power Consumption      | 3-7 W                                      | 50 mA                               | 160 mA                           |

In conclusion, after carefully considering the options, the ESP32 is selected as the ideal microcontroller for the SmartNav. The ESP32 can be a cost-effective solution with low-power consumption while having built-in wireless network (Wi-Fi and Bluetooth) functionality to process and transmit the real-time

sensor readings. With enough processing power to handle obstacle detection, navigation algorithms and API calls without major performance degradation, its dual-core 32-bit LX6 microcontroller can run up to 240 MHz. In addition, the ESP32's minimalistic approach aligns with our objective of creating an easily accessible, portable device. The new Raspberry Pi 4 is arguably the biggest competitor, due to its processing power and ability to run multiple applications simultaneously for a fraction of the average price. The fact that it consumes much more power and produces more heat means we would need to include a cooling mechanism. This would ultimately increase the overall weight of our cane which is undesirable.

### **3.1.2 Object Detection Technology**

In developing an advanced system for visually impaired users, the selection of appropriate object detection sensors is crucial for ensuring safety and usability in various environments. This research focuses on identifying the most effective sensors for our smart stick project, which aims to enhance the independence and confidence of visually impaired individuals navigating university campuses. After thorough consideration of available technologies, we examined a range of sensors, including Time-of-Flight (ToF) sensors, LiDAR, radar systems, photoelectric sensors etc. Each of these technologies employs different principles to measure distances and detect obstacles from laser beams to provide high-accuracy distance measurements and radio waves, allowing them to penetrate through certain materials and detect objects in diverse weather conditions.

In our analysis, we prioritized sensors that can provide accurate, real-time feedback on obstacles surrounding the user, effectively sensing distances and angles to enhance spatial awareness. We specifically considered how these sensors could effectively identify hazards such as curbs, benches, and pedestrians, ensuring users can navigate safely and efficiently. Additionally, we aimed to implement a multi-modal feedback system that integrates audio cues and haptic vibrations, enhancing user interaction and guidance. Ultimately, the goal of this research is to select sensors that not only provide precise distance measurements but also operate effectively in varied environments, supporting the overarching mission of empowering visually impaired individuals in their daily lives. In our design we aim to implement two object detection sensors, one at the bottom of the cane to detect steps, curbs, and other obstacles or height changes, and one at the top of the cane for a 180 degree field of view. These two positions are crucial to implement the design of our project effectively.

#### **3.1.2.1 Time-of-Flight (ToF)**

Time-of-flight (ToF) sensors implement distance measurement by using light pulses, usually infrared or laser, to calculate the time it takes for the light to travel to an object and return. The speed of light is the foundation of the time-of-flight

principle. The receiver, also known as a SPAD (Single Photon Avalanche Diode), detects photons that are reflected by the target after being sent by an emitter. The precise distance of the target in millimeters is determined with high accuracy by the time difference between the emission and the reception. Using the equation  $\text{distance} = (\text{speed of light} \times \text{time delay}) / 2$ , ToF sensors can determine how far an object is with millimeter-level accuracy. Since light moves at an incredibly fast speed of approximately 299,792,458 m/s, the sensor requires extremely precise time measurements, often in the nanosecond range. One potential candidate for our smart stick project is the VL53L1X ToF sensor, developed by STMicroelectronics, which has been widely adopted for its high accuracy, fast response time, and ability to function in a variety of lighting conditions, including complete darkness. This sensor boasts an extended range of up to 4 meters, millimeter-level precision, and an adjustable field of view, allowing it to detect obstacles at different distances and angles. Compared to ultrasonic sensors, which can struggle with accuracy due to sound wave dispersion and sensitivity to certain surfaces, the VL53L1X provides a more consistent and reliable performance across a wide range of materials. Historically, ToF technology has been used in industries such as robotics, automotive safety, and smartphones for facial recognition and augmented reality, but its application in our assistive device for the visually impaired represents a growing and impactful use. Integrating this sensor into the smart stick could significantly enhance user awareness by detecting curbs, stairs, objects at various heights, and approaching obstacles in real-time, providing immediate haptic or auditory feedback to the user. Additionally, the smart stick's low power consumption makes it ideal for portable applications, guaranteeing that it will continue to operate effectively and efficiently for extended periods of time. Utilizing this cutting-edge ToF sensor, we can create a more responsive and intelligent mobility assistance that improves visually impaired people's safety, independence, and simplicity of navigating.

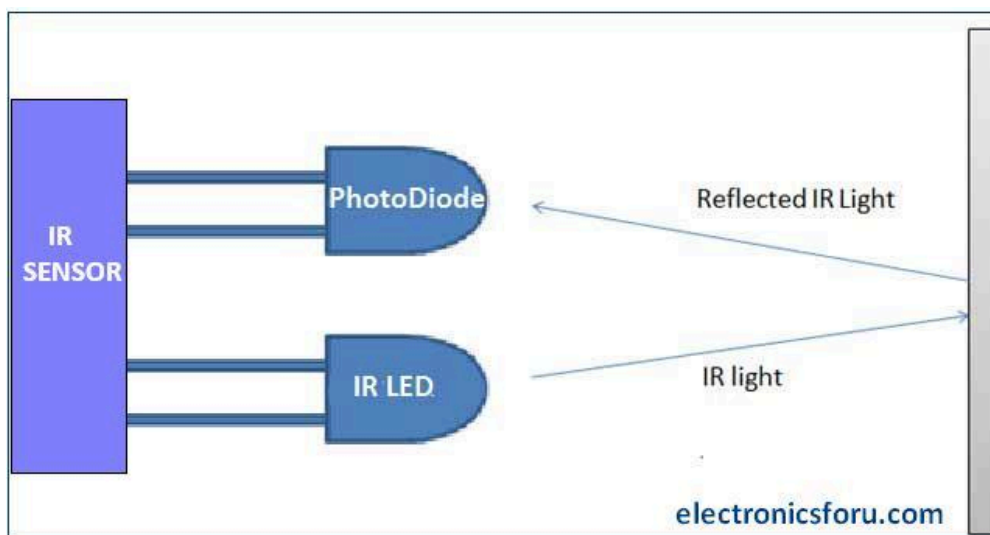


**Figure 3.1.** *Principle of an Infrared sensor.*

### **3.1.2.2 Lidar**

Light Detection and Ranging is a remote sensing technology that utilizes laser pulses to measure distances with exceptional accuracy. The physics behind the LiDAR is similar to Time-of-Flight (ToF) but operates with higher power laser beams, allowing for longer-range detection and greater precision. LiDAR sensors emit laser pulses, typically in the infrared spectrum, which travel to a target object and reflect back to a receiver. By measuring the time taken for the laser to return and applying the equation  $\text{distance} = (\text{speed of light} \times \text{time delay}) / 2$ , the sensor can create a precise depth map of the environment. Unlike ToF sensors, which generally provide single-point distance measurements, LiDAR systems can generate a detailed 3D map of surroundings by scanning across multiple angles, making them highly effective for spatial awareness applications. The origins of LiDAR trace back to the 1960s, when it was developed following the invention of the laser, initially for military and atmospheric research. Over time, advancements in miniaturization and cost reduction have made LiDAR widely used in fields like autonomous vehicles, robotics, and navigation aids for visually impaired individuals.

A strong contender for our smart stick project is the benign wave TFmini-S LiDAR, which has a compact size, low energy consumption and is not expensive. It is a LiDAR module designed for the short range of detection (0.1 to 12 meters). The sensor operates using a 905 nm infrared laser, and is highly resistant to ambient lighting variations. Unlike ultrasonic sensors which struggle with irregular surfaces and environmental noise, LiDAR is consistent and fast. It does not have problems with different materials, it simply represents whatever the light reflects off of into data. Using the TFmini-S in our project would immensely improve obstacle detection at different heights, making for a more reliable and responsive navigation system that could be used in practice by people with visual impairments. With LiDAR technology in our smart stick design, we are creating a real-time navigator that is both highly efficient at providing independent safety to visually impaired people and lets them find and navigate their surroundings more quickly than when they rely on traditional close-in sensors.

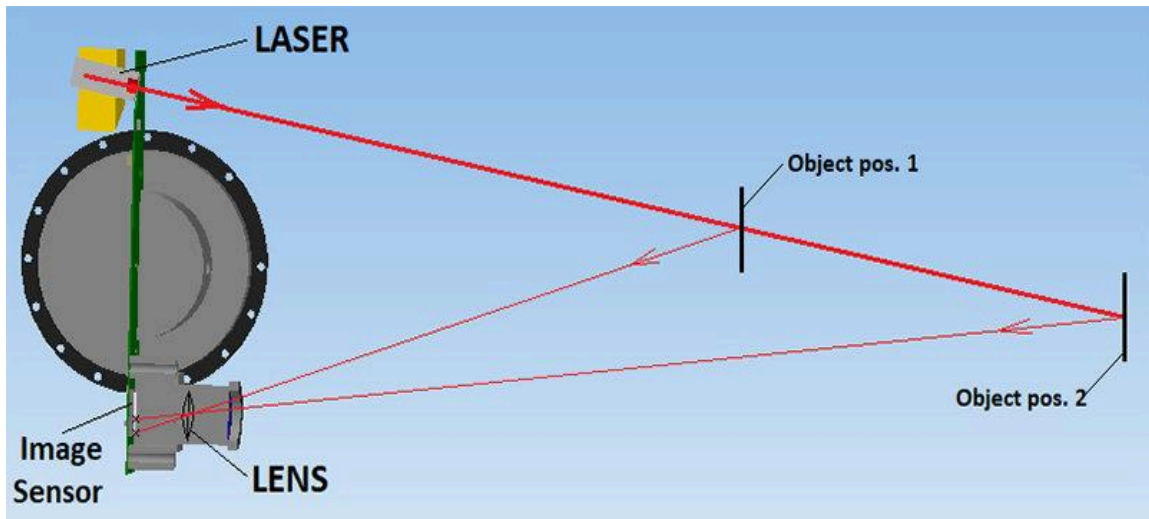


**Figure 3.2.** Principle of a LIDAR sensor.

### 3.1.2.3 Ultrasonic

A more traditional and popular choice, an ultrasonic sensor, has also been considered for our smart stick. Ultrasonic sensors work by using ultrasonic waves to measure distance. It sends a pulse out and catches the echo to determine how far it is, so the transducer transmits and receives a high-frequency sound wave (between 20 kHz to 40 kHz) above the human hearing range. The sensor calculates the distance to that object by measuring the time it takes the pulse to reach the object and return. One major advantage of ultrasonic sensors is that they are independent of lighting conditions (in contrast to optical sensors such as Time of Flight or LiDAR), which allows them to be used in different environments, from illuminated daylight to low-lit and even dark areas. Besides that, they won't care with the surface texture as ultrasonic sensors can detect objects with different texture. They have a wide range of possible applications because they can reliably operate in both indoor and outdoor environments under any lighting condition.

As mentioned earlier, the sensor operates within a frequency range of 20kHz - 40 kHz, which is well above the human hearing range, ensuring it does not interfere with environmental sounds. It has a detection range from 2 cm to 4 meters, making it ideal for detecting obstacles at various distances, which is essential for a mobility aid like the smart stick. The sensor offers an accuracy of up to 1 cm, providing precise feedback to the user. The field of view (FoV) is typically 15 degrees, which allows it to detect objects within a specific range of angles, helping the stick sense obstacles to the front and sides of the user. With a typical response time of 20 ms, the sensor can update its measurements rapidly, ensuring that users receive timely feedback as they navigate their environment. The compact design and low power consumption (typically 15 mA at 5V) make it

suitable for integration into the smart stick, ensuring it does not significantly impact battery life while providing continuous, real-time obstacle detection.

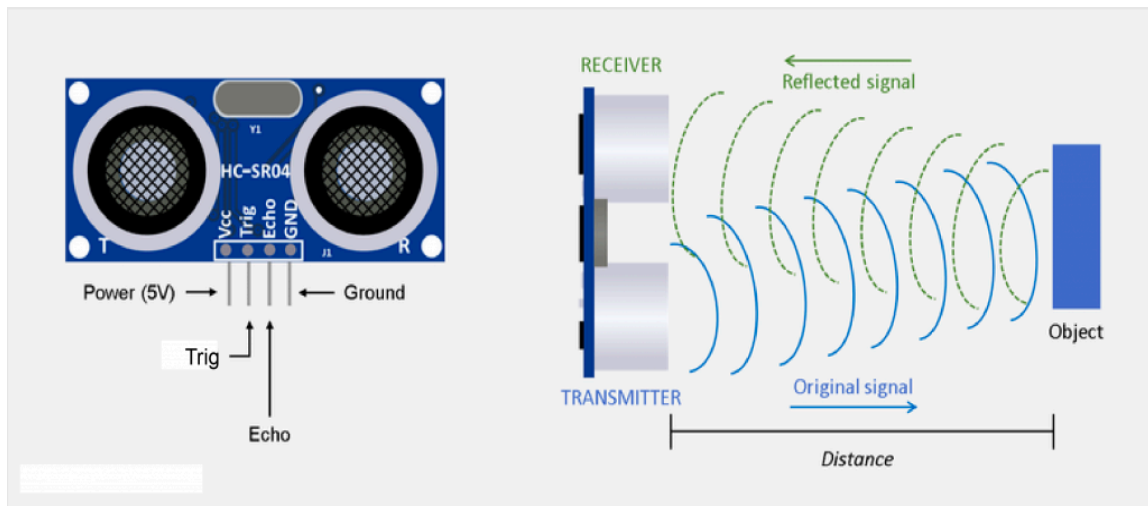


**Figure 3.3.** Principle of an ultrasonic sensor.

#### 3.1.2.4 Photoelectric

Another sensor technology we have considered is photoelectric sensing, which is known for its high reliability in various settings. These sensors work by emitting infrared light and detecting the reflection or interruption of that light beam. We carefully considered the three main types of photoelectric sensors, through-beam, retro-reflective, and diffuse. Each of these offers different advantages depending on the environment and application. Through-beam sensors, which use separate light transmitters and receivers placed opposite each other, detect objects when they interrupt the light beam. Retro-reflective sensors combine the transmitter and receiver into one unit and rely on a reflector placed across the sensor. The sensor detects objects by receiving the light reflected back from the reflector. Diffuse sensors, unlike the other two, use a single unit for both transmitting and receiving light and detect objects based on the light reflected from them.

When considering the SmartNav design, we explored ways these sensors could interface with the system and allow for detection of obstacles in the environment. Since the SmartNav stick would need to detect obstacles in both the near and far field, it was advantageous that photoelectric sensors have a great deal of versatility in the ranges of objects they could detect (either millimeters or several meters, depending on the model; many models have a range of up to 100 meters). The response time on these sensors ranges from 1 ms to 10 ms which matched our desired goals for fast obstacle detection with real-time feedback for the user. Along with different options to set these photoelectric sensors to one of two operating modes, light-on and dark-on, they also gave us flexibility in terms of what we could use in how we configured sensors to respond to obstacles in

the SmartNav environment. In terms of key specifications, photoelectric sensors are available in models that can work in unfavorable environmental conditions, i.e., with protection ratings of IP65 to IP69K. These ratings are important because the sensors may need to withstand dust, water, and cleaning processes, particularly if the SmartNav stick will be used outdoors, where the weather can vary. Also, photoelectric sensors can run for a very long time, with many models having lifetimes in excess of 10 million operations, which makes them durable for a long-term use. When looking at the power draw the spec for photoelectric sensors shows us that they generally run between 10V and 30V DC, which can be achievable in our project.

While photoelectric sensors are highly accurate and versatile, we also acknowledge that they may face limitations when detecting certain materials, such as those that do not reflect light well. However, in environments where there is sufficient light and the materials to be detected are reflective, photoelectric sensors provide a highly efficient and reliable solution. Ultimately, incorporating photoelectric sensors into the SmartNav mobility stick would enhance its ability to detect obstacles and provide the user with timely feedback, making it a viable option for ensuring the safety and usability of the device.

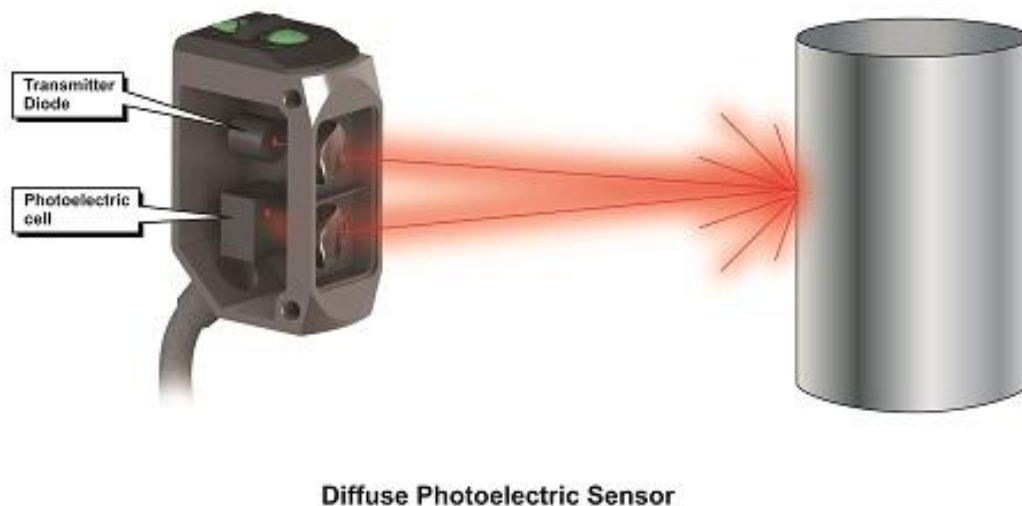


**Figure 3.4.** *Principle of a Photoelectric sensor.*

### **3.1.2.5 mmWave**

For our SmartNav project, we also considered mmWave (millimeter wave) sensors due to their exceptional abilities in obstacle detection, which fit the needs of a mobility stick designed for the blind. These sensors operate in the 30 GHz to 300 GHz frequency range, corresponding to wavelengths from 1 millimeter to 10 millimeters, giving them their name. This frequency range allows mmWave sensors to emit electromagnetic waves with high precision, offering excellent

resolution in detecting objects from a distance. The primary advantage of mmWave technology is its ability to penetrate through various materials like clothing, plastic, wood, and even some metals. This makes them ideal for dynamic environments, where traditional sensors like infrared or ultrasonic might struggle. The physics behind mmWave sensors is based on the reflection of high-frequency electromagnetic waves. When these waves encounter an object, they bounce back to the sensor, which calculates the distance based on the time it takes for the waves to return. This provides a highly accurate and real-time mapping of the environment, detecting obstacles even at a distance greater than most other sensors can achieve. Additionally, mmWave sensors can measure the velocity of moving objects, which is crucial for navigating environments with dynamic obstacles, such as pedestrians or moving vehicles. The typical range of mmWave sensors is between 3 to 10 meters, with some advanced models extending farther. The angular resolution of these sensors can be as precise as 1 degree, allowing for accurate identification of narrow obstacles and distinguishing objects in crowded spaces. These sensors also provide a wide field of view, often greater than 120 degrees, which enables comprehensive scanning of the environment without the need for multiple sensors.

Considering the power consumption, mmWave sensors are known for their energy efficiency, which is particularly important for our SmartNav project, as the mobility should be long-lasting. The low power consumption ensures that the device can operate for extended periods without frequent recharges. The compact design of mmWave sensors makes them a perfect fit for the SmartNav stick, as they do not add significant bulk to the device. However, one drawback is that mmWave sensors can be affected by adverse weather conditions such as rain, snow, or fog, which may cause signal attenuation, potentially reducing the sensor's range and accuracy. Despite this, the ability of mmWave sensors to detect both stationary and moving objects with high resolution and long-range capabilities makes them a valuable addition to our project. By using mmWave sensors, we can enhance the SmartNav stick with sophisticated, real-time obstacle detection that works well in complex environments, ultimately improving the stick's functionality, safety, and usability for visually impaired users.

#### **3.1.2.6 Infrared Distance**

Infrared distance sensors could be a suitable option for the cane's lower end to detect curbs, steps, and drop-offs. It uses an infrared beam and photodiode to measure distance via triangulation, offering a long range of about 1–5 meters. It has a narrow field-of-view, effectively focusing on what's directly ahead on the ground, which is ideal for distinguishing a step edge or an upcoming curb without being confused by surrounding objects. Infrared rangefinders provide continuous analog distance readings with good accuracy over the required range. In testing, a Sharp IR sensor was able to detect the floor at various tilt angles "with accuracy," whereas a comparable ultrasonic sensor gave more erratic readings. This consistency is crucial for reliably sensing a drop-off versus flat ground. An IR

sensor can notice a sudden change in distance (e.g. floor distance jumping from ~2 m to “no return”) that indicates a step down or open drop. In one study, mounting an IR sensor at ~35° downward angle led to 97% detection success for a 4 ft drop (train platform) and 90% for staircase steps in real-world trials. The IR sensor’s narrow beam means it would look at a specific spot ahead of the cane, which is advantageous for ground detection. This is because it will primarily sense the ground directly in front of the user. A wide FOV sensor would constantly “see” the ground and might have trouble distinguishing a sudden change. The Sharp sensor’s effective cone is roughly on the order of a few tens of degrees, which proved sufficient when angled properly. By mounting it at ~35° downward, it monitors a patch of ground a meter or two ahead; when that patch yields no reflection (or a much longer range reading), it clearly indicates a drop-off. This focused detection helped the system differentiate a true drop from a shallow gutter or small object. The IR rangefinder actively measures distance and doesn’t require motion to detect an object. It can pick up a static curb or stair edge just as well as a fast-moving object near the ground (for example, a rolling ball or an approaching vehicle tire) – as long as the object enters its line of sight.

### 3.1.2.7 Sensor Technology Comparison

**Table 3.2. Sensor Technology Comparison.**

| Parameters    | ToF           | Lidar      | Ultrasonic              | Photoelectric   | mmWave                     | IR Distance          |
|---------------|---------------|------------|-------------------------|-----------------|----------------------------|----------------------|
| Principle     | Light delay   | Laser scan | Sound echo              | IR reflect      | EM reflect                 | Triangulation        |
| Range         | ~4m           | 0.1–12m    | 2cm–4m                  | 0–100m          | 3–10m                      | 1–5m                 |
| Accuracy      | High          | Very High  | Mod                     | High            | High                       | Mod                  |
| Field of View | Var           | Wide       | Narrow                  | Var             | Wide                       | Narrow               |
| Robust        | Good          | Good       | Good                    | Med             | Med                        | Med                  |
| Cost          | Med           | High       | Low                     | Med             | High                       | Low                  |
| Power         | Low           | Med        | Low                     | Low             | Low                        | Low                  |
| Integration   | Easy          | Mod        | Easy                    | Easy            | Hard                       | Easy                 |
| Pros          | Precise, fast | 3D mapping | Low cost; works in dark | Fast; versatile | Detects moving; wide cover | Effective for ground |

|      |                                   |                    |                             |                 |                               |                              |
|------|-----------------------------------|--------------------|-----------------------------|-----------------|-------------------------------|------------------------------|
| Cons | Timing critical;<br>range limited | Complex;<br>costly | Noise sensitive;<br>low res | Light dependent | Weather sensitive;<br>complex | Limited range;<br>narrow FoV |
|------|-----------------------------------|--------------------|-----------------------------|-----------------|-------------------------------|------------------------------|

For our project, the infrared distance and the time-of-flight sensing technologies are the best to implement. The planned approach of a dual sensor system is crucial in achieving high positive feedback and accurate results, which can be achieved through the use of these technologies. An IR sensor provides a low cost / low power solution with a larger detection range that is effective for detecting objects above ground level. On the other hand, a ToF sensor provides a higher priced / low power solution with high object detection accuracy for objects near the ground. These two technologies significantly complement each other and provide the best option of attaining the precise object detection accuracy we desire, which is why they are the best technologies moving forward in our project.

### 3.1.3 Object Detection Sensor Component Selection

In this section, we detail the rationale behind our decision to employ a dual-technology approach for object detection in the SmartNav Smart Stick, utilizing Time-of-Flight (ToF) and Infrared (IR) sensor technologies. Our preliminary analysis indicates that these two sensor types offer complementary capabilities that are ideally suited for enhancing user safety. The ToF technology, renowned for its high precision and rapid response, is optimally positioned at the bottom of the stick to detect immediate, low-level obstacles such as steps, curbs, and small objects that could pose a tripping hazard. This strategic placement ensures that detailed distance measurements are captured at close range, which is critical for safe navigation. Conversely, IR sensors are recognized for their capacity to cover a broader detection area with a wide field-of-view, making them well-suited for mounting near the top of the stick to sense larger, more distant obstacles. This top-mounted configuration provides early warnings for overhead or far-field hazards, creating a layered detection strategy that significantly enhances overall system reliability. Our component selection strategy will focus on evaluating sensor models that meet our stringent requirements for range, accuracy, power efficiency, and ease of integration. By establishing this dual-sensor framework, we lay the groundwork for a robust, multi-modal object detection system that not only improves the functionality of the SmartNav Smart Stick but also ensures a higher level of safety for its users. Detailed sensor models and their comparative specifications will be introduced in the subsequent sections.

#### 3.1.3.1 Analysis of ToF Sensors for Low-Level Obstacle

For the SmartNav Smart Stick designed to assist visually impaired users in detecting lower obstacles such as steps, curbs, and uneven surfaces we conducted an in-depth comparison of three Time-of-Flight (ToF) sensors to determine the optimal choice for our design. The VL53L1X stands out as our primary candidate, offering a detection range of up to 4 meters, high millimeter-level precision, and an adjustable field-of-view that can be tuned to cover a wide area, making it highly adaptable for integration into our stick's lower section. Its moderate power consumption and robust performance in diverse lighting conditions position it well despite its higher cost compared to other models. In contrast, the VL53L0X, which operates on a similar principle but with a shorter detection range of around 2 meters, presents a simpler and more cost-effective solution; however, its limited range may leave gaps in obstacle detection on uneven or rapidly changing terrains. The VL6180X, known for its ultra-short-range measurement and exceptionally fast response times, excels in capturing immediate proximity data but is constrained by a detection range typically under 1 meter, making it ideal for rapid detection of very close objects yet insufficient as a standalone sensor for comprehensive low-level mapping. When evaluating these options, we considered not only the raw specifications such as range, precision, and field-of-view but also integration complexity, cost-effectiveness, and power requirements. The VL53L1X, while more complex and pricier, provides the best balance of range and adaptability for ensuring that our smart stick reliably alerts users to hazards at various distances, whereas the VL53L0X and VL6180X could serve as complementary sensors to bolster performance in niche scenarios or in a hybrid system. This detailed comparison underscores the critical importance of aligning sensor capabilities with our design goals to deliver a safe, efficient, and user-friendly navigational aid.

**Table 3.3. Sensor ToF Comparison.**

| Parameter              | VL53L1X          | VL53L0X          | VL6180X             |
|------------------------|------------------|------------------|---------------------|
| Detection Range        | Up to 4m         | Up to 2m         | Under 1m            |
| Accuracy               | Millimeter-level | Millimeter-level | Ultra-fast response |
| Field of View          | Adjustable       | Fixed            | Narrow              |
| Power Consumption      | Moderate         | Low              | Very Low            |
| Price                  | High             | Low              | Low                 |
| Integration Complexity | Moderate         | Low              | Low                 |
| Response Time          | Fast             | Moderate         | Very Fast           |
| Operating Voltage      | 3.3V             | 3.3V             | 3.3V                |

|                           |                                                             |                                                    |                                                                           |
|---------------------------|-------------------------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------------|
| Interface                 | I2C                                                         | I2C                                                | I2C                                                                       |
| Environmental Suitability | Adaptable to varied lighting                                | Optimized for close-range                          | Best for immediate proximity sensing                                      |
| Comments                  | Optimal for SmartNav with extended range and adjustable FoV | Good cost-effective option for limited range needs | Excellent for rapid close obstacle detection but limited standalone range |

### 3.1.3.2 Analysis of Infrared Distance Sensors for High-Level Obstacle Detection

We examined infrared distance sensors intended for deployment at the higher end of the SmartNav Smart Stick. By positioning a sensor at the top, our goal is to achieve a comprehensive 180° field-of-view that can capture obstacles and hazards in both the forward and overhead regions, providing early warnings to users. Our primary candidate is the Sharp GP2Y0A710K0F, which offers an extended detection range of 1–5 meters and high accuracy, making it attractive for high-level monitoring. To ensure we select the most suitable component for our application, we compare the GP2Y0A710K0F with two alternative sensors the Sharp GP2Y0A21YK0F and the Sharp GP2Y0A41SK0F. Although these sensors are typically used for close-range detection, we are evaluating their potential for high-level integration by analyzing key parameters such as detection range, accuracy, field of view, power consumption, price, integration complexity, response time, operating voltage, interface, and environmental suitability. This detailed comparison allows us to weigh the trade-offs between performance, cost, and design complexity, ensuring that the selected sensor not only meets the functional requirements but is also optimally adapted for a 180° detection configuration in our SmartNav system.

**Table 3.4. Sensor Infrared Comparison.**

| Parameter         | Sharp GP2Y0A710K0F | Sharp GP2Y0A21YK0F | Sharp GP2Y0A41SK0F |
|-------------------|--------------------|--------------------|--------------------|
| Detection Range   | 1-5 m              | 10-80 cm           | 4-30 cm            |
| Accuracy          | High               | Moderate           | Moderate           |
| Field of View     | Narrow             | Narrow             | Narrow             |
| Power Consumption | Moderate           | Low                | Low                |
| Price             | High               | Low                | Low                |

|                           |                                                                           |                                                                     |                                                                     |
|---------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------|
| Integration Complexity    | Moderate                                                                  | Low                                                                 | Low                                                                 |
| Response Time             | Fast                                                                      | Fast                                                                | Fast                                                                |
| Operating Voltage         | 5V                                                                        | 5V                                                                  | 5V                                                                  |
| Interface                 | Analog                                                                    | Analog                                                              | Analog                                                              |
| Environmental Suitability | Moderate (Varied)                                                         | Indoor                                                              | Indoor                                                              |
| Comments                  | Best for extended low obstacle detection; ideal for step and curb sensing | Effective for short-range detection; limited for extended obstacles | Ideal for close-range detection; suitable as a supplementary sensor |

### 3.1.4 Global Navigation Satellite System (GNSS) Module

The SmartNav Cane should be capable of providing real-time location and navigation guiding users through moderately busy to low traffic areas to efficiently reach their destinations.

In choosing the appropriate GNSS module, several factors came into play. We had to choose a module that not only has high accuracy, but also has low power consumption, small form factor, and fast response rate. The communication interface must also be taken into consideration, as it could be UART, I2C, SPI, or USB. In any case, it must be compatible with our chosen MCU. Other considerations include the type of antenna used. Whether it be internal or external, it must be reliable at receiving a signal without being too bulky.

#### 3.1.4.1 NEO-M9N-00B

The NEO-M9N-00B GNSS is a 92-channel module from u-blox. The NEO-M9N-00B is capable of concurrently receiving 4 GNSS; GPS, GLONASS, Galileo, and BeiDou constellations. This will result in higher accuracy in certain places. An external antenna is required and will be implemented via a U.FL connector. This module has a 25 Hz max update rate with multi GNSS as well as single GNSS configuration, which will be more than sufficient for the application in which it is being used. This module has an accuracy rating of about 2.0m CEP. This module runs off 3.3 V and draws about 31mA for GPS tracking using the breakout board/circuit design from SparkFun. There is also a low power mode for optimized battery life of the SmartNav Cane. The supported protocols are UBX, NMEA( 4.10, 4.0, 2.3 and 2.1),and RTCM 3.3 with communication interfaces over UART, SPI, I2C, and USB.

#### **3.1.4.2 NEO-F10N-00B**

The NEO-F10N-00B is another module from u-blox. It is a 49-channel module which is capable of concurrently receiving 3 GNSS constellations; GPS, Galileo, and BeiDou constellations. This receiver uses the L1/L5 bands. The L5 band is more accurate than other commonly used bands such as L1 and L2. The use of the L5 band along with the L1 band provides for a more optimal navigation/location experience. This receiver will use a SubMiniature version A (SMA) antenna via a SMA connector. This module has a 10 Hz max update rate with a multi GNSS configuration and 20 Hz max update rate with a single GNSS configuration. This module has an accuracy rating of about 1.5m. This module runs off 3.3 V and draws from 21mA - 26mA using the breakout board/circuit design from SparkFun. The supported protocols are UBX and NMEA( 2.1, 2.3, 4.0, 4.10 and 4.11). The only supported communication interface is UART.

#### **3.1.4.3 MAX-M10S**

The MAX-M10S GNSS module is yet another one from u-blox. This module can concurrently receive from 4 GNSS constellations; GPS, GLONASS, BeiDou, and Galileo. The required external antenna used will be a SMA antenna via a SMA connector. This module has a 10 Hz max update rate with a multi GNSS configuration and 18 Hz max update rate with a single GNSS configuration. This module has an accuracy rating of about 2.0 m CEP. The MAX-M10S runs off 3.3 V and draws about 6mA to 25mA depending on the number of constellations and tracking state. It is also capable of very low power consumption. This module will also be implemented using the breakout board/circuit design from SparkFun. The supported protocols for this module are UBX and NMEA(2.1, 2.3, 4.0, and 4.10) with communication over UART and I2C.

#### **3.1.4.4 L86-M33**

The L86 is a module from Quectel. It has 99 acquisition channels and 33 tracking channels. This module can concurrently receive from 4 constellations; GPS GLONASS, Galileo and QZSS. This module has an integrated patch antenna on top (POT) and has the MT3333 GNSS chipset from MediaTek. With the use of the breakout board from Soldered, there will also be an optional IPEX/U.FL antenna connection if the integrated antenna proves to be insufficient. The L86 has a max update rate of about 10 Hz (1 Hz default). The accuracy rating for this module is < 2.5 m CEP. This is a measure of the “circular error probable”. Meaning that 50% of the time the position reading will be within a 2.5 meter radius of the true position. With the use of the breakout board/circuit design from Soldered, this module will run off of 3.3 V. and draw about 7 uA - 30 mA depending on the number of constellations and tracking state. This module features a standby mode and backup mode which are very low power

consumption states. The supported protocols for this module are NMEA 4.0 and PMTK with these being implemented over the communication interface UART.

#### **3.1.4.5 ZED-F9P-05B**

The ZED-F9P-05B is a high precision GNSS module from u-blox. This module can concurrently receive the 4 major GNSS constellations; GPS, GLONASS, Galileo and BeiDou with the addition of SBAS and QZSS. To avoid an unnecessary use of excessive power, the module can be configured for a select group of constellations. The ZED-F9P-05B will need an active antenna with an integrated low noise amplifier(LNA). This will ensure the performance is maintained during normal operating signal reception. There is consideration of u-blox dual band antenna. This module has a max update rate of about 25 Hz and this varies with the number of constellations. This high precision module is capable of centimeter level accuracy with the use of real time kinematics(RTK), precise point positioning RTK (PPP\_RTK) and/or state space representation (SSR). This module will run on an input supply voltage of 2.7 V - 3.6 V with current draw of about 75 mA to 95 mA depending on constellations and tracking mode. Current peaks at 120 mA for GPS and 130 mA for multi constellations. The protocols supported are UBX, NMEA(4.11, 4.10, 4.0, 2.3 and 2.1), RTCM 3.4 and SPARTN 2.0.1. The communication interfaces supported are UART, SPI, I2C, and USB.

#### **3.1.4.6 NEO-6M**

The NEO-6M is a 50 channel high performance GPS module from u-blox. This module feature u-blox 6 position engine. This module uses the L1 frequency band for the GPS constellation. This module supports the use of an active or passive antenna. This module has a max update rate of 5 Hz. The NEO-6M has an accuracy rating of 2.5m with GPS (autonomous) and 2.0m with the use of SBAS. This module runs on 2.7 - 3.6 V and has a max current draw of 67mA with the average being 47mA for acquisition and 39mA for tracking (max performance), 37mA tracking (Eco mode) and 11mA tracking(power save mode). The protocols supported for this module are UBX, NMEA and RTCM. The supported communication interfaces are UART, USB, SPI, and DDC. The dimensions of this module are 16.0x12.2x2.4 mm<sup>3</sup>.

#### **3.1.4.7 CD-PA1616S**

The PA1616S GPS module by CDtop technology has 66 channels for searching and can track up to 22 satellites simultaneously. This module is built around the MT3339 GNSS chipset from MediaTek. This module has a built-in POT antenna and can support an external antenna. This module has an automatic antenna switching function and protection from short circuits. This module has Antenna Advisor, a system that aids in detections and notifications and various antenna statuses. This module uses the L1 frequency band. This module has a max

update rate of 10 Hz with a default update rate of 1 Hz. The position accuracy for this module falls around 3.0m CEP autonomously and 2.5m CEP using DGPS(SBAS(WAAS,EGNOS,MSAS)). The power required for this module is 3.0 V to 4.3 V. There is a current draw of about 25mA during acquisition and 20mA during tracking at 3.3 V and a 1 Hz update rate. The supported protocols are NMEA and RTCM. The communication interfaces are UART and I2C.

### 3.1.4.8 GNSS Modules Comparisons

**Table 3.5. GNSS Modules Comparisons.**

| Parameter                   | NEO-M9N-00B                     | Neo-F10N-00B                    | Max-M10S                          | L86-M33                         | ZED-F9P-05B                            | NEO-6M                          | CD-PA1616S                      |
|-----------------------------|---------------------------------|---------------------------------|-----------------------------------|---------------------------------|----------------------------------------|---------------------------------|---------------------------------|
| Accuracy (CEP)              | 2.0m                            | 1.0m                            | 2.0m                              | <2.5m                           | 1.5m                                   | <2.5m                           | 3.0m                            |
| Best (Assisted)             | 2.0m                            | 1.0m                            | 2.0m                              | <2.5m                           | 0.01m                                  | <2.0m                           | 2.5m                            |
| Multi-GNSS                  | GPS, GLONASS, Galileo, BeiDou   | GPS, Galileo, BeiDou            | GPS, GLONASS, Galileo, BeiDou     | GPS, GLONASS, Galileo, QZSS     | GPS, GLONASS, Galileo, BeiDou          | GPS                             | GPS, QZSS                       |
| Max Update Rate             | 25 Hz                           | 20 Hz                           | 18 Hz                             | 10 Hz                           | 8 Hz                                   | 5 Hz                            | 10 Hz                           |
| Power Consumption (Average) | 28mA Acquisit.<br>19mA Tracking | 26mA Acquisit.<br>27mA Tracking | 6.5mA Acquisit.<br>6.0mA Tracking | 26mA Acquisit.<br>22mA Tracking | 75mA Acquisit.<br>70 mA Tracking       | 47mA Acquisit.<br>39mA Tracking | 25mA Acquisit.<br>20mA Tracking |
| Voltage                     | 2.7 - 3.6 V                     | 2.7 - 3.6 V                     | 2.7 - 3.6 V                       | 3.0 - 4.3 V                     | 2.7 - 3.6 V                            | 3.1 - 5.5 V                     | 3.0 - 4.3 V                     |
| Protocols Supported         | UBX, NMEA 4.10, RTCM 3.3        | UBX, NMEA 4.11                  | UBX, NMEA 4.10                    | NMEA, PMTK                      | UBX, NMEA 4.11, RTCM 3.4, SPARTN 2.0.1 | UBX, NMEA 3.0                   | NMEA                            |
| Communication INT.          | UART, SPI, I2C, USB             | UART                            | UART, I2C                         | UART                            | UART, SPI, I2C, USB                    | UART, USB, SPI, DDC             | UART, I2C                       |
| Dimensions (mm)             | 16.0x12.2x2.4                   | 16.0x12.2x2.4                   | 10.1x9.7x2.5                      | 18.4x18.4x6.45                  | 21.8x16.8x2.2                          | 16x12.2x2.4                     | 16x16x5.2                       |
| Weight                      | 1.6g                            | 1.0g                            | 0.6g                              | 7.6g                            | 1.8g                                   | 1.6g                            | 4.0g                            |
| Price (USD)                 | \$27                            | \$49                            | \$21                              | \$15.29                         | \$129                                  | \$6.75                          | \$29.95                         |
| Breakout Board              | \$74.95                         | \$39.95                         | \$44.95                           | \$27.33                         | 249.95                                 | \$9.00                          | \$29.95                         |

### 3.1.5 Antenna

With the goal of the SmartNav Cane being capable of real-time location and navigation assistance, a suitable antenna will be required. Many factors play a role in the selection of this antenna. Considering the budget of this project, cost will be a major determining factor. In addition, the GNSS module will significantly contribute to this consideration. We want to choose an antenna that is comparable to the module's capabilities. If we are investing into a top performing module, then we'll want the antenna to match. Having a sub-par antenna could allow for error and introduce unwanted issues. Optimally, we would want the antenna facing the open sky without any obstructions. The durability of the antenna must also be considered. The antenna's durability is a critical aspect of its design and functionality. The intended environment and potential exposure to the elements must be considered. Will the antenna be mounted in a location with limited exposure to moisture and heat, or will it be subjected to harsher conditions? Given that this device is intended for outdoor use and could face direct sunlight, rain, and potentially other weather conditions, it's imperative to ensure that the antenna's performance isn't compromised due to these environmental factors. Moisture can lead to corrosion and electrical shorts, while excessive heat can damage the antenna's materials and affect its signal reception capabilities. Therefore, the antenna must be constructed with materials and protective measures that can withstand these environmental challenges and maintain its operational integrity over time.

We must also consider the frequency capabilities of the antenna, because it must match the frequency capabilities of the GNSS module used. We also don't want to choose a single band antenna if the intended purpose is to utilize a multi-constellation GNSS module and use multi-constellation mode. The different frequency bands include the L bands consisting of L1, L2, L3, L6, and L5. The L band is used by GPS, GLONASS and QZSS sharing center frequencies with other constellations like Galileo's E1, E6, and E5a and BeiDou's B1C, B2a band. For a multi-constellation mode of operation increasing the overall accuracy of location and navigation the antenna will need to be able to receive on the other constellations' frequencies bands.

The actual type of antenna will also need to be taken into consideration. Many modules have patch on top (POT) antennas that are built into the module but also come with the capability of adding an external antenna. These patch antenna designs typically require attachment to a metal ground plane for best performance. Also it's apparent that the placement of the antenna is limited. As it is, the antenna will reside where the module is. It must be determined if a patch antenna will be sufficient. There is also an external patch antenna. These antennas are typically attached to a PCB.

We will also need to consider whether we use an active antenna or a passive antenna. With an active antenna comes the benefit of amplification of a signal that has been degraded due to longer cable run or an non optimal placed antenna or operating conditions generating weaker signals, as would be our case, but this comes with the added use of power and because this device is battery powered, we wouldn't want to unnecessarily consume more battery. The type of antenna polarization must also be considered. GPS signals from the satellites broadcast right hand circularly polarized signals(RHCP), therefore for optimal reception of the signal we will want to assure the antenna is RHCP. If we were to choose a linearly polarized antenna we would stand the chance of significant signal loss due to the antenna's mounting orientation. We will also stand the chance of errors due to reflected signals and because linear polarized antennas can receive signals from various directions, this makes them more susceptible to the errors associated with these signals such as multipath errors.

The size of the antenna will play a crucial role in the selection process. Having an antenna that is too large could potentially become more of a hindrance. This could cause the antenna to be placed in a location that is not optimal for receiving a signal. This antenna could also become an obstruction for other sensors. There is a limited amount of space for the various components associated with the SmartNav Cane and having a small form factor yet highly precise antenna is ideal.

The type of connector used will also play a major role in the antenna selection process. The connector must be compatible with the connector on the PCB. Not only will the compatibility of the connector require consideration but the connector type itself. The U.FL, IPEX and other types of miniature connectors must be durable enough to withstand the normal operation of the cane. This is another point to consider when thinking about the mounting location. If the antenna wiring can potentially be moved, this may not be the best solution. The subminiature version(SMA) connectors are larger and more durable than the previously mentioned. With this larger sized connector also comes the consideration for the mounting location, potentially thicker wire size. All these factors must be taken into consideration with space being limited.

#### **3.1.5.1 ANN-MB-00**

The ANN-MB-00 is an active multi-band antenna from u-blox. This antenna supports the GPS, GLONASS, Galileo and BeiDou constellations. It is a RHCP antenna on all bands. This antenna has a supply voltage requirement of 3.0 - 5.0 V with a typical current draw of 15mA. This antenna has a weight of 178g with the dimensions of 82.0 x 60.0 x 22.5 mm<sup>3</sup>. The connector option type is a SMA. Other versions include a SMB or MCX connector. The cable length is 5m RG174. This antenna has a magnetic base for mounting or can be mounted using 2 x M4

screws. The operating and storage temperature is -40 - 80°C. It has an IP67 rating.

#### **3.1.5.2 Cirocomm**

The Cirocomm 5cm ceramic is an active multi-band patch antenna from Cirocomm Technology Corp. This antenna supports the GPS, GLONASS, and Galileo constellations. This antenna is RHCP. The weight of this antenna is 15g with the dimensions of 25x25x4mm<sup>3</sup> atop a PCB of dimensions 28x28x1mm<sup>3</sup>. This antenna uses an IPEX connector. The cable length is 50mm of 1.13mm coaxial. The operating and storage temperature is -40 - 85°C.

#### **3.1.5.2 ANT-GNCP-CA188L165**

The GNCP-CA188L165 is an active multi-band patch antenna from Linx Technologies. This antenna supports GPS, Galileo, GLONASS, BeiDou, IRNSS and QZSS. This antenna is RHCP. This antenna input voltage is 1.8 - 5.5 V with a typical voltage of 3.0 V. The current draw is typically 10.0mA at 3.0 V with a maximum of 15.0 mA. This antenna is mounted by soldering the LNA base to a PCB, a bracket, or by using an adhesive backing. The weight of this antenna is 7.4g with dimensions 18.0x18.0x7.1mm<sup>3</sup>. This antenna uses a U.FI connector. The cable length is 65mm of 1.13mm coaxial. The operating and storage temperature is -40 - 85°C.

#### **3.1.5.3 Adafruit GPS Antenna**

Adafruit offers a selection of compact passive patch GPS antennas that are suitable for lightweight and space-constrained applications like the SmartNav mobility stick. These antennas are designed specifically for GPS-only reception and are ideal for use with modules that do not require multi-constellation support. They are linearly polarized, which means their reception quality may be more sensitive to orientation and movement compared to circularly polarized antennas this is an important consideration when mounting on a mobile device like a walking aid, where the antenna may shift during motion. The antennas are mounted using adhesive backing, making installation straightforward and secure without requiring complex mechanical fixtures. Two size variants are available: one weighing 5.5 grams with dimensions of 15 mm × 15 mm × 6.8 mm, and a smaller version weighing 2.4 grams at 9 mm × 9 mm × 6.5 mm, offering flexibility depending on space constraints within the enclosure. Both variants feature a 50 mm long 1.13 mm coaxial cable terminated with a U.FL connector, allowing easy integration with most GPS breakout boards or embedded modules.

Although passive antennas do not include internal amplification like active GPS antennas, they are more power-efficient and ideal for systems that operate under line-of-sight conditions. Their compact size, low weight, and compatibility with U.FL connectors make them a practical choice for embedded GPS applications,

especially when the design prioritizes portability and simplicity. This would be a great potential candidate for the GPS module integration

### 3.1.5.4 Antenna Comparisons

**Table 3.6. Antenna Comparisons.**

| Parameter                      | ANN-MB-00                      | GNCP-CA188L16<br>5                         | Cirotech                | Adafruit-GPS-1          | Adafruit-GPS-2            |
|--------------------------------|--------------------------------|--------------------------------------------|-------------------------|-------------------------|---------------------------|
| Type                           | Patch                          | Patch                                      | Patch                   | Patch                   | Patch                     |
| Active/Passive                 | Active                         | Active                                     | Active                  | Passive                 | Passive                   |
| Multi-GNSS                     | GPS, GLONASS, Galileo, BeiDou  | GPS, GLONASS, Galileo, BeiDou, IRNSS, QZSS | GPS                     | GPS                     | GPS                       |
| Polarization                   | RHCP                           | RHCP                                       | RHCP                    | Linear                  | Linear                    |
| Gain<br>LNA Gain<br>Total Gain | 3.5 dBi<br>21.4 dB<br>24.9 dBi | 13.7 dBi                                   | 2 dBi<br>28 dB<br>32 dB | -2 dBi<br>-2 dBi        | 1 dBi<br>1 dBi            |
| Voltage                        | 3 - 5 V                        | 1.8 - 5.5 V                                | 3 - 5 V                 |                         |                           |
| Current                        | 15mA                           | 10mA typ. ,15mA max                        | 10mA                    |                         |                           |
| Weight                         | 178g                           | 7.4g                                       | 9.07g                   | 2.4g                    | 5.5g                      |
| Dimensions                     | 82x60x22 mm <sup>3</sup>       | 18x18x7.1 mm <sup>3</sup>                  | 28x28x5 mm <sup>3</sup> | 9x9x6.5 mm <sup>3</sup> | 15x15x6.8 mm <sup>3</sup> |
| Cable Length                   | 5m RG174                       | 65mm, 1.13mm coax                          | 50mm, 1.13mm coax       | 50mm, 1.13mm coax       | 50mm, 1.13mm coax         |
| Price(USD)                     | \$60                           | \$10.36                                    | \$6.28                  | \$4.95                  | \$3.95                    |

### 3.1.6 Audio Output

The audio output system for the SMARTNAV must fulfill various design criteria, as it is an important aspect to help produce the actual directions for the user. It should be small and light, while still being power efficient and producing clear, intelligible audio indicators. In portable devices, component footprint should be at a minimum. For basic alert signals or beeps, piezoelectric buzzers are an ultra-compact and very low power device that would work, but this is insufficient in the scope of our project. If a more detailed, lower-fidelity audio is required (for example, spoken direction), multiple small speakers with Class-D audio amplifiers can produce clear audio while maintaining a low power draw. Alternatively, surface-mount digital-to-analog converters (DACs) and integrated audio codecs can provide a clean solution, offering processing and audio output in a single, compact unit. All of these options have trade-offs in size, power consumption, and audio quality. The following sections aim to provide more

comprehensive details about the various options for our project, and which is the most suitable.

#### **3.1.6.1 Traditional Dynamic Speakers**

Traditional dynamic speakers are the most common audio transducers found in consumer electronics. They work on electromagnetic principles, where an alternating current drives a voice coil placed within a magnetic field, causing an attached diaphragm (or cone) to vibrate and generate sound waves. This technology has evolved over decades from bulky early models to the compact and efficient designs we see today, such as the UI Audio 3W 4Ω Mini Loudspeaker. These speakers deliver a broad frequency response and high-fidelity sound, making them excellent for clear voice prompts, music, and other multimedia outputs. For our SmartStick, dynamic speakers offer reliable and mature technology, capable of providing robust audio output while balancing size, power consumption, and sound quality. Their main trade-offs include a higher power draw compared to simpler alert devices and the need for careful acoustic enclosure design to optimize performance in a portable, outdoor setting.

#### **3.1.6.2 Piezoelectric Buzzers**

Piezoelectric buzzers utilize the piezoelectric effect where certain ceramic materials generate mechanical vibrations when subjected to an electrical voltage to produce sound. Typically, these devices consist of a piezo element bonded to a diaphragm that vibrates to emit a tone or series of beeps. Historically, piezo buzzers have been widely used in applications requiring simple auditory alerts, such as timers, alarms, and basic notification systems, because of their robustness, low power consumption, and compact form factor. In the context of the SmartStick, piezo buzzers could serve as a backup or supplementary audio alert mechanism, especially in conditions where minimal power usage and high durability are paramount. However, their limited frequency range and lower sound fidelity mean they are less suitable for high-quality audio output, making them best for simple, discrete notifications rather than rich audio feedback.

#### **3.1.6.3 Electrostatic Speakers**

Electrostatic speakers represent a more advanced and less conventional approach to audio reproduction. They function by suspending a thin, electrically charged diaphragm between two conductive plates; when an audio signal is applied, the resulting electrostatic forces move the diaphragm to produce sound. This technology is known for its exceptional clarity, low distortion, and a very flat frequency response, attributes that make it highly desirable for high-fidelity audio applications. Although electrostatic speakers have traditionally been used in high-end audio systems due to their complex drive requirements and need for

high-voltage circuitry, they offer a unique perspective when compared with conventional dynamic speakers and piezo buzzers. For the SmartStick, the potential integration of electrostatic speakers could deliver superior audio clarity for critical voice prompts and detailed auditory feedback. However, their higher power demands, more intricate amplifier designs, and delicate construction pose significant challenges for portable, rugged applications. As a result, while electrostatic speakers provide an intriguing option for ultimate audio performance, the practical constraints often make traditional dynamic speakers the more viable choice for our design.

### 3.1.6.4 Audio Output Technology Comparison

**Table 3.7. Audio Technology Comparison.**

| Parameter                         | Dynamic Speakers | Piezo Buzzers | Electrostatic Speakers |
|-----------------------------------|------------------|---------------|------------------------|
| Sound Quality (1-10)              | 8                | 5             | 9                      |
| Freq Range (Hz)                   | 20-20k           | 2k-4k         | 30-20k                 |
| Power Output (W)                  | 3W               | 0.5W          | 2W                     |
| Impedance ( $\Omega$ )            | 4 $\Omega$       | -             | -                      |
| Mechanism Code                    | 1 (EM)           | 2 (Piezo)     | 3 (Electrostatic)      |
| Cost (1=high, 10=low)             | 6                | 9             | 3                      |
| Size (1=large, 10=compact)        | 7                | 10            | 5                      |
| Complexity (1=simple, 10=complex) | 5                | 4             | 8                      |
| Efficiency (1-10)                 | 6                | 9             | 4                      |
| Noise (1-10, 10=low noise)        | 8                | 6             | 9                      |

Dynamic speakers are our clear choice for the SmartStick's audio output. Their electromagnetic design delivers high sound quality and a wide frequency range that is essential for clear, robust user feedback. While they require a bit more power compared to simpler alternatives like piezo buzzers, the proven performance and reliability of dynamic speakers exemplified by models such as the UI Audio 3W 4 $\Omega$  Mini Loudspeaker make them ideal for our portable design. Their balanced performance in terms of sound fidelity, cost, and integration complexity meets the demands of our application, ensuring that auditory cues are both effective and dependable.

### **3.1.7 Dynamic Audio Output Speakers**

Dynamic Audio Output Speakers play a pivotal role in the SmartNav design by providing immediate, clear auditory feedback essential for navigation and obstacle alerts. In this section, we explore a range of potential speaker options, evaluating each on criteria such as sound clarity, power efficiency, size, and integration compatibility with our amplifier circuitry. Our investigation considers factors like the optimal balance between output power and battery consumption.

#### **3.1.7.1 Adafruit Mini**

The Adafruit Mini 3W 4Ω Enclosed Speaker is engineered to reliably convert amplified electrical signals into clear acoustic output, making it an ideal component for a smart cane that delivers voice directions. At its core, the speaker operates via a miniature full-range driver: an alternating current applied to a coil creates a magnetic field that interacts with a permanent magnet, generating a Lorentz force ( $F = I \times L \times B$ ) that drives a lightweight diaphragm. This diaphragm's vibrations produce sound waves that are further enhanced by a meticulously tuned enclosure designed to resonate at specific frequencies, thus optimizing acoustic output while minimizing distortion. In quiet indoor environments, the speaker's well-balanced frequency response ensures that every word is reproduced with high fidelity. In noisier or outdoor settings, although external sounds and wind may slightly affect the perceived clarity, the high efficiency of the speaker when paired with a Class-D amplifier (like the MAX98357A) maintains sufficient sound pressure levels for intelligible voice commands. Moreover, environmental factors such as temperature and humidity can subtly influence the mechanical properties of the diaphragm and enclosure, potentially shifting resonance frequencies or affecting acoustic damping, but the robust design mitigates these effects to a large extent. Additionally, when mounted on a smart cane, the device's inherent vibrations and shocks are countered by secure enclosure design and optimized mounting strategies, ensuring that the speaker continues to deliver consistent, clear audio even under dynamic physical conditions.

#### **3.1.7.2 Dayton Audio CE Series**

Dayton Audio CE Series Mini Enclosed Speaker (Model RS-4) : This enclosure uses a full-range driver that has been constrained into a sealed box, which not only greatly increases overall clarity, but also improves bass response on the whole. Essentially, the driver works on the basis of electromagnetic induction: current passes through a coil, which produces a magnetic field that pushes on a permanent magnet to create a Lorentz force on the diaphragm as well, forcing it to vibrate and creating sound waves. The climate-controlled sealed enclosure is critical in that it controls the rearward sound wave, ensuring that

there is no phase cancellation and also enabling lower frequencies to build, for a full-bodied tonality.

Such a design is useful when specific voice directions need to be delivered on a smart cane as this wide frequency response gives a rich and clear output of the content delivered with all the nuances of speech. Indoor, quiet environments: sounds clear and articulate; even outdoors or in noisy conditions, its built-in efficiency and clarity brought about by the sealed design hold intelligibility when partnered with an appropriate Class-D amplifier. This meant that while the model itself may be a bit chunkier than your ordinary ultra-compact fare, its performance is very much reliant on being carefully partnered with an efficient amplifier design in order to achieve the right drive levels while consuming as little power as possible. The material and environmental factors, such as temperature variations and humidity, can also indirectly influence diaphragm response, but the solid build of the RS-4 and its proper mechanical setup within the smart cane counteract these influences, despite the design's high sensitivity, to maintain consistent performance in a diverse range of operating conditions..

### 3.1.7.3 PUI Audio

The PUI Audio 3W 4Ω Mini Loudspeaker is a high-performance, compact audio solution engineered specifically for portable, battery-powered devices like a smart cane. It operates using an electromechanical driver, where an alternating current drives a voice coil positioned in a magnetic field, generating a Lorentz force ( $F = I \times L \times B$ ) that moves a lightweight diaphragm to produce sound. This pre-enclosed design not only protects the sensitive components from environmental hazards but also helps finetune the acoustic response for clarity and efficiency. In quiet indoor settings, the loudspeaker delivers crisp, articulate voice output, ensuring that every word is easily understood. In noisier outdoor environments, its efficient energy conversion and focused diaphragm movement help maintain intelligibility despite ambient noise. However, as a passive device, it requires proper matching with a high-efficiency external amplifier to reach optimal drive levels without sacrificing power consumption an essential consideration for battery-operated systems. While its specialized design may result in a slightly higher cost compared to more common consumer models, the balance of compact size, clear audio reproduction, and power efficiency makes the PUI Audio 3W 4Ω Mini Loudspeaker an excellent choice for ensuring reliable and intelligible voice directions in your smart cane.

### 3.1.6.4 Audio Output Speakers Comparisons

**Table 3.8.** *Audio output Comparisons.*

| Parameter | Adafruit Mini 3W 4Ω Enclosed Speaker | Dayton Audio CE Series Mini Enclosed Speaker (RS-4) | PUI Audio 3W 4Ω Mini Loudspeaker |
|-----------|--------------------------------------|-----------------------------------------------------|----------------------------------|
|-----------|--------------------------------------|-----------------------------------------------------|----------------------------------|

|                          |           |           |           |
|--------------------------|-----------|-----------|-----------|
| Power Rating (W)         | 3         | 3         | 3.1       |
| Impedance ( $\Omega$ )   | 4         | 4         | 4         |
| Frequency Response (Hz)  | 200-20000 | 250-20000 | 300-18000 |
| Sensitivity (dB @ 1W/1m) | 85        | 87        | 83        |
| Dimensions (mm)          | 27x27x20  | 35x35x25  | 25x25x15  |
| Input Voltage (V)        | 5         | 5         | 3.3-5     |
| Built-in Amplifier       | No        | No        | No        |

After comparing our options, we have decided that the PUI Audio 3W 4 $\Omega$  Mini Loudspeaker will be the best option for the SmartStick. It is specifically tailored for small, battery-powered devices and ensures that the voice output is both crisp and clear, while being power efficient, a vital parameter for handheld applications. It does need to be carefully matched to a high-efficiency amplifier, and it is a little more expensive than more ubiquitous consumer models, but its performance is tailored for low-power, high-clarity audio, and we've found it to be the perfect solution for reliably delivering intelligible voice directions as part of our design.

### 3.1.7 Audio Amplification

Efficiently amplifying sound is essential to allow our compact speakers to deliver clarity while preserving battery life. Some of the best universal audio amplifiers are the MAX98357A, TPA3118, and PAM8302A. The MAX98357A provides a Class-D digital gain stage that takes the I<sup>2</sup>S directly from the ESP32 and has high efficiency, with minimal power loss, with output of about 3W into a 4 $\Omega$  load; also, its low quiescent current and small footprint make it suitable for battery-operated devices. By contrast, while the TPA3118 can provide higher output levels with plenty of power on tap, it is seen as overkill in our low-power, compact use case and the PAM8302A, though compact and efficient in low-power mode, simply does not deliver the range and clarity we needed for crisp voice directions. Considering our needs for a small, efficient amplifier that closely matches with our speakers and helps to satisfy the low-power constraints of a portable smart cane, the MAX98357A is the clear winner it strikes the best balance between performance and simplicity of integration, and energy efficiency.

### **3.1.8 Microphones**

In selecting the optimal microphone, several factors including noise rejection, power efficiency, compact size, and ease of integration with our ESP32 system are critical. MEMS microphones, which convert sound pressure waves into electrical signals via a vibrating diaphragm coupled with a capacitive sensing element, are the leading candidates for such applications. Among the top options are the INMP441, SPH0645LM4H-B, and ICS-43434. The INMP441 offers robust performance with a balanced sensitivity and integrated digital I<sup>2</sup>S output, making it versatile but sometimes demanding careful power supply filtering to avoid interference.

#### **3.1.8.1 Digital MEMS Microphones**

Digital MEMS microphones use microelectromechanical systems to convert sound into electrical signals through a capacitive sensing mechanism. In these devices, a tiny vibrating diaphragm moves relative to a fixed backplate, creating changes in capacitance that are immediately digitized within the chip. This technology exemplified by the INMP441, SPH0645LM4H-B, and ICS-43434 delivers a direct digital output (typically via an I<sup>2</sup>S interface) that simplifies integration with microcontrollers like the ESP32. Their digital nature minimizes external interference, ensuring a clean signal with a typical frequency response from 20 Hz to 20 kHz and low power consumption (usually around 0.5–0.6 mA). The compact size (approximately 3×3×1 mm) and high sensitivity make them ideal for portable devices such as the SmartStick. Despite their many benefits, digital MEMS microphones require careful power supply filtering and clock management to avoid digital artifacts, but overall, their integration ease and robust performance make them the prime choice for our application.

#### **3.1.8.2 Analog MEMS Microphones**

Analog MEMS microphones also rely on microelectromechanical systems with a capacitive element to capture sound, yet they differ by providing an analog voltage output rather than digitizing the signal internally. These devices offer the advantage of a customizable signal path, allowing designers to tailor amplification and filtering circuits according to the specific acoustic requirements of the application. With performance characteristics comparable to their digital counterparts such as a similar frequency range and compact size analog MEMS microphones can be especially useful when high-end, bespoke audio processing is needed. However, because they output an analog signal, they are more vulnerable to external noise and require an additional analog-to-digital conversion stage if interfaced with a digital system like the ESP32. This extra circuitry not only adds complexity but may also increase power consumption and susceptibility to interference, which are critical considerations in the SmartStick's portable, battery-powered design.

### 3.1.8.3 Electret Condenser Microphones (ECMs)

Electret Condenser Microphones (ECMs) represent a traditional microphone technology that has been in use for decades. Unlike MEMS microphones, ECMs employ a permanently charged electret material in conjunction with a sensitive diaphragm to convert sound pressure into an analog electrical signal. Their operation typically requires an external bias voltage and amplification circuit to achieve the desired output level. ECMs are known for their excellent sensitivity and cost-effectiveness, and they have been widely used in a range of consumer electronics. However, their larger physical size, higher power requirements, and the need for additional circuitry can make them less attractive for modern, ultra-compact applications like the SmartStick. Although ECMs can deliver high-quality audio, the additional complexity of integration and potential for increased noise levels mean that, for our design, MEMS-based solutions particularly digital MEMS microphones are generally more suitable given the size, power, and integration advantages they offer.

### 3.1.8.4 Microphone Technology Comparisons

**Table 3.9. Microphone Technology Comparisons.**

| Parameter              | Digital MEMS Microphones     | Analog MEMS Microphones | Electret Condenser Microphones |
|------------------------|------------------------------|-------------------------|--------------------------------|
| Operating Principle    | Capacitive, I <sup>2</sup> S | Capacitive, Analog      | Electret, Analog               |
| Signal Output          | Digital (I <sup>2</sup> S)   | Analog                  | Analog                         |
| Frequency Range (Hz)   | 20-20k                       | 20-20k                  | 20-20k                         |
| Power Consumption      | ~0.5 mA                      | ~1 mA                   | ~1 mA+                         |
| Size                   | ~3×3×1 mm                    | ~3×3×1 mm               | Larger                         |
| Integration Complexity | Low                          | Moderate                | High                           |
| Noise Performance      | Low                          | Moderate                | Moderate                       |
| Sensitivity            | ~60-65 dB                    | ~60-65 dB               | ~55-60 dB                      |

|      |          |          |     |
|------|----------|----------|-----|
| Cost | Moderate | Moderate | Low |
|------|----------|----------|-----|

Digital MEMS microphones are our preferred choice for the SmartStick design due to their direct digital output and ease of integration with the ESP32 system. Their capacitive sensing mechanism, combined with an I<sup>2</sup>S interface, ensures a clean, low-noise signal that minimizes the need for extensive external filtering. Additionally, their compact size and low power consumption make them ideally suited for portable, battery-operated devices, while offering robust performance and excellent sensitivity for voice command recognition.

### 3.1.9 Digital MEMS Microphones

In our SmartNav design, extensive research into digital MEMS microphones is instrumental in selecting the optimal component for high-fidelity audio capture. Digital MEMS microphones offer exceptional benefits including compact size, low power consumption, and high signal-to-noise ratios attributes that are essential for reliable voice command input and effective obstacle detection in dynamic environments. Their integrated analog-to-digital conversion and robust digital interfaces simplify system integration by reducing the need for external circuitry, thus streamlining the overall design. This focused research has helped us ensure that the chosen microphone meets the performance and efficiency requirements critical to the SmartNav's real-time navigation capabilities

#### 3.1.9.1 INMP441

In the SmartNav design, the INMP441 emerges as a robust and versatile digital MEMS microphone that simplifies audio capture by delivering a clean I<sup>2</sup>S digital output directly to the ESP32. This microphone features a frequency response from 20 Hz to 20 kHz and a signal-to-noise ratio of approximately 60 dB, ensuring that both ambient sounds and user voice commands are captured with sufficient clarity for navigation purposes. Operating at 3.3 V with a typical current consumption around 0.6 mA, the INMP441 is highly efficient for battery-operated devices. Its compact package, roughly 3x3x1 mm, allows it to be easily integrated into the limited space of a smart cane without compromising performance. However, its omnidirectional pickup pattern, while beneficial for capturing commands from various angles, can also introduce ambient noise, which may require additional signal processing or noise cancellation techniques. Overall, the extensive datasheet support and widespread adoption of the INMP441 make it a reliable candidate for ensuring accurate voice command recognition in our SmartNav system.

#### 3.1.9.2 SPH0645LM4H-B

One such alternative is the SPH0645LM4H-B, which is a superior acoustic performer and energy-efficient, making it an ideal solution. With a signal-to-noise ratio reaching 65dB and a frequency response finely adjusted to match the human voice, this microphone ensures an accurate and clear capture of voice commands like navigation directions. The SPH0645LM4H-B operates at 1.8V to 3.3V and draws around 0.5mA so is well-suited for small, battery-powered products. Its ultra-small form factor (around 3x3x1mm) is conducive to integration into the space-limited housing of a smart cane. High sensitivity is important for the microphone, but can in some cases also lead to capturing unwanted background noise, which can be avoided through the proper positioning and digital filtering. Packed into this compact sound sensor, supported with expansive datasheets and various demo kit materials, is a powerful solution that encourages both optimization and comprehension; SPH0645LM4H-B presents itself as a solid candidate for embedded applications targeting voice navigation.

### 3.1.9.3 ICS-43434

The ICS-43434 is a high-performance digital MEMS microphone designed to deliver reliable audio capture in demanding environments, making it a noteworthy option for the SmartNav system. Featuring a digital I<sup>2</sup>S interface, it converts sound into electrical signals with a frequency response ideally suited for the human voice, ensuring that spoken commands are accurately captured even in the presence of background noise. With a sensitivity of around –26 dB and an SNR typically in the range of 60 to 64 dB, the ICS-43434 provides a balanced output that supports both speech recognition and ambient sound monitoring. Its operating voltage of 1.8V to 3.3V, coupled with low current consumption, aligns well with the power constraints of portable, battery-operated devices. Although the ICS-43434 may lack the extensive community support found with alternatives like the SPH0645LM4H-B, its robust performance and compact design (comparable to 3x3x1 mm dimensions) make it a reliable option for precision voice command capture in a smart cane. Its slightly more challenging integration process is offset by its solid performance in terms of clarity and durability, ensuring accurate user guidance in a SmartNav environment.

### 3.1.9.4 Digital MEMS Microphones Comparison

**Table 3.10.** *Microphone Digital MEMS Comparisons.*

| Parameter               | INMP441      | SPH0645LM4H-B                  | ICS-43434                      |
|-------------------------|--------------|--------------------------------|--------------------------------|
| Frequency Response (Hz) | 20Hz - 20kHz | 20Hz - 20kHz (voice optimized) | 20Hz - 20kHz (voice optimized) |

|                            |                                                   |                                      |                                                     |
|----------------------------|---------------------------------------------------|--------------------------------------|-----------------------------------------------------|
| Signal-to-Noise Ratio (dB) | ~60 dB                                            | 65 dB                                | 60-64 dB                                            |
| Operating Voltage (V)      | 3.3 V                                             | 1.8V - 3.3V                          | 1.8V - 3.3V                                         |
| Current Consumption (mA)   | ~0.6 mA                                           | ~0.5 mA                              | ~0.5 mA                                             |
| Package Dimensions         | ~3x3x1 mm                                         | ~3x3x1 mm                            | ~3x3x1 mm                                           |
| Interface                  | I <sup>2</sup> S                                  | I <sup>2</sup> S                     | I <sup>2</sup> S                                    |
| Pickup Pattern             | Omnidirectional                                   | Omnidirectional                      | Omnidirectional                                     |
| Additional Comments        | Robust, well-documented; may need power filtering | Superior acoustics, energy-efficient | Reliable and robust; integration can be challenging |

The INMP441 is our chosen digital MEMS microphone for the SmartNav design due to its robust performance and ease of integration. Its clean I<sup>2</sup>S digital output, wide frequency response (20 Hz to 20 kHz), and high signal-to-noise ratio (~60 dB) ensure that voice commands and ambient sounds are captured accurately. Additionally, its low power consumption (around 0.6 mA) and compact 3x3x1 mm package make it ideal for battery-operated, space-limited applications like our smart cane. Overall, extensive documentation and proven reliability make the INMP441 a solid choice for achieving precise audio capture in our system.

### 3.1.10 Voltage Regulators

When designing our Smartstick, voltage regulation is a crucial factor especially in applications where precision and reliability matter. In the SmartStick design, the smart stick we have muzzle load and both lower in high speed, so over voltage can danger for bare chip MCU. Voltage Regulators: Voltage regulators are the guardians of our circuit, ensuring that voltage levels from the power supply are smoothed out and that our components are protected from overvoltage or under-voltage conditions that could result in failure or permanent component damage. We explored how these devices operate on a principle of stability of output voltage under varying conditions of input supply or load conditions. Such regulation is critical for the operation of modern electronics; a small change in power can heavily affect performance and life expectancy.

Voltage regulators will play a crucial role in improving the overall robustness of our setup by making sure that the MCU and other essential elements get a stable voltage. The talk will cover some design considerations that affect the choice and configuration of voltage regulation approaches. We explored critical areas like efficiency, thermal management, and noise suppression. Where these trade-offs often come into play as we seek to optimize performance. We found the application of voltage regulators in the context of overall power management schemes, emphasizing their importance in protecting sensitive circuitry and in reducing power consumption where possible, which is especially important in battery and portable applications. The voltage conversion from the power source to that required by the processor will be done using a regulator, which we'll need to find that will accept the voltage we started with (in general some 5V source) and output the desired 3.3V. In battery powered or energy sensitive applications, high efficiency becomes a design imperative, and a low dropout regulator (LDO), which outputs the minimum possible voltage above input, exists to minimize the voltage differential between supply and output so as to minimize power dissipation as heat. As the combination of the voltage differential and the load current generates a power dissipation which needs to be dissipated, the efficiency of the regulator has a direct impact on the overall system performance. For digital rails, apart from efficiency the regulator has to be low noise and low ripple because digital signals are time sensitive and ripple of the regulator can introduce errors into the system. From a discipline-specific perspective, they are rules of thumb based on Ohm's law and energy loss: design is substantially tied to the conversion of electrical power into thermal energy; the required conversion must be balanced with the inevitable heat given off through reliable thermal management techniques like integrated thermal shutdown and limits on current flow. For a deep dive into particular voltage regulation solutions and configurations. 21.2 THT Explained: The voltage on THT is very important for the proper operation of the system and it needs to be maintained below 5.5V and we have already seen that the 2-lead input MDO5 is a good choice to achieve this.

### **3.1.10.1 Linear Voltage Regulators**

Linear voltage regulators have been a cornerstone of electronic power management since the early days of integrated circuit development. These devices operate on the fundamental principle of maintaining a steady output voltage by dissipating the excess input voltage as heat. In practical terms, when a linear regulator such as those in the AMS1117 or LM1117 series is used, the difference between the higher input voltage (typically around 5V in our SmartStick design) and the required output (3.3V) is dropped across an internal transistor or series pass element. This inherent simplicity results in extremely low noise and a stable, ripple-free output, which is ideal for powering sensitive microcontrollers and analog circuitry. Over the decades, linear regulators have earned a reputation for reliability and ease of use, thanks to their minimal external component requirements and straightforward design considerations. However, the very characteristic that makes them simple namely, their method of

dissipating excess energy as heat also renders them less efficient in situations where the voltage difference is significant. In our SmartStick, where battery life and thermal management are critical, the high dropout voltage and resultant heat generation can pose challenges, especially under high load conditions or in compact, portable designs. Despite these limitations, linear regulators remain highly attractive for specific portions of the SmartStick circuit that demand a clean, noise-free supply, making them a viable option when efficiency is not the paramount concern and when thermal design measures, such as heat sinks or proper PCB layout, can mitigate their drawbacks.

#### **3.1.10.2 Switching Voltage Regulators**

Switching voltage regulators represent a significant evolution in power conversion technology, offering a high-efficiency alternative to linear regulators. Unlike linear devices that simply burn off excess voltage, switching regulators, or DC-DC converters, use high-frequency switching elements typically transistors that rapidly turn on and off to transfer energy between the input and output. This process, which often involves pulse-width modulation (PWM), uses inductors and capacitors as energy storage elements to smooth out the output voltage, achieving conversion efficiencies that can exceed 90%. The physics underlying this method relies on the controlled charging and discharging of inductors, which allows the regulator to step down voltage with minimal energy loss. Historically, as electronics evolved toward portable, battery-operated devices, the need for more efficient power regulation spurred the development of switching regulators. For our SmartStick design, which is powered by a battery and must maintain a steady 3.3V for sensitive components, the high efficiency of switching regulators translates into prolonged battery life and reduced thermal stress. However, this approach comes with its own set of challenges; the rapid switching can introduce electromagnetic interference (EMI) and output voltage ripple, necessitating careful PCB layout, the use of filtering components, and sometimes even additional shielding to protect sensitive circuitry. Despite the increased complexity and the need for more external components (such as inductors, diodes, and capacitors), the benefits of high efficiency and lower heat generation make switching regulators an attractive choice for parts of the SmartStick where power efficiency is critical and where the design can accommodate the additional complexity and potential noise mitigation measures.

#### **3.1.10.3 Charge Pump Voltage Regulators**

Charge pump voltage regulators offer an innovative and compact solution for voltage conversion, especially in scenarios where current requirements are relatively low and board space is at a premium. Unlike both linear and switching regulators, charge pumps use capacitors as energy storage elements to transfer charge from the input to the output through a series of controlled switches. This method, often referred to as switched-capacitor regulation, eliminates the need

for bulky inductors, making these regulators particularly attractive for designs where component footprint and electromagnetic interference must be minimized. The underlying physics of charge pumps involves the periodic charging and discharging of capacitors to effectively double, invert, or otherwise adjust the voltage level, and while this method can provide decent efficiency for low-current applications, it tends to struggle with higher load demands. Historically, charge pumps have been utilized in integrated circuits where space-saving and low noise are more critical than delivering high current. In the context of the SmartStick, charge pump regulators could be employed in auxiliary or signal-conditioning circuits where the current draw is minimal and the priority is on maintaining a stable voltage with low ripple. Their advantages include low electromagnetic interference and a reduced component count, which simplifies the overall design and can be beneficial for reducing system cost and board size. However, their limitations in current handling and efficiency under heavy load conditions mean that charge pumps are best suited for powering non-critical sections of the circuit or for providing backup regulation in a multi-tiered power management system. By carefully evaluating the current requirements and operational demands of each subsystem within the SmartStick, we can strategically incorporate charge pump regulators to complement the primary power regulation strategy while optimizing both performance and footprint.

#### 3.1.10.4 Voltage Regulators Comparison

**Table 3.11. Voltage Regulators Comparison.**

| Parameter          | Linear Voltage Regulators | Switching Voltage Regulators | Charge Pump Voltage Regulators |
|--------------------|---------------------------|------------------------------|--------------------------------|
| Type               | Linear                    | Switching                    | Charge Pump                    |
| Principle          | Heat drop                 | PWM/Inductor                 | Capacitor switching            |
| Efficiency         | Low                       | High                         | Med (low load)                 |
| Dropout            | High                      | Low                          | Low                            |
| Noise              | Low                       | High                         | Med                            |
| Thermal            | High heat                 | Low heat                     | Low heat                       |
| Current Capability | ~1A                       | High                         | Low                            |
| Size               | Small                     | Larger                       | Tiny                           |
| Accuracy           | Stable                    | Mod ripple                   | Stable (low load)              |
| Voltage Range      | Limited                   | Wide                         | Limited                        |
| Ripple             | Minimal                   | Some                         | Moderate                       |

|                |                 |              |                    |
|----------------|-----------------|--------------|--------------------|
| Control        | Basic           | Advanced     | Basic              |
| History        | Old tech        | Modern       | Niche              |
| Applications   | Noise-sensitive | Battery apps | Low-power circuits |
| Cost           | Low             | Med/High     | Low                |
| Load Variation | Varies          | Stable       | Low load only      |
| Stability      | Clean           | Filtered     | Stable (low load)  |

After careful evaluation of the available voltage regulation technologies, we have chosen linear voltage regulators for our SmartStick design. Their ability to deliver a stable, low-noise power supply is crucial for the sensitive microcontrollers and analog circuits in our device. Although they are less efficient than switching regulators, their simplicity, proven reliability, and minimal electromagnetic interference make them the best fit for our low-power, noise-sensitive application. With proper thermal management in place, the advantages of linear regulators clearly outweigh their drawbacks for our design.

### 3.1.11 Linear Voltage Regulators

#### 3.1.11.1 AMS1117-3.3

The AMS1117-3.3 (ASM 711, widely used linear Voltage regulator with 3.3V on typical 5V input (support current load to 1A) linear voltage regulator - drop-out voltage, the difference between input voltage and output voltage - this different voltage should be high enough for maintaining line regulation. The regulator is out there, plenty of documentation with it that makes it easy to integrate into all sorts of designs. Together with its great performance and affordability, this is what makes the AMS1117-3.3 has become a common preference for many engineers who want a simple way to regulate voltage and why we would also consider it for our system design. While there are numerous benefits to having this approach, there are also a few caveats. The regulator features a relatively high dropout voltage that could decrease efficiency in battery-powered or energy-sensitive applications. Now, the added dropout can mean that more of the power has to be "wasted" as heat, especially at higher current loads, and that can affect the thermal aspects of the whole system. So, the AMS1117-3.3 While the TTL2603 3 still delivers a rock-solid 3.3V on cheap 3 wires in 3 as poor voltage dropping cases, it should be considered carefully based on the application needs.

**Table 3.12.** Absolute Maximum Ratings Digikey.

| Characteristics                             |            | Symbol            | Rating             | Unit |
|---------------------------------------------|------------|-------------------|--------------------|------|
| Input Supply Voltage                        |            | V <sub>in</sub>   | 18                 | V    |
| Lead Temperature                            |            | T <sub>lead</sub> | 260                | °C   |
| Operating Junction Temperature Range        |            | T <sub>j</sub>    | 150                | °C   |
| Short Temperature Range                     |            | T <sub>stg</sub>  | -65 ~ + 150        | °C   |
| Power Dissipation                           |            | P <sub>d</sub>    | Internally Limited | mW   |
| Thermal Resistance from Junction to Ambient | SOT-223-3L | R <sub>ja</sub>   | 88                 | °C/W |
|                                             | TO-252-2L  |                   | 86                 |      |
|                                             | SOT-89-3L  |                   | 102                |      |
| ESD Tolerance                               |            | ESD               | 2000               | V    |

These absolute maximum ratings are very important as we will use the information to help us choose the best voltage regulator for our SmartStick. They define the safety perimeter within which the regulator needs to operate if it is not to fail. For example, the 18V input supply voltage rating tells us that this device can withstand voltages up to this range without risk of damage, certainly allowing for far less voltage when operating normally. Lead temperature and junction operating temperature range (up to 150°C) are essential during soldering and regular use indicates the thermal stresses the device can handle. It is also important to notice the thermal resistance values available for different packages, such as 88°C/W for the SOT-223-3L, 86°C/W for the TO-252-2L and 102°C/W for the SOT-89-3L, as these can impact the decision-making process when considering thermal considerations. They allow us to use the equation  $P_{D(max)} = (T_{J(max)} - T_{amb})/R_{ja}$  to determine the maximum power this regulator can safely handle when under load without overheating.

### 3.1.11.2 MIC5365-3.3

The MIC5365-3.3 regulator is a compelling candidate for our SmartStick design because it provides a stable 3.3V output with a very low typical dropout voltage, 150–300mV. This low dropout performance is especially advantageous for the SmartStick, where battery energy is paramount and the input voltage may only slightly exceed the needed 3.3V. Also, the enable pin makes routing power considerably flexible, letting us perform power sequencing and management based on the SmartStick's operational states. The MIC5365-3.3 typically allow

for lower maximum output currents than something like AMS1117. While the SmartStick should work for many peripheral designs, this limitation may need to be taken into account if the SmartStick is to drive a high-current peripheral. Overall, the MIC5365-3.3 that serves to have a reliable 3.3V supply while optimizing power (i.e. duty cycle from project's perspective with portable battery-operated SmartStick) the best of traits included in traditional regulators

Additionally, the enable pin on the MIC5365-3.3 adds considerable flexibility to our power management strategy. It allows us to perform precise power sequencing and management tailored to the SmartStick's various operational states, which is critical for optimizing energy consumption in a portable device. However, it is important to note that this regulator typically supports lower maximum output currents compared to options like the AMS1117. While this is sufficient for many peripheral designs, it may need to be considered carefully if the SmartStick is intended to drive high-current components. Overall, the MIC5365-3.3 delivers a reliable 3.3 V supply while optimizing power usage, making it an ideal fit.

### **3.1.11.3 LM1117-3.3**

The LM1117-3.3 is a popular linear voltage regulator that has been designed to output a steady 3.3V while providing moderate to high current outputs, usually between 800mA to 1A. This feature is vital for our SmartStick design, as it enables primary components, such as the microcontrollers and peripherals, to receive a reliable and stable power source. It has been used extensively by countless designs, and thus has developed an excellent reputation for reliability and robustness for applications where current may vary or peak during operation. Nonetheless, the LM1117-3.3 has a trade-off for this gives a slightly higher dropout voltage like the AMS1117 series at 3.3V. This implies that in order for the regulated output to persist at 3.3V, the input voltage needs to be considerably greater than the output. If the available input voltage is only slightly greater than 3.3V, which can happen in battery-powered equipment, the regulator can be wasteful. This excess voltage is in a sense 'wasted' as heat, which can result in thermal management issues, mainly in heavy load conditions. This feature, however, makes it important that the input supply is critically evaluated to ensure that the supply overhead voltage is enough for adequate performance but not to create any adverse heating in our SmartStick which is portable and energy efficient.

The LM1117-3.3. As we integrate 3.3 into our SmartStick project, we must understand not only its general current-holding capability, but also the inherent dropout voltage limits the device exhibits. This proven track record makes the regulator a strong candidate for applications that prioritize reliability and smooth power delivery. But if our design context includes cases when the input voltage is marginal or where power efficiency is vital such as prolonged battery-operated situations we may need to consider other design features or different regulators

with lower dropout performance. In summary, although the LM1117-3. While 3 provides a clean answer to generate a steady 3.3V output, whether we will be implementing 3 in our SmartStick ultimately depends on whether the entire power path can compensate for its dropout voltage whilst balancing performance with thermal/efficiency requirements.

### 3.1.11.3 MIC29302-3.3

The MIC29302 is a robust, high-current low-dropout regulator capable of delivering up to 3 A with a dropout of about 340 mV at full load, built-in current limiting and thermal shutdown for protection, and a quiescent current under 200  $\mu$ A. In our smart-cane system, we'd use the MIC29302 to step the Li-ion battery's 4.2 V-to-3.0 V range down to a rock-solid 3.3 V rail, ensuring the ESP32, VL53L1X ToF sensor, Sharp IR module, and audio amplifier all get clean, noise-free power crucial for reliable distance measurements and crisp audio feedback in varying environmental conditions.

### 3.1.11.4 Linear Voltage Regulators Comparisons

**Table 3.13.** *Linear Voltage Regulators Comparisons.*

| Feature             | AMS1117-3.3    | MIC5365-3.3          | LM1117-3.3     | MIC29302-3.3 |
|---------------------|----------------|----------------------|----------------|--------------|
| Type                | Linear         | Linear               | Linear         | Linear       |
| Output Voltage      | 3.3 V          | 3.3 V                | 3.3 V          | 3.3 V        |
| Maximum Current     | 1 A            | <1A                  | 800mA-1A       | 1A           |
| Input Voltage Range | Up to 18V      | 3.5–5.5 V            | 4.75 V - 15 V  | 1.5 - 3.8V   |
| Drop out Voltage    | Moderate       | Very low (150-300mV) | Moderate       | Low          |
| Efficiency          | Lower (linear) | Higher than AMS1117  | Lower (linear) | High         |
| Thermal Resistance  | 88°C/W         | 250 °C/W             | 65 to 90 °C/W  | 300°C/W      |
| Heat Dissipation    | Higher         | Moderate             | Higher         | Low          |

|                              |                    |                    |                    |     |
|------------------------------|--------------------|--------------------|--------------------|-----|
| External Components Required | Few (2 capacitors) | Few (2 capacitors) | Few (2 capacitors) | Few |
| PCB Layout Complexity        | Low                | Low                | Low                | Low |

After evaluating all options, the MIC5365-3.3 and the MIC29302-3.3 are the best choices for our SmartStick due to their low dropout voltage, high efficiency, and built-in enable pin for flexible power management. Its low power consumption and compact design make it ideal for our battery-operated, portable application, providing a stable 3.3V output with minimal heat generation meeting both performance and energy efficiency needs.

### 3.1.12 DC-DC Converter

A DC-DC converter is a class of power electronics used to pass current from one direct current voltage level to set another level. These converters are crucial in our SmartStick design as they enable us to convert the output power from a small, lightweight battery, generally around 3.7V, to meet the diverse voltage requirements of the various components within the device. When the input voltage is less than the desired output voltage, boost converters are important (LED drivers are typically between 24V and 60V, hence boost converters help to convert to high frequency regions). A boost converter operates by rapidly switching a transistor on and off, during the “on” portion, an inductor stores energy, during the “off” portion it releases that energy to the load at a greater voltage. This is finely controlled to provide continuous and regulated output as we can then boost the battery’s 3.7V output up to 5V for those devices and components needing a higher voltage to operate. The performance of this method is essential in our SmartStick, as it improves battery life and reduces the overall weight and size of the system. The inclusion of a boost converter ensures that we can meet the voltage thresholds required by our sensors, microcontrollers, and other peripherals yet preserve the energy-efficient and reliable operation crucial for a portable, battery-operated assistive device.

#### 3.1.12.1 DC-DC Boost Converter Technology

DC-DC Boost Converter modules use high-frequency switching along with inductors and capacitors to step up a lower input voltage to a higher output level. These converters are designed to accept a wide range of input voltages typically from as low as 2V up to 24V making them flexible for battery-powered applications. Their efficiency is generally high (often between 80% and 93% under moderate load conditions), and many designs allow for fine adjustment of the output voltage via a built-in potentiometer. However, because they operate by

rapidly switching, they inherently produce voltage ripple and electromagnetic interference (EMI), which may require additional filtering for sensitive circuits. Furthermore, while their maximum rated output current might be high (around 1–2A), the practical continuous current under steady loads is often lower. Overall, boost converters offer excellent adaptability and efficiency, but their noise and current limitations must be carefully managed in portable designs like our SmartStick.

### 3.1.12.2 Step-Up Voltage Regulator Technology

Step-up Voltage Regulators are engineered to provide a stable and precise output voltage, such as a constant 5V, across a narrower but sufficient input range (commonly around 2.5V to 10V). Unlike generic boost converters, these regulators incorporate advanced protection features such as short-circuit protection, thermal shutdown, and current limiting, ensuring robust performance even under abnormal conditions. Although they typically offer less output adjustability, their focus on safety and consistent voltage delivery makes them ideal for applications where circuit integrity and minimal noise are critical. The trade-offs include a higher cost and slightly less flexibility in adjusting the output voltage. Nonetheless, the enhanced stability and built-in protection of step-up voltage regulators make them a compelling choice for powering delicate components in systems like our SmartStick.

### 3.1.12.3 Boost Technology Converter Comparison Table

**Table 3.14.** *Boost Converter Comparison Table.*

| Parameter                 | Boost Converter Module       | Step-Up Voltage Regulator                  |
|---------------------------|------------------------------|--------------------------------------------|
| Efficiency                | 80% – 93%                    | High, stable                               |
| Input Voltage Range       | 2V – 24V                     | 2.5V – 10V                                 |
| Output Voltage            | Adjustable (e.g., 5V)        | Fixed (e.g., 5V)                           |
| Noise/Ripple              | Higher (switching noise)     | Lower (filtered output)                    |
| Adjustability             | High (potentiometer control) | Limited adjustability                      |
| Protection Features       | Basic; minimal protection    | Advanced (short-circuit, thermal shutdown) |
| Continuous Output Current | ~1–2A (practical lower)      | Consistent, model-dependent                |
| Cost                      | Lower cost                   | Higher cost                                |

|                  |                                     |                                              |
|------------------|-------------------------------------|----------------------------------------------|
| Size/Form Factor | Compact                             | Compact & robust                             |
| Complexity       | Moderate (may need extra filtering) | Simple integration with built-in protections |

For the SmartStick, we chose a step-up voltage regulator because it offers a clean, stable output with robust safety features that are crucial for a portable, battery-powered device. While boost converters provide high efficiency and adjustability, their inherent switching noise and current limitations can compromise sensitive circuitry. The step-up voltage regulator built-in protections such as short-circuit and thermal shutdown ensure reliable performance even under abnormal conditions, making it the more suitable choice despite a slightly higher cost and reduced adjustability.

### 3.1.13 Step-Up Voltage Regulators

#### 3.1.13.1 Pololu 5V Step-Up Voltage Regulator

The Pololu 5V Step-Up Voltage Regulator is a dedicated solution engineered to boost lower battery voltages, such as from a 3.7V LiPo, to a stable and fixed 5V output. Operating over an input range of approximately 2.5V to 10V, it is designed for robust, portable applications. Notably, it features advanced protection mechanisms including short-circuit protection and thermal shutdown, which are essential for safeguarding sensitive circuitry in the SmartNav. With an efficiency typically in the range of 85–90%, it minimizes power loss and helps extend battery life. Its compact form factor is particularly well-suited for integration into the confined spaces of our SmartStick, ensuring reliable power delivery for critical components. Although it offers less output adjustability compared to other converters, its stability and high-grade protection make it a strong candidate for our design, especially where performance and safety are paramount.

### 3.1.13.2 TPS61030 Step-Up Voltage Regulator

The TPS61030 from Texas Instruments is a high-efficiency DC-DC boost converter that offers an adjustable output, allowing it to be precisely set to 5V as required by the SmartNav design. It operates over a versatile input range of about 2V to 6V, accommodating battery voltage fluctuations effectively. With efficiency reaching up to 90%, the TPS61030 minimizes energy waste, which is crucial for battery-powered applications. While it requires additional external components such as inductors and capacitors adding some complexity to the design this also provides flexibility in fine-tuning the output voltage. Integrated protection features further safeguard against voltage spikes and other irregularities. Its very small, chip-scale footprint ensures that it occupies minimal board space, making it an attractive option for our compact, portable SmartStick where efficient power management and precise voltage regulation are essential.

### 3.1.13.3 TPS61200 Boost Converter

The TPS61200 is a boost converter tailored for ultra-low power applications, making it particularly attractive for energy-sensitive, battery-operated devices like our SmartNav. It is capable of operating from extremely low input voltages (as low as 0.3V) up to 5.5V, which is ideal for extending battery life as voltage drops occur. Although its continuous output current is lower typically around 600 mA it offers competitive efficiency, generally between 80–85%, especially under low-load conditions. Its ultra-compact design and minimal external component requirements make it perfect for space-constrained applications. While the TPS61200 may offer moderate adjustability and fewer advanced protection features compared to the Pololu regulator, its excellent performance in low-current scenarios and remarkable power efficiency make it a suitable candidate for powering peripheral circuits and sensors within the SmartNav system.

### 3.1.13.4 Step-Up Voltage Regulators Comparison

**Table 3.15. Step-Up Voltage Regulator Comparison.**

| Parameter           | Pololu 5V Regulator | TPS61030               | TPS61200               |
|---------------------|---------------------|------------------------|------------------------|
| Input Voltage Range | 2.5V – 10V          | 2V – 6V                | 0.3V – 5.5V            |
| Output Voltage      | Fixed 5V            | Adjustable (set to 5V) | Adjustable (set to 5V) |
| Efficiency          | ~85–90%             | ~90%                   | ~80–85%                |

|                           |                         |                           |                   |
|---------------------------|-------------------------|---------------------------|-------------------|
| Continuous Output Current | ~1A                     | ~1A                       | ~600mA            |
| Protection Features       | Advanced (S/C, thermal) | Integrated protections    | Basic protections |
| Size/Form Factor          | Compact                 | Very small (chip-scale)   | Ultra-compact     |
| Complexity                | Low                     | Moderate (external parts) | Low               |
| Adjustability             | Fixed                   | High                      | Moderate          |
| Noise/Ripple              | Low                     | Moderate                  | Moderate          |

We chose the TPS61200 boost converter because it offers exceptional efficiency and a broad input voltage range in a compact package, making it ideally suited for our SmartNav Stick. Its ability to reliably step up low battery voltages to a stable output is crucial for maintaining consistent performance across our various sensors and processing components. Moreover, the low quiescent current and robust load regulation characteristics of the TPS61200 ensure longer battery life and improved overall system stability, which are critical for a portable.

### 3.1.13 Power Source

For our SmartNav device, the power source is an important factor to think about, it should be able to deliver power to all components and do this efficiently. The power source needs to be compact and lightweight enough to fit comfortably on our Smart stick, as this is an important design constraint, it should be able to deliver power for long periods of time but having sufficient longevity. This means that we heavily considered a rechargeable solution in order to fit our design constraints. Our smart stick should function without the need of utilizing disposable batteries. For our rechargeable batteries, we decided to focus on specifically lithium-ion and lithium-polymer. Because of the high energy offered and longer life cycle, all this is possible while also being lightweight which are our key considerations for this project. We looked at specs like voltage, capacity, charge time and longevity. Ultimately, the goal is to ensure that the battery would allow the SmartNav to power the GPS, sensors, and other critical components without compromising on size or weight, making it practical for everyday use while still being powerful enough to support all its features.

#### 3.1.13.1 Lithium-Ion (Li-ion) vs. Lithium-Polymer (Li-Po) Batteries

During the selection of the power source of the smart stick, we compared the : Lithium-Ion (Li-ion) and Lithium-Polymer (Li-Po) batteries, as these were the most common rechargeable battery types on the market that also fit our constraints. They both offer high energy, long life cycles and they are also

lightweight, making it very suitable for our portable mobility stick. We compared their key differences that may impact the performances and integration into our design. These key differences include Li-ion batteries' traits of their high capacity and energy density, making them ideal for applications that require long-lasting power. They provide a high mAh rating per unit size, allowing our smart stick to operate for extended periods of time with just a single charge. A key issue for these batteries is the structure, they have a limited design flexibility which may pose a challenge trying to integrate them into our smart stick design.

Li-Po batteries are somewhat different in design and are more flexible than those of Li-Ion batteries, making it easier to have thinner and lighter batteries that can be placed into slim enclosures. They also have a lower risk of leak than conventional Li-ion cells. But in general Li-Po batteries have a lower energy density and may not last as many charge cycles as Li-ion batteries. Charging and discharging them needs to be directionally managed to prevent from swelling or gradual degradation over long periods of time. All this in mind then, a Lithium-Ion battery (Li-Ion battery), particularly the 18650 Li-ion cell - high capacity, stable, and long lifetime that can maintain the SmartNav stick-up for supporting essential components such as GPS and sensors without the need for frequent recharging.

**Table 3.16. Battery Tech Comparison.**

| Parameter           | Lithium-Ion (Li-ion)             | Lithium-Polymer (Li-Po)        |
|---------------------|----------------------------------|--------------------------------|
| Energy Density      | High (~150–200 Wh/kg)            | Medium (~100–130 Wh/kg)        |
| Form Factor         | Rigid (cylindrical, rectangular) | Flexible (can be flat/thin)    |
| Weight              | Light                            | Lighter                        |
| Design Flexibility  | Low                              | High                           |
| Cycle Life          | ~500–1000 cycles                 | ~300–600 cycles                |
| Charge Time         | Moderate (1.5–3 hrs)             | Fast (1–2 hrs)                 |
| Self-Discharge Rate | Low                              | Very Low                       |
| Durability          | Robust                           | Sensitive to swelling          |
| Cost                | Lower                            | Higher                         |
| Risk of Leakage     | Medium                           | Very Low                       |
| Maintenance         | Lower                            | Needs careful charging control |

|                      |                              |                               |
|----------------------|------------------------------|-------------------------------|
| Typical Applications | Power tools, laptops, mobile | RC devices, drones, wearables |
|----------------------|------------------------------|-------------------------------|

After comparing the two technologies, we selected Lithium-Ion (Li-ion) batteries for our SmartNav project due to their high energy density, longer cycle life, and reliable power delivery. Despite having a more rigid form factor than Li-Po, Li-ion batteries like the 18650 cell are compact enough to fit within the design constraints of our smart stick while providing long-lasting, stable power to critical components such as GPS modules and sensors. Their lower cost, less maintenance, and proven track record in portable electronics make them an ideal match for our goal of creating a dependable, rechargeable navigation aid for visually impaired users.

### 3.1.13.2 18650 Li-ion

When developing the SmartNav project, we needed a power source that would have a good mix of high energy density and cycle life with a small package, to deliver it the way we needed to run efficiently and reliably. All of the examined options showed the 18650 Li-ion battery characterized the best performances for portable electronics and embedded systems. The 18650 is a kind of cylindrical lithium-ion battery which has been popular in applications requiring for a light-weight power source, and its name is based on its dimension (18mm diameter and 65mm length). The 18650 Li-ion battery, in particular, offers high capacity (usually from 2500mAh to 3500mAh), meaning that it can keep essential components such as GPS, sensors, and microcontrollers running for a long time while needing to be charged, SmartNav stick. By the way, Li-ion is notably energy rich compared to other rechargeable batteries – more than twice as much energy density as NiMH or LiFePO<sub>4</sub> which is another good reason why this chemistry is a good power solution for our mobility device. Lifecycle charge of 18650 battery is another important reason for considering and choosing 18650.

It has a lifecycle ranging from 300 to 500 charge cycles, guaranteeing durability while minimizing the need for replacements, thus marking the SmartNav as a device that is both economical and eco-friendly. Moreover, Li-ion batteries have a low self-discharge rate, allowing them to retain their charge when not in use; thus, the stick will be ready for operation even after longer periods of inactivity. Keep in mind safety So why using 18650 batteries? Li-ion batteries are prone to becoming unstable, fragile to overcharge, deep discharge and heating, so we added a protection circuit (PCM/BMS) to eliminate overvoltage, overcurrent, overheating operating conditions and ensure safe and reliable operation. The 18650 battery consists of one of the smallest cylindrical shapes that still holds significant capacity allowing the batteries to easily be fit around the SmartNav without a great deal of extra weight or bulk. That said, when it comes to space

optimization with flat Li-Po (lithium-polymer) batteries, the stiff form factor of Z-Packs could create some very small limits. However, the 18650 Li-ion battery is still one of the most efficient and practical in terms of stable voltage, decent capacity and long-term reliability in the market, allowing SmartNav to function all day, seamlessly.

### **3.1.13.3 Flat Li-ion Pouch Battery**

Another variant within the Li-ion family that we considered was the flat Li-ion pouch cell, often seen in mobile phones, power banks, and compact devices. These batteries offer the same 3.7V nominal voltage but are packaged in a thin, lightweight pouch format that provides more flexibility in form factor, allowing them to be embedded into tighter or irregular enclosures. Pouch-style Li-ion batteries typically come in capacities ranging from 1000mAh to 3000mAh, which makes them suitable for mid-sized portable electronics where space and weight are priorities. The flexible shape of these batteries allows designers to create slimmer, more integrated devices.

However, while they offer excellent space efficiency, pouch cells often require additional physical protection, as their soft casing is more vulnerable to puncture or mechanical damage compared to the rigid cylindrical shell of 18650 cells. Their thermal management is also a concern in compact environments, as they can heat up during charging or discharging cycles. These batteries may also have slightly shorter cycle lives (~300–400 cycles) and are more prone to swelling if not charged correctly. Like 18650s, they require a Battery Management System (BMS) to control overcharge, under-voltage, and overheating. Despite these challenges, flat Li-ion pouch batteries remain a viable option for ultra-slim applications but for a device like SmartNav, which benefits from robust capacity and stable power delivery, their limitations in durability and capacity made them a secondary option.

### **3.1.13.4 Power Source Comparisons**

To evaluate whether our potential power sources were feasible, we made a preliminary calculation of the power requirements of the SmartNav stick. The idea was to see what the average power draw would be across the essential components during normal use, helping us to select a battery that has enough capacity while still retaining the device's portability. We will use a central microcontroller unit (MCU) along with multiple environmental sensors (e.g., time-of-flight and radar-based modules), AMD GPS module/antenna for positioning, a small speaker for audio feedback, input interfaces (e.g., buttons), and a voice recognition module for user interaction.

The following table provides an approximate list of current draw numbers seen under typical operating conditions for this type of devices:

**Table 3.17. Component Current and Voltage Draw.**

| Component Category       | Estimated Current Draw   | Operating Voltage |
|--------------------------|--------------------------|-------------------|
| MCU                      | 10–50 mA                 | 3.3V              |
| Sensor Modules           | 10–100 mA                | 3.3V              |
| Audio Feedback (Speaker) | 20–100 mA (intermittent) | 5V                |
| GPS Module/Antenna       | 20–40 mA                 | 3.3V              |
| UI (Buttons/Voice Input) | 1–30 mA                  | 3.3V              |

These estimates suggest that, under typical conditions, the SmartNav stick might have an average current draw in the range of 100 to 300 mA.

To assess battery suitability, we calculated estimated run times based on an average draw of 300 mA:

**18650 Li-ion (3000mAh) →  $3000 \text{ mAh} / 300 \text{ mA} = 10 \text{ hours of use}$**

**Flat Li-ion Pouch Battery →  $2000 \text{ mAh} / 300 \text{ mA} \approx 6.67 \text{ hours of use}$**

This initial power consumption estimation helps ensure that our chosen battery solution whether it be Li-ion or Li-Po can reliably support all system components during daily use. Further measurements and testing will be essential to fine-tune these estimates and optimize power management in the final design.

**Table 3.18. Li-ion Battery Comparison Table.**

| Parameter           | 18650 Li-ion Battery      | Flat Li-ion Pouch Battery       |
|---------------------|---------------------------|---------------------------------|
| Form Factor         | Cylindrical (18mm x 65mm) | Flat, flexible pouch            |
| Capacity Range      | 2500–3500mAh              | 1000–3000mAh                    |
| Nominal Voltage     | 3.7V                      | 3.7V                            |
| Energy Density      | High (~150–200 Wh/kg)     | Medium to High (~130–180 Wh/kg) |
| Cycle Life          | ~300–500 cycles           | ~300–400 cycles                 |
| Self-Discharge Rate | Low                       | Low                             |
| Weight              | Moderate                  | Lightweight                     |
| Thermal Management  | Good (rigid shell)        | Needs careful design            |

|                            |                               |                                    |
|----------------------------|-------------------------------|------------------------------------|
| Durability                 | High (sturdy casing)          | Lower (soft pouch)                 |
| Integration Flexibility    | Moderate                      | High (thin/slim)                   |
| Protection Circuitry (BMS) | Required                      | Required                           |
| Swelling Risk              | Low                           | Medium to High                     |
| Common Applications        | Power tools, e-bikes, laptops | Phones, tablets, compact wearables |

After thorough research as well as understanding the project requirements we have preferred 18650 Li-ion battery as our power source. It has a higher energy density and cycle life so that critical components such as the MCU, sensors, GPS, and audio modules can run for a long time; also, this small cylindrical battery can be placed in the SmartNav stick housing. While we could have used Li-Po batteries as they are mechanically more flat, we decided that the proven performance and durability of the 18650 cell are worth a slight adaptation of our design. By visiting the internal space, particularly with a compartment to ensure secure axial mounting, the 18650 can reside comfortably in the confines of our housing without affecting the ergonomic and lightweight nature of the stick, making it a sustainable and robust power solution for our smart mobility device.

### 3.1.14 Communication Protocol

#### 3.1.14.1 UART

The Universal Asynchronous Receiver Transmitter (UART) protocol is a hardware protocol that uses two wires (full-duplex), a transmit wire and a receive wire, for serial data communication between electronic devices. The UART device performs a parallel to serial data conversion (using shift registers) when transmitting and serial back to parallel conversion when receiving. UART doesn't require the use of a shared clock signal but rather uses a start and stop bit, which allows devices with different clock speeds to communicate. An optional parity bit is used for error checking and there is typically 5 to 8 bits of data. UART relies on the transmitter and receiver having the correct, matching baud rates for synchronization. Having mismatched baud rates would lead to errors caused by timing issues during sending and receiving. The UART data transmission rate is typically 230 kbps to 460 kbps.

The UART protocol is widely used in computers, embedded systems and in microcontrollers taking the parallel data, transforming it for serial communication with various devices and sensors. This protocol is typically used for

communication between 2 devices. This is a commonly used form of communication when connecting the microcontroller to the computer when debugging systems, such as the SmartNav Cane, allowing device monitoring and interaction over the serial interface to capture data and current events. Many GNSS modules, like the NEO-6M that will be used in this project, use UART as a form of communication to the microcontroller. This protocol provides a simple yet reliable form of communication that is supported on many devices. The ESP32 supports 3 UART devices, has a programmable baud rate, supports various lengths of data bits and supports both software and hardware flow control.

### **3.1.14.2 I2C**

The Inter-Integrated Circuit (I2C) is a hardware protocol that uses two wires(half-duplex), one serial data (SDA) and one serial clock (SCL), for short distance communication between electronic devices. This protocol uses what is known as the master-slave architecture. In this architecture the master facilitates the communication between devices. The master initiates the data transfers, while the slave devices respond to the requests from the master. The master device controls the clock signal in this synchronous protocol. The devices using this protocol are addressable allowing for multiple device communication on the same data bus. With each device having its own unique 7-bit address the master is able to select which device it wants to communicate with. There can be multiple masters on the same bus but only one can be in control at a time. With that being said, I2C is able to do what UART cannot and that's allow the microcontroller to communicate with multiple devices using 2 lines. I2C is also much faster than UART with data speeds ranging anywhere from 100 kbps to 5 Mbps.

I2C is widely used in various electronics and embedded systems designed for low-speed and short distances between the devices. This protocol is used in configuration interfaces with voltage regulators, analog-to-digital converters, and digital-to-analog converters. Sensors such as temperature, pressure, gyroscopes, accelerometers, etc... that are used in smart devices commonly use the I2C protocol. The VL53L0X ToF sensor that is used in the SmartNave Cane uses I2C to communicate with the microcontroller.

### **3.1.14.3 I2S**

The Inter-Integrated Circuit Sound is a hardware protocol that uses three wires, one serialclock/bitclock (SCK/BCLK), one wordselect/LeftRightclock (WS/LRCK) and one serialdata(SD) for synchronous, serial communication of digital audio data using pulse code modulation (PCM). This particular setup ensures high-quality audio transmission with low latency because it is designed to be transmitted without compression. This protocol is typically used in the unidirectional transmission of two-channel (stereo) audio data. This protocol uses

the serial clock line as a continuous clock, the word select directs the audio to the left or right channel and the serial data line is for the actual audio data. In this simple setup, high-quality is maintained using PCM allowing for clean precise audio transmission.

This protocol is widely used in many digital audio applications especially those where high-quality audio is desired. Devices such as digital-to-audio converters (DACs) used in high quality audio playback or in audio recording where analog-to-digital converter audio to the microprocessor for processing of the user's voice commands.

#### **3.1.14.4 USB**

The Universal Serial Bus (USB) protocol was created to standardize the way data and power is shared between devices. This allows for the high speed transfer of data between devices. USB also uses the master-slave architecture where there is only one master (host computer) and various connected slaves (peripherals). The USB protocol can support over one hundred connected devices to a single USB HUB. The host facilitates all traffic on the hub. These addressed devices send and receive data in packets including specific information pertaining data, commands, and status updates. The data types found in the USB packets include a start of frame packet, a token packet, a handshake packet and a token packet. The various fields found in these packets include an 8-bit sync field marking the beginning of the packet, an 8-bit PID field identifying the type of packet, a 16-bit CRC field containing a checksum for the integrity of the packet, and a data field containing the data to be sent.

Since its creation, the USB protocol has evolved over the years. Currently, the USB protocol is on USB4 version 2., which can support data speed upward of 5 Gbps, max voltage of 48 V, and max current of 5 A. USB has replaced serial ports as a faster, more efficient way for computers to communicate with peripherals such as printers, mouse, keyboards, etc..There are many legacy devices and current electronics that still utilize UART as a means to communicate with embedded systems. The ESP32, depending on the version, has a USB port that can be used for direct communication between the computer and microcontroller offering a more advanced debugging. Otherwise other versions of the ESP32 will require a USB-to-UART bridge chip to allow for communication between the computer and ESP32 microcontroller.

### **3.2 Software Selection**

In researching the required software needed to implement the various components associated with this project some considerations we had in

mind were familiarity, ease of usage and having a smooth overall development process. We focused on researching the protocols, algorithms and libraries associated with our chosen components.

We needed to first identify the components we planned to use or those of high consideration. Looking at the specifications, we found the various platforms these components are based on.

### **3.2.1 Development Environment**

#### **3.2.1.1 Arduino IDE**

The Arduino IDE is an open source software development program that is used in coding. This software supports C/C++ programming. The Arduino IDE is known to be well suited for beginners and has large community support. The Arduino IDE will be used to write, compile, and upload the code to the board. This is how we will design the overall operation of the cane. Sensor processing, battery management, and user interfaces will all be designed via the development environment. This IDE contains many libraries that can be used in the development of the SmartNav Cane.

#### **GPS and Navigation**

TinyGPS++ is a library that is a part of the Arduino IDE that can be used to get the GPS data from the GPS module and parses the NMEA sentences. This library is small yet provides a means of simplifying the processing of GPS data such as speed, position, time, date, etc. This library has an easy to use user interface making it an ideal option.

NeoGPS is another library that is also used to parse GPS data from the module but offers more advanced control. This library makes it possible to efficiently take GPS data and process it using minimal resources. This library also supports the usage of proprietary protocols like UBX, which is a binary protocol developed by u-blox, the makers of the NEO-6M GPS module. Using the UBX protocol will provide for a more compact and efficient way to interface with the GPS vs using the standard NMEA ASCII- based protocol. So when it comes to providing adequate location and navigation support using the NEOGPS goes above and beyond. NEOGPS will provide for a highly customizable and more intricate control over the data processing and interaction with the GPS module.

#### **Object Detection**

NewPing is a library that can be used to interface with ultrasonic sensors. This library supports a number of sensors such as the HC-SR04. This library is known to enhance the overall effectiveness of using an ultrasonic sensor. NewPing is

compatible with various architectures such as avr, arm, megaavr and esp32. This library will simplify and reinforce the usage of an ultrasonic sensor for our object detection.

The VL53L0X library developed by Pololu is supported in the Arduino IDE. This library makes it easier to interface with the VL53L0X TOF sensor. This library contains much of the functionality needed to effectively receive and utilize the data provided by the sensor. There is an Adafruit\_VL53L0X library that is also available but requires additional Adafruit libraries be installed.(Adafruit Unified Sensor and Adafruit BusIO)When it comes to using LiDAR for object detection we also found there are a number of sensors supported with their respective libraries, simplifying the software development.

Using Arduino IDE's library manager we are able to navigate to the compatible sensor library and install. We can also manually install libraries.

When looking into the mmWave sensor we found this will require a hands on approach which may require us to keep things simple to avoid time consuming testing and debugging of more complex tasks. We would be interfacing with the sensor via UART getting information such as the presence of an object or the distance.

## **User Interface**

The user interface with the SmartNav Cane will include receiving audio alerts as well a haptic feed when an object is detected. We found that the ledc write library that is found in the Arduino IDE performs pulse width modulation(PWM) where we can create an algorithm that provides haptic feed in correlation to an object's distance. Another option we found is to use the DRV2605L from Adafruit in conjunction with the Arduino library provided to implement the haptic feedback response to object detection.

For the audible alerts we found we could also use a buzzer in conjunction with the ledc write utilizing PWM to alert the user when an object is detected and change the frequency of buzzing in correlation to the distance of the object. This requires us to create an algorithm that relates distance to the duty cycle of the buzzer. We have also considered using the Adafruit mini speaker or similar and a text to speech(TTS) module. There is also the option of cloud based TTS but this would come with added costs. The use of the TTS module requires us to create strings of text to be sent to the module and we found libraries, such as Talkie, supporting this functionality.

## **Battery Management**

With this being a battery powered device, efficiently managing power is essential. We found that this can be done sufficiently with the libraries and functions provided in the Arduino-IDE. Performing tasks like reducing the clock speed of

the CPU or placing it into a sleep mode can significantly reduce power consumption. This IDE makes it simple to disable the Wifi and Bluetooth, which are the most power hungry components. The MCU can go from about a 240 mA current draw to about 5uA while in hibernation.

### **3.2.1.2 PlatformIO**

PlatformIO is an open-source development environment that leverages C/C++ for coding, similar to the Arduino IDE but with enhanced capabilities. It features intelligent code completion and a smart code linter, which streamline the development process by reducing syntax errors and improving code quality. Additionally, PlatformIO offers superior library management by automatically downloading and updating necessary libraries, thereby reducing manual overhead and ensuring compatibility across project components. Faster compilation and integrated debugging further contribute to a more efficient and productive development cycle, making PlatformIO a powerful tool for building sophisticated projects like the SmartNav Stick.

### **3.2.1.3 ESP-IDF**

The ESP-IDF is an open-source development framework from Espressif Systems, the makers of the ESP32. This is the official development framework for the ESP32. Like the other development environments mentioned it contains various tools, libraries and APIs needed in the development of ESP32 based applications

ESP-IDF allows for a more specialized development experience and is suited for the more advanced projects. It contains programs such as Cmake and Ninja, which are used in conjunction with one another to reduce the compilation time of programs.

It also has ESP-IDF Monitor, which provides more advanced debugging over environments like Arduino-IDE. This environment has software components that were specifically designed to interface with the hardware of the ESP32. FreeRTOS, which is included in the SDK, is an open-source, real-time operating system designed for embedded devices. Amongst the many libraries and APIs, this environment contains the peripheral drivers for the GPIOs, I2C, SPI, UART, PWM, etc. It also has the Wi-Fi and Bluetooth APIs for networking and communication that were specifically designed for the ESP32. This would be particularly useful in the implementation of the stretch goals such as smartphone interface. We found power management tools to efficiently conserve battery life such as power saving modes such as, light sleep, deep sleep, and ultra low power. While we found the ESP-IDE does provide more advanced and specifically

developed tools for the ESP32, we found this particular development environment is not as user friendly for the novice user.

### 3.2.1.4 Development Environment Comparison

While each of the different environments has its own strengths, we found that for the complexity and level of programming required for this project, the Arduino-IDE is the best fit due to its simple user interface, existing libraries, code structures and large community support. For future endeavors, such as those listed in the stretch goal and beyond IDEs such as PlatformIO or ESP-IDF that offer maximum performance and control may be a better fit.

**Table 3.19. Development Environment Comparisons.**

|                   | Adruino_IDE                                      | PlatformIO                         | ESP-IDF                                                      |
|-------------------|--------------------------------------------------|------------------------------------|--------------------------------------------------------------|
| Language Support  | C/C++                                            | C/C++, Python                      | C/C++, Python, Rust                                          |
| Use case          | Beginners/<br>user-friendly/quick<br>prototyping | Intermediate,<br>advanced features | Experts/steep<br>learning<br>curve/performance-critical apps |
| Board Support     | Arduino and ESP32                                | Various platforms and frameworks   | ESP framework                                                |
| Debugging         | Limited(serial monitor)                          | Advanced(JTAG, GDB)                | Full                                                         |
| Community Support | Large/ many examples                             | good/ growing                      | Official Espressif Support                                   |
| Compiler          | Slow                                             | Moderate                           | Optimized(fastest)                                           |

## 3.2.2 Programming Language

The right language is critical for the SmartNav Cane's firmware: it must deliver low-latency sensor processing, efficient power use, and reliable connectivity within tight memory and CPU constraints. We evaluated three candidates C/C++ for maximum performance and direct hardware control, MicroPython for rapid development and simplicity, and Rust for built-in memory safety and concurrency. Each offers a unique balance of speed, resource usage, and developer productivity, and our comparison will identify which best meets the SmartNav Cane's real-time, battery-powered requirements.

### 3.2.2.1 C/C++

C/C++ has always been the language of choice for embedded systems because of its ability to directly access low-level hardware resources, a critical feature for

a complex, real-time system such as our Smart Navigation Stick. With these languages, developers can write highly optimized code to do everything from reading sensor inputs to managing complex audio processing to executing multiple API calls in parallel. Due to its compile-to-machine-code nature, C/C++ offer the best possible performance with almost zero runtime overhead, which is vital when dealing with data coming from countless sensors and achieving low-latency communication with other pieces of the system. A huge ecosystem of libraries and widely used frameworks are available for rapid development for microcontrollers, alongside sophisticated debugging and profiling tools that can identify problems in performance-critical parts of the code. This level of control, on the other hand, is a double-edged sword, has a steep learning curve as the developer will deal with memory directly and also need to work with pointer arithmetic, which can be the source of subtle bugs and stability issues if the code is not properly managed. On the other hand, for our Smart Navigation Stick, where every millisecond matters and hardware resources are limited, we required the level of control and predictability that C/C++ offered, although it came at the price of a steeper development curve and higher risk for low-level programming errors.

### **3.2.2.2 Micropython**

MicroPython makes bringing the beauty of Python into embedded systems and the ultimate high-level programming language is super attractive here because of rapid prototyping and iterative development. For our Smart Navigation Stick, we showcase MicroPython since it allows for more readable and shorter code in each block, which becomes extremely useful when integrating a wide range of functions such as sensor data acquisition, and network communications, and audio processing into a single system. MicroPython has many great features, one of which is its interactive REPL (Read-Evaluate-Print Loop), which allows developers to test code snippets and debug in real time, significantly reducing development time and reducing the difficulty barrier when working with embedded programming.

Because MicroPython is an interpreted language, it is inherently slower than a compiled language such as C/C++, and its memory management is limited (though much less so). This is a limiting factor in cases of very time-critical operations, or for audio processing and sensor fusion over very large data buffers. However, for most of the Smart Navigation Stick's features particularly those that aren't, well, ultra time-sensitive the rapid development, easy maintenance, and active, welcoming community afforded by MicroPython make it a very appealing option. While there's a raw performance trade-off, the productivity improvements more than make up for it, and so it's in the right place for developers who value power and clarity over performance in production, if only barely so.

### **3.2.2.3 Rust**

Rust is a systems programming language that is becoming increasingly popular in the embedded systems space, primarily due to its focus on safety, concurrency, and performance. Rust, in contrast to C/C++, is built to ban the common causes of bugs from ever occurring, like null pointer dereferencing, or data races from competitive accesses to shared memory by using its borrowing model and compile-time security explains the last.) In Rust at the same time, this translates to strong code that isn't prone to crashing, while also being quick and secure which, given the number of modules that need to work smoothly in our Smart Navigation Stick, from sensor input to network communication, is a key benefit. Hopefully you can see that Rust brings a lot of the awesome features you love about modern languages without sacrificing performance, due to night and day speed comparisons between Rust and C/C++.

Furthermore, its contemporary tooling and package management (via Cargo) make managing project dependencies and builds easier, streamlining the overall development process. That said, the embedded ecosystem still includes very young libraries and frameworks, and the immediate availability of resources are still different in comparison with the very matured environment offered by C/C++ or even MicroPython. While that makes the language powerful, its steep learning curve can also be a barrier, as developers must adapt to compile-time rules and programming paradigms exclusive to the language. Even, with all these difficulties, Rust is an innovative and safe language, which may significantly improve the overall system stability and maintainability of the future smarter systems, particularly the Smart Navigation Stick which is inherently a complex system and should ensure high level of reliability along with concurrent processing for further usability.

### **3.2.2.4 Programming Language Comparison**

Each of these programming languages brings a unique set of advantages and trade-offs to the table. C/C++ offers unparalleled performance and granular hardware control, making it indispensable for real-time, resource-constrained applications, though it requires meticulous management of low-level details. MicroPython, with its high-level, easy-to-read syntax and interactive development environment, accelerates development time and simplifies code maintenance but sacrifices some performance. Rust, on the other hand, aims to combine the best of both worlds by delivering modern safety features and concurrent performance while still being efficient enough for embedded applications, albeit with a steeper learning curve and a less mature ecosystem. The final choice will depend on the specific priorities of our project whether we need the absolute best in performance and control, rapid development and ease of use, or a balance of safety and modern language features for long-term reliability in our Smart Navigation Stick.

**Table 3.20. Programming Language Comparison.**

| Programming Language      | C/C++    | MicroPython | Rust                                      |
|---------------------------|----------|-------------|-------------------------------------------|
| Execution Model           | Compiled | Interpreted | Compiled                                  |
| Performance (1-5)         | 5        | 3           | 5                                         |
| Hardware Control (1-5)    | 5        | 3           | 4                                         |
| Ecosystem Maturity (1-5)  | 5        | 4           | 3                                         |
| Development Speed (1-5)   | 3        | 5           | 3                                         |
| Concurrency Support (1-5) | 4        | 3           | 5                                         |
| Memory Management         | Manual   | Automatic   | Automatic with strict compile-time safety |
| Learning Curve (1-5)      | 5        | 2           | 4                                         |

### 3.2.3 Navigation API Integration

In our software system an important aspect of the navigation sequence is being able to access the turn by turn directions, this is obtained from an api call made within our code. The api calls are processed by various providers, for our system we need an efficient, fast and affordable navigation sequence provider. In our research three main websites were able to fulfill our constraints regarding Nav api calls. In this chapter we presented those websites.

#### 3.2.3.1 Google Maps API

One of the most powerful navigation APIs available today, Google Maps API is an ideal choice for the SmartNav Cane. The Google Maps API is specifically engineered to disseminate highly granular geographical data, offering everything from detailed mapping capabilities to turn-by-turn navigation and real-time traffic updates, as well as dynamic rerouting capabilities that would be essential for assisting visually impaired individuals across a variety of environments. For response timing, again, under ideal circumstances, response times are typically under 300 ms, so the flight data is provided in a timely manner, allowing a quick turn-taking. Our API comes with a full set of RESTful endpoints, enabling full integration with other software components in the SmartNav system. Moreover, Google Maps is supported by a massive, continually updated

database across the globe, ensuring it remains accurate and trustworthy even in less densely mapped areas. One of the downsides of using the Google Maps API however, is that it can get costly: the more requests you make, the more you'll need to pay, and the API imposes limits on the number of requests you can make; so in a resource-constrained project these need to be managed carefully. Nonetheless, its unmatched data quality, extensive documentation, and straightforward integrations make it an excellent contender for a navigation aid that should

### **3.2.3.2 OpenStreetMap APIs**

OpenStreetMap (OSM) APIs provide a fully open-source, community-driven mapping platform that is exceptionally well suited for budget-conscious projects like the SmartNav Cane. OSM's tile servers deliver both raster (PNG) and vector (Mapbox Vector Tile) formats via standard REST endpoints, with average response times of 200–350 ms under normal network conditions. There are no usage fees or hard request limits—only fair-use policies—making OSM effectively free beyond infrastructure costs. For routing, OSM integrates seamlessly with open-source engines such as OSRM (Open Source Routing Machine) or GraphHopper, each capable of sub-100 ms route computations on hosted servers and offering pedestrian-optimized profiles with wheelchair-accessible path data. Geocoding via Nominatim returns structured JSON in <300 ms and supports bulk queries with configurable rate limiting. Offline use is natively supported by downloading vector tile packs (MBTiles) and hosting a local routing engine, ensuring zero latency navigation in low-connectivity environments. Tile data refreshes weekly—providing up-to-date map information—and the platform's open data model allows custom tagging (e.g., curb cuts, tactile paving) to enhance accessibility. Pros: zero licensing cost; complete control over data, offline hosting, highly customizable routing profiles. Cons: variable regional data completeness; requires self-hosting for high availability and performance tuning of routing servers; lacks built-in traffic or dynamic hazard alerts.

### **3.2.3.3 Mapbox Directions API**

The Mapbox Directions API is a robust, developer-friendly solution that offers highly customizable mapping and navigation services tailored for a wide range of applications, including pedestrian-focused projects like SmartNav. Built with flexibility and performance in mind, Mapbox supports turn-by-turn navigation, pedestrian and wheelchair-accessible routing, and real-time traffic data, making it ideal for assistive mobility technologies. One of its standout features is Map Matching, which intelligently aligns raw GPS data to road or path networks—this not only improves route accuracy but also reduces erratic behavior in urban environments where signal noise is common. In terms of responsiveness, Mapbox delivers competitive API response times averaging between 150–250 ms, ensuring near real-time interaction when handling route requests and updates. This is particularly important for SmartNav's real-time navigation,

where delays can affect the usability of voice feedback and route recalculations. Additionally, Mapbox's offline tile caching capabilities are a major advantage in scenarios where stable internet access may not be guaranteed especially in areas with poor connectivity. This offline support allows devices to continue operating even when disconnected, using pre-cached maps and routing data. Mapbox provides comprehensive SDKs and API libraries for a variety of platforms, including embedded C/C++, Python, and web-based environments, which simplifies integration into both lightweight microcontroller systems and higher-level backend services. Developers have access to a fully configurable UI and styling tools, allowing SmartNav to visually or audibly adapt the map interface to meet accessibility requirements for visually impaired users.

From a cost perspective, Mapbox offers a generous free tier of 50,000 map loads per month, making it accessible for early-stage development and testing. Beyond this limit, pricing starts at \$0.50 per 1,000 requests, which can add up depending on the scale of deployment. While this pricing model is reasonable for many commercial applications, it may become a limitation for budget-constrained or high-frequency projects. Pros of using the Mapbox Directions API include its strong offline support, high-quality pedestrian routing, extensive customization options, and professional-grade GPS smoothing features. However, cons include a steeper learning curve for developers seeking to tailor the platform's full capabilities and higher costs once usage exceeds the free tier. Nonetheless, its powerful feature set and reliable performance make it a strong candidate for integration into advanced assistive navigation systems such as SmartNav.

### 3.2.3.4 Directions API Comparisons

**Table 3.21. Navigation API Comparison.**

| API Name               | Google Maps API                              | OpenStreetMap APIs                     | Mapbox Directions API                                 |
|------------------------|----------------------------------------------|----------------------------------------|-------------------------------------------------------|
| Response Time          | <300 ms                                      | 200–350 ms                             | 150–250 ms                                            |
| Cost (Free Tier)       | \$200 credit/month;<br>\$5 per 1000 requests | Free                                   | 50k map loads/month;<br>\$0.50/1k requests thereafter |
| Request Limits         | 100k/day                                     | Fair-use policy (no hard limit)        | 50k/month                                             |
| Offline Support        | No (paid SDK)                                | Yes (MBTiles + local server)           | Yes (offline tile caching)                            |
| Routing Accuracy       | Very High                                    | Moderate–High (region dependent)       | High                                                  |
| Accessibility Features | Pedestrian routing;<br>wheelchair routing;   | Custom tags for<br>curb cuts & tactile | Pedestrian routing;<br>Map Matching for               |

|                        |              |               |          |
|------------------------|--------------|---------------|----------|
|                        | transit info | paving        | accuracy |
| Integration Complexity | Moderate     | Moderate–High | High     |

### 3.2.4 Embedded Operating Systems

Choosing the right real-time operating system (RTOS) was a fundamental choice for the SmartNav Cane’s microcontroller. It affects responsiveness, power efficiency, and long-term reliability of our navigation aid. An RTOS synchronizes all of these processes running concurrently, with deterministic timing guarantees that bare-metal firmware cannot offer, even for frequent tasks such as high-frequency sensor polling, audio feedback, and wireless communication and power management. So an operating system with an appropriate footprint that has just enough functionality and peripherals, zero-latency handling, and a streamlined development pipeline is essential and ideal: an overstuffed kernel wastes precious RAM and battery cycle life, while a minimal-scheduler forces us to reimplement critical services. We examine three deployed candidates of embedded RTOS (FreeRTOS, Zephyr OS, and RIOT OS) in terms of memory costs, interrupt latency, driver ecosystems, and power-reduce solutions to find out which RTOS meets the tight constraints of our SmartNav Cane design with regards to form factor, battery life, and real-time obstacle detection in the following sections.

#### 3.2.4.1 FreeRTOS

FreeRTOS is the most popular open-source real-time operating system for microcontrollers in the world, running on everything between 8-bit AVR chips and 32-bit ARM Cortex-M and RISC-V cores. It uses ~8–12 KB RAM and ~30KB flash only, and context-switch latencies as small as 5–10 $\mu$ s on a 48–120MHz MCU. Built-in primitives (tasks, queues, semaphores) support deterministic scheduling of concurrent sensor polling, audio I/O, and networking, with worst-case execution times that satisfy sub-millisecond deadlines. Support for tickless sleep modes that lower idle current under 5 $\mu$ A further extends battery life in portable devices. Its vast ecosystem (20+ years in the market) boasts mature driver libraries for I<sup>2</sup>C, SPI, UART (and many peripheral sensors), OTA and secure boot frameworks. FreeRTOS maximizes performance, minimizes overhead, and achieves rock-solid reliability for a battery-powered SmartNav cane so it's the clear default choice for resource-constrained MCUs.

#### 3.2.4.2 Zephyr OS

Zephyr OS: A full-featured modular RTOS for 32-bit MCUs (ARM Cortex-M, RISC-V, Xtensa) that require richer connectivity and security. Sits in at  $\approx 30$  KB RAM + 200KB flash, with context-switch latency of about 15–20 $\mu$ s; includes built-in Bluetooth LE, IEEE 802.15.4, file systems, hardware abstraction via device trees, significantly reducing custom driver work. Deep-sleep currents hover under 3 $\mu$ A with advanced power-management features the perfect fit for an array of multi-sensor systems. But its larger memory footprint and greater configuration complexity make it more appropriate for devices that require complex networking stacks, or that require future expandability (cloud sync, machine learning) than a lean, assistive cane.

### 3.2.4.3 RIOT OS

RIOT OS targets ultra-low-power IoT nodes (ARM Cortex-M0/M3, RISC-V) with a minimal footprint of 5–8 KB RAM + 30–50 KB flash and sub-5  $\mu$ s context-switch times. It implements a tickless scheduler for  $\mu$ A-level idle currents and supports 6LoWPAN mesh networking out-of-the-box. RIOT's simplicity and deterministic behaviour minimize code size and power use, but its small ecosystem lacks mature drivers for advanced peripherals (audio codecs, high-resolution ToF sensors). Unless the SmartNav cane's roadmap prioritizes mesh-networked sensing over local processing and user feedback, RIOT's gains in efficiency come at the cost of additional driver development.

### 3.2.4.4 RTOS Comparison

**Table 3.22. RTOS Comparison.**

| RTOS                    | FreeRTOS                                     | Zephyr OS                                 | RIOT OS                   |
|-------------------------|----------------------------------------------|-------------------------------------------|---------------------------|
| Footprint (RAM + Flash) | 8–12 KB + $\sim 30$ KB                       | $\approx 30$ KB + 200 KB                  | 5–8 KB + 30–50 KB         |
| Context-Switch Latency  | 5–10 $\mu$ s                                 | 15–20 $\mu$ s                             | <5 $\mu$ s                |
| Idle Current            | <5 $\mu$ A                                   | <3 $\mu$ A                                | <1 $\mu$ A                |
| Built-in Connectivity   | None (add-on)                                | Bluetooth LE, IEEE 802.15.4, file system  | 6LoWPAN mesh              |
| Driver Ecosystem        | Extensive (20+ years)                        | Strong, growing                           | Modest                    |
| Complexity              | Low                                          | High                                      | Moderate                  |
| Best Use Case           | Resource-constrained MCUs; real-time sensing | Feature-rich devices needing connectivity | Ultra-low-power IoT nodes |

### **3.2.5 Cloud-Based Speech-to-Text**

In order for the SmartNav Cane to react quickly and accurately to spoken commands, it needs to use cloud-based speech-to-text (STT) services instead of trying to transcribe on the device. Modern automatic speech recognition has high computational requirements—deep neural network inference, noise reduction, natural language processing—incompatible with the memory and processing power of a typical microcontroller. Offloading transcription into the cloud enables such superhuman results (>95% word recognition) and sub-second response times(<300 ms), and continuous model updates without increasing the device's size, weight, or power consumption. In this section, we evaluate four leading STT providers (Google Speech-to-Text, OpenAI Whisper, AssemblyAI, and Azure Speech) against criteria that are critical for a portable navigation aid: latency, accuracy, cost, integration complexity, and robustness in noisy environments. This analysis ensures we select the solution that delivers the fastest, most reliable voice interface for users navigating dynamic campus settings.

#### **3.2.5.1 Google Speech-to-Text API**

Google's Speech-to-Text API is widely regarded as the industry gold standard for cloud-based automatic speech recognition, delivering end-to-end transcription latencies of just 200–300 milliseconds—fast enough to provide near-instantaneous voice feedback on the SmartNav Cane. Its recognition accuracy exceeds 95% for English, even in moderately noisy environments, thanks to Google's advanced neural network models and continuous global data updates. Developers receive 60 free minutes per month, after which usage is billed at \$0.006 per audio minute, providing predictable and competitive pricing for a device expected to process short voice commands rather than continuous streams. Integration into an MCU is straightforward via standard RESTful HTTPS calls using `ArduinoHttpClient` and `ArduinoJson` libraries; both streaming and batch transcription modes are supported, enabling either continuous listening or discrete command capture. Key technical features include built-in speaker diarization (separating multiple voices), robust noise cancellation, and support for dozens of languages and dialects critical for campus settings with varied accents. The primary downsides are the requirement for a Google Cloud account and costs that scale linearly with usage, necessitating careful request management in a battery-powered device. Overall, Google Speech-to-Text strikes the ideal balance of sub-second responsiveness, enterprise-grade accuracy, and manageable cost, making it the premier choice for delivering reliable, real-time navigation commands to visually impaired users of the SmartNav Cane.

#### **3.2.5.2 OpenAI Whisper API**

The OpenAI Whisper API offers cutting-edge speech recognition performance, particularly excelling in challenging acoustic environments where background

noise, multiple speakers, or strong accents would degrade other services. End-to-end latency ranges from 300–500 milliseconds, making it fast enough for responsive voice commands while trading off a slight delay compared to Google Speech-to-Text. Whisper’s transcription accuracy hovers around 94%, but its advanced neural network architecture demonstrates superior robustness against ambient noise and diverse speech patterns—a critical advantage for outdoor campus navigation under windy or crowded conditions. New users receive a \$5 credit (≈833 minutes) to test the service, and subsequent usage is billed at \$0.006 per audio minute, matching the per-minute cost of Google’s API but without hard usage quotas. Integration is straightforward via a single RESTful POST of a WAV payload and parsing the JSON “text” field in the response; however, Whisper currently lacks a streaming endpoint, requiring discrete audio captures rather than continuous listening. This batch-only design simplifies implementation on resource-limited MCUs but precludes real-time streaming transcription. Overall, OpenAI Whisper is the SmartNav Cane’s optimal choice for high-fidelity speech recognition in noisy or variable outdoor environments, ensuring commands are accurately understood when clear audio capture is most difficult.

### **3.2.5.3 Azure Speech Service**

Azure Speech Service from Microsoft provides best in class transcription performance with end-to-end latencies hovering around 200 milliseconds making it the fastest Speech to Text provider among major cloud players. Its recognition accuracy is greater than 95% for English, and it has the unique capability to tune custom acoustic (and language) models to fine-tune performance for specialized vocabularies (e.g., locations on campus). It offers 5 free hours each month for developers, then charges \$ 1.00 /audio hour (≈\$ 0.017/minute) for usage. Integrate using simple REST APIs or native C/C++ and Python SDKs that include built-in authentication, automatic token renewal, enterprise-grade security (TLS encryption, GDPR or HIPAA compliant). Its features include real-time streaming transcription, speaker identification, and profanity filtering all backed by a 99.9% SLA. The key benefit of using Azure Speech Service is its ultra-low latency and rock-solid reliability, which provides virtually instantaneous voice feedback—a primary requirement for safe, real-time navigation on the SmartNav Cane. The trade-off is a higher cost per-minute and slightly more complicated OAuth-based authentication than simpler REST-only services. In scenarios where every millisecond of response time is critical and budget allows, the best option to give our assistive navigation device immediate and reliable voice command recognition capability is to use Azure Speech Service.

### **3.2.5.4 Speech-to-Text Comparison**

***Table 3.23. Speech-to-Text Comparison.***

| Service               | Latency    | Accuracy | Free Tier             | Pricing     | Integration                            | Key Specs                                                |
|-----------------------|------------|----------|-----------------------|-------------|----------------------------------------|----------------------------------------------------------|
| Google Speech-to-Text | 200–300 ms | >95%     | 60 min/mo             | \$0.006/min | REST (ArduinoHTTPClient + ArduinoJson) | Streaming & batch; speaker diarization; noise robustness |
| OpenAI Whisper API    | 300–500 ms | ~94%     | \$5 credit (~833 min) | \$0.006/min | REST (batch only)                      | State-of-the-art model; excellent noise resilience       |
| Azure Speech Service  | ~200 ms    | >95%     | 5 hr/mo               | \$1/hr      | REST + SDKs (C/C++, Python)            | Custom models; enterprise SLA; built-in security         |

Based on detection latency, accuracy, economics, ease of integration, and real-world robustness, we ultimately decided that the Google Speech-to-Text API is the superior choice for our SmartNav Cane. It incorporates industry-leading transcription accuracy (>95%), sub-300ms end-to-end latency and a predictable pricing model (\$0.006 per minute after a generous 60-minute free tier). It has a RESTful interface that integrates very nicely with our MCU's HTTPClient + ArduinoJson stack, and Google's global infrastructure guarantees low response times no matter local conditions. Although other services do better in specific verticals (e.g. noise resilience in Whisper or ultra-low latency in Azure), the balance achieved by Google regarding speed, reliability, cost-efficiency and developer support allows to deliver the most practical and performant solution for near-instantaneous voice command recognition for visually impaired users in dynamic campus setups.

### 3.2.6 Schematic and PCB Design Software

In developing the SmartNav Smart Stick a device that integrates advanced sensor arrays, microcontrollers, and robust communication interfaces the choice of schematic design tools is pivotal. This chapter examines KiCad, Fusion 360, and Eagle, focusing on their respective advantages and disadvantages when applied to building schematics and PCBs for our smart stick. The analysis considers factors such as library comprehensiveness, ease of component importation, and overall integration with our design workflow, ensuring that the selected tool not only meets the technical requirements but also streamlines the development process.

### **3.2.6.1 KiCad**

KiCad stands out as an open-source, cost-effective solution that has steadily gained popularity for PCB design and schematic capture, making it an attractive option for the SmartNav Smart Stick project. Its extensive and ever-growing library, driven by a vibrant global community, offers a wide array of components that can be easily imported and customized, ensuring that specialized sensors and modules required for the smart stick are readily accessible. Although its interface might feel less polished compared to commercial alternatives and may present a steeper learning curve for new users, KiCad's flexibility and active development community provide ample support for troubleshooting and enhancements. For a project like ours, where budget constraints and the need for adaptable design files are paramount, KiCad offers an ideal balance between functionality and cost.

### **3.2.6.2 Fusion 360**

Fusion 360 offers a comprehensive design environment that uniquely blends electronic schematic design with powerful 3D mechanical modeling, a feature that can be particularly beneficial for the SmartNav Smart Stick. This integration allows designers to visualize the placement of PCBs and sensors within the stick's physical enclosure, ensuring that all electronic components align perfectly with the device's ergonomic and structural requirements. The tool's streamlined process for importing 3D models and linking them with circuit designs simplifies iterative prototyping and simulation of real-world interactions. However, the breadth of Fusion 360's features can result in a steeper learning curve, and its electronic design capabilities, while robust, may not match the depth of specialized tools in terms of component library size and intricate schematic functionalities. Despite these challenges, its holistic approach makes it a compelling choice for ensuring that both the electronic and mechanical aspects of the smart stick are cohesively developed.

### **3.2.6.3 Eagle**

Eagle, developed by Autodesk, is renowned for its dedicated focus on electronic design and its user-friendly schematic capture and PCB layout capabilities, making it a strong contender for developing the SmartNav Smart Stick. Its extensive component libraries are a significant advantage, allowing designers to quickly find and import the precise parts needed for our sensor arrays, microcontrollers, and power management circuits. Eagle's intuitive interface, combined with robust error-checking and design rule enforcement, facilitates a smoother design process for complex circuit boards. On the downside, the free version comes with limitations regarding board size and layer count, potentially necessitating a transition to a paid license as the project scales. Additionally,

while Eagle excels in electronic design, it lacks the integrated mechanical modeling environment provided by Fusion 360, which may require supplementary tools to ensure perfect alignment with the smart stick's physical design. Nonetheless, for projects primarily focused on detailed electronic schematics and PCB layouts, Eagle remains a highly effective and efficient option.

### 3.2.6.4 Design Software Comparison

**Table 3.24. Design Software Comparison.**

| Parameter                | KiCad                                 | Fusion 360                                      | Eagle                                     |
|--------------------------|---------------------------------------|-------------------------------------------------|-------------------------------------------|
| License/Cost             | Open-source / Free                    | Subscription-based (free for students/startups) | Freemium / Paid tiers                     |
| Component Library Size   | Extensive, community-driven           | Moderate                                        | Large, Autodesk maintained                |
| Ease of Component Import | High                                  | Moderate                                        | High                                      |
| Mechanical Integration   | No                                    | Yes                                             | Limited                                   |
| 3D Visualization         | Basic 3D viewer                       | Advanced mechanical + electronic                | Yes, for PCB only                         |
| Learning Curve           | Moderate to steep                     | Steep                                           | Low to moderate                           |
| User Interface           | Less polished                         | Polished, modern                                | User-friendly                             |
| Community Support        | Strong, active community              | Moderate                                        | Good                                      |
| Design Rule Checks       | Yes                                   | Yes                                             | Yes                                       |
| Free Version Limitations | Few, mostly none                      | None for paid, free version limited in use      | Board size/layer limits                   |
| Best For                 | Budget-conscious, flexible PCB design | Integrated mechanical + electronic design       | Dedicated electronic schematic/PCB design |

Using Fusion 360 for PCB designs is a great software, it has multiple library tools that make the process a lot easier such as the auto router tool for tracing PCB airwires automatically, this is one of the various tools that the software integrates when implementing PCB designs, it's also very easy to backtrack and avoid confusion throughout multiple process. Which is why we chose this software for our design.

## **Chapter 4: Standards and Design Constraints**

### **4.1 Standards**

Standards are the set of guidelines and specifications put in place to ensure consistency, safety, and reliability in engineering design and manufacturing. The most commonly known standards come from organizations such as ISO (International Organization for Standardization), IEEE (Institute of Electrical and Electronic Engineers), and ANSI (American National Standards Institute). These practices foster an environment where products seamlessly integrate, making the use more convenient for users. For example, wi-fi is based upon an IEEE standard that defines how wi-fi devices communicate, including information such as frequency bands, data rates, and security protocols. This allows for different devices to connect and communicate in an efficient manner.

There are a plethora of standards that we must adhere to throughout the design and creation process of our project to produce a functioning cane. Examples of these standards are listed below.

#### **4.1.1 Assistive Technology Standards**

##### **ISO 9999: Assistive Products for Persons with Disability**

International assistive technology standards must be recognized to ensure the SmartNav Cane meets the specific needs of individuals with disabilities, especially those with visual impairments. \* The ISO 9999 standard categorizes assistive technology products, ensuring that products such as the SmartNav Cane are classified and designed to serve people with disabilities. The standard establishes specific performance criteria, usability norms and accessibility requirements that assistive mobility devices must satisfy. Because ISO 9999 provides an international framework for categorizing assistive devices that aid in independence and mobility, compliance with this classification is essential for the regulatory approval of our device in both the United States and foreign markets. Following this standard, SmartNav Cane can consistently provide its

functionality and be trusted by users and the regulatory bodies across the globe.

### **ANSI/RESNA Standards for Assistive Technology**

In addition to ISO 9999, the SmartNav Cane is also designed to meet the ANSI/RESNA standards established by the Rehabilitation Engineering and Assistive Technology Society of North America. These standards focus on ensuring durability, effectiveness, and user safety through rigorous testing under real-world conditions. A significant aspect of these standards is the emphasis on ergonomic design, which ensures that the device is not only functional but also comfortable and easy to use for individuals with disabilities. Conformity with laws such as the Americans with Disabilities Act (ADA) is another crucial component, as it mandates accessibility and usability criteria that are essential for integration into government programs and healthcare initiatives. Meeting the ANSI/RESNA standards demonstrates the SmartNav Cane's commitment to quality and accessibility, making it a reliable and market-ready solution for those who depend on assistive mobility devices.

### **4.1.2 Sensor Performance and Integration Standards**

#### **IEEE 1451 Series – Smart Transducer Interfaces**

The IEEE 1451 series of standards, on the other hand, is a milestone in sensor integration, establishing a common framework for smart transducer interfaces and propelling the field of data acquisition and instrumentation forward. These standards, developed by the Institute of Electrical and Electronics Engineers (IEEE), aimed to create a common ground among different sensor technologies and the digital systems that handle their data. The IEEE 1451 standard plays a crucial role in the SmartNav Cane, as one of its vital components involves enabling robust communication between a range of sensors, including accelerometers, ultrasonic distance sensors, and environmental monitors, and the central microcontroller through which they interface. The standardization was a natural consequence of the proliferation of sensor technologies in the late 20th century, which gave birth to common language that the sensors would need to speak to each other to minimize compatibility and streamline the engineering of sensor integration. It contributes not only the transport protocols and data formats but also the electrical properties and interface standards, enabling a seamless data exchange across different systems. This standardization helps minimize potential errors and latency in the acquisition of the sensor data to make the navigation assistance of the SmartNav Cane accurate and reliable. We've chosen to adopt IEEE 1451 for our design, as it gives us a strong, widely accepted framework that both allows for interoperability of components and speeds up the development process, improving sensor operation and therefore user safety.

### **ISO/IEC 17025 – Calibration and Testing Laboratories**

ISO/IEC 17025 is a cornerstone standard in the realm of calibration and testing, setting out the criteria for the competence of testing and calibration laboratories around the world. First introduced in the late 1990s and subsequently revised to keep pace with technological advances, ISO/IEC 17025 ensures that measurements and test results are both accurate and reproducible, a critical requirement for any system that relies on sensor data. For the SmartNav Cane, compliance with this standard guarantees that every sensor is calibrated against internationally recognized benchmarks, thereby ensuring the highest level of precision in obstacle detection and navigation. The historical significance of ISO/IEC 17025 lies in its ability to instill confidence in measurement processes making it indispensable for quality control in industries ranging from healthcare to aerospace. Regular calibration, as mandated by ISO/IEC 17025, helps identify and correct even the slightest deviations over time, reducing the risk of cumulative errors that could compromise the device's performance. In essence, integrating ISO/IEC 17025 into our design process not only enhances the reliability of sensor outputs but also builds a robust foundation for continuous quality improvement, ensuring that the SmartNav Cane delivers consistent and accurate navigation assistance in real-world conditions.

### **MIL-STD-810 – Environmental Testing**

MIL-STD-810 is an extensive military standard that describes the various environmental test methods used to qualify the equipment along with a range of environmental conditions. Originally designed for the United States Department of Defense in the 1980s, MIL-STD-810 has since matured into an industry standard for durability and reliability that spans a wide range of industries, including consumer electronics and assistive technologies. In addition to being IP67-rated, for the SmartNav Cane, compliance with MIL-STD-810 essentially means that each and every one of the sensor components and the whole system, for that matter is rated for resistance to extreme temperatures, humidity, vibration, and shock. A standard such as MIL-STD-810 is especially relevant as the SmartNav Cane is designed to operate in a wide variety of environments, from outdoor weather elements to the daily wear and tear that could occur during its use, under this standard, the device remains operational and reliable as intended.

These environmental tests ensure the reliability and robustness of components can ultimately add to the safety and longevity of the device. By incorporating this standard in our design process, we are showcasing our dedication to quality and resilience, so the SmartNav Cane can provide reliable navigation assistance despite external circumstances. The SmartNav Cane integrates sensor-specific standards for seamless communication and interfacing (IEEE 1451), precise calibration and testing (ISO/IEC 17025), and rigorous environmental durability (MIL-STD-810), ensuring it surpasses industry benchmarks in its sensor performance. This Sensor Standardization and

everything else behind it enables the sensors to accurately detect obstacles and provide acceptable navigation assistance, making it safe, efficient and human-centric.

### **4.1.3 Manufacturing and Quality Control Standards**

#### **ISO 9001: Quality Management Systems (QMS)**

ISO 9001 is an internationally recognized standard that examines the quality and consistency of an organization's processes, ensuring that the SmartNav Cane adheres to strict quality control at every stage from design and manufacturing to comprehensive testing. This standard provides a detailed framework for establishing standardized manufacturing processes and robust quality management systems, with a primary focus on ensuring consistent product quality and reliability. By following ISO 9001 guidelines, our organization commits to a well-defined approach that enhances customer satisfaction through high production standards and continuous process improvement. The standard not only streamlines operations, reduces defects and malfunctions through rigorous testing, but also promotes transparency and efficiency across all phases of production. This culture of continual improvement and proactive quality control results in increased productivity, reduced waste, and improved customer trust. In essence, compliance with ISO 9001 serves as both a roadmap and a testament to our dedication to delivering a SmartNav Cane that meets and exceeds industry benchmarks for quality and reliability.

### **4.1.4 Electrical and Safety Standards**

#### **IEC 60601-1: Medical Electrical Equipment Standard**

The SmartNav Cane integrates a variety of electronic components such as sensors, microcontrollers, and batteries which necessitates strict adherence to the IEC 60601-1 standard for medical electrical equipment. This standard was developed to ensure that devices used in sensitive environments are safe from electrical hazards. It lays out comprehensive safety requirements that include the prevention of electric shock, proper battery safety measures, and protection against short circuits. Additionally, IEC 60601-1 addresses environmental protection criteria, ensuring that the device operates reliably under various conditions, whether indoors or outdoors. By complying with this standard, the SmartNav Cane is not only engineered to minimize the risk of electrical failure but also to maintain its safety and operational integrity across diverse settings. This adherence helps guarantee that the device can be safely used by individuals in daily life, ultimately building trust and reliability in its performance.

#### **FCC Part 15: Radio Frequency (RF) Compliance**

In parallel, the SmartNav Cane must comply with FCC Part 15 standards, which regulate radio frequency emissions for electronic products sold in the United States. Given that the cane incorporates wireless communication features such

as GPS, or Wi-Fi, adherence to FCC Part 15 is crucial to prevent interference with other nearby wireless devices, including hearing aids and smartphones. This standard sets stringent limits on RF emissions, ensuring that the device operates within safe exposure levels while also maintaining efficient power consumption. Compliance with FCC Part 15 not only facilitates legal use of wireless technology but also promotes harmonious operation in environments crowded with various RF-emitting devices. The assurance provided by this standard enhances the overall safety and usability of the SmartNav Cane, making it suitable for both personal and public applications without risk of harmful interference.

By rigorously adhering to these electrical and safety standards, the SmartNav Cane is engineered to deliver a secure, reliable, and compliant user experience. Compliance with IEC 60601-1 guarantees robust protection against electrical hazards, while FCC Part 15 ensures that the device's wireless capabilities function safely within regulated limits. Together, these standards underpin the device's commitment to high-quality performance and user safety in every aspect of its operation.

#### **4.1.5 Other Standards**

##### **Cybersecurity and Data Privacy Standards**

In case The SmartNav Cane Gathers or sends user data (for instance GPS location or Bluetooth connections), then it needs to conform to Data security regulations. GDPR or General Data Protection Regulation (EU). Offers protection to user privacy by overseeing data storage and sharing. Requires encryption of communication for secure data transfer. User permission should be obtained before data acquisition. ISO 27001 is for Information Security Management Systems. It ensures cybersecurity protocols for devices equipped with wireless and Bluetooth capabilities. Likewise, it protects against unauthorized access and hacking. Benefit Of Compliance: Prevents legal problems concerning user privacy. It also ensures secure interactions with the Smartnav Cane.

#### **4.2 Constraints**

Engineering constraints are the limitations that must be considered when designing and developing a product. In the real world, all engineering projects will have constraints imposed on them through various channels. These may include clients, companies, regulatory bodies, or others. Because constraints play a vital role in the design and shaping of the final product, it is always important to consider the respective constraints in the early stages of an engineering project. In the case of our project, the largest constraints placed upon us are time, usability, and safety.

##### **4.2.1 Time**

Similar to the other groups, we have quickly found time to be the main constraint of this project. Due to the nature of senior design, we have only two semesters (fall and summer) to complete the research, design, and fabrication of our product. This obviously plays a large role in the knowledge and skills we can gain over this short amount of time to build a successful product exceeding ours and others expectations. This fact has been kept in mind by our group in order to deliver all parts of our project in a timely and professional manner.

Diving into the specifics of this constraint, it is clear that we have to meet all deadlines for the various parts of this project because extensions cannot be granted past two semesters. We must progress at a consistent rate in order to meet all deadlines. This course serves as a final milestone in our academic career at the University of Central Florida and displays the knowledge / skills gained that will translate to our future careers. Failure to follow a strict schedule and fall behind would result in having to retake senior design delaying graduation. Therefore, the amount of time we are limited to plays a large role in the knowledge we can acquire to help build our product.

Another issue pertaining to time is the research and ordering of parts. Specifically, we see a potential issue in ordering printed circuit boards that have to ship internationally. We must do extensive research on the option of available parts for our project and order them in time to receive them for testing and construction. The adaptability of parts and shipment time are important to remember throughout this semester in order to work on things in the most optimal manner.

Another issue posed by the time constraint is the amount of design, testing, and debugging that must be done. We know there will be issues throughout our project that must be solved in a timely manner in order to remain on track to deliver a successful project. This may cause changes in the design of our product to scale the complexity in a higher or lower direction.

The results of the above constraints lead to stress on the expected functionality of our product and to all members of our group. We hope to follow a strict schedule and keep each other accountable in order to provide the product of our vision. As stated above, this project serves as a final demonstration to faculty and others of the knowledge and technical skills we have gained throughout our academic careers, and we will therefore provide our best efforts to deliver a product we are proud of regardless of the time constraints placed upon us.

#### **4.2.2 Usability**

Another constraint posed by our project is usability. This constraint refers to the ability of a user to learn a new product, and the satisfaction of the user with the product. In terms of any assistive technology, it is vitally important to design and

create a product with ease of use. This concept specifically applies to our project because of its purpose to aid the visually impaired.

The first design consideration of this constraint is the user interface. We must be able to design simple controls that the visually impaired can use. In addition, it must be able to give clear, intuitive feedback. With that being said, the availability of users' sight is a limiting factor in the way our product can relay information, therefore it will be important to provide feedback through different avenues such as audio and touch.

Along the same lines, our product will include navigation assistance through the use of audio. It will be important that we make the process of inputting and outputting destination data simple to avoid lengthy interactions. This would negatively impact the usefulness and satisfaction of users. Due to the difficulty of providing navigational data in an efficient manner, this could pose challenges to our group and affect the usability of our product.

Furthermore, the ergonomics and adjustability is an important aspect of our product. The market for this product includes people of various heights who will require different size canes. This means our product will need to offer some form of adjustability, which could pose an issue in the design of electrical components in the cane. In addition, components will have to be placed strategically to avoid complications from expanding or collapsing the cane.

Analyzing the above constraint, our group recognizes the potential for difficulty in creating a simple and efficient product. It is critical for us to create something that will be easy to use and learn because our aim is to provide assistive technology to the visually impaired. To alleviate this constraint, we plan to do extensive research on products and technology available that we can implement into and onto our product.

### **4.2.3 Safety**

The most important constraint that has been and will continue to be considered throughout this project is safety. Creating an assistive technology product means it helps a user complete a task that would otherwise be difficult or impossible to achieve without the product. For this project, we are taking on the responsibility of helping the visually impaired navigate from point a to point b in a safe and efficient manner. Safety must always be highly prioritized in projects such as this in order to avoid injuries to the user or others around them.

The first major design challenge caused by this constraint is required accuracy of obstacle detection. We will need to be diligent in researching different modes of obstacle detection and implementing the ones with very high accuracy to be

useful for this product. By selecting and using the wrong objection detection sensors, we would jeopardize the safety of the user and others around them.

Furthermore, the feedback of the cane needs to be accurate to avoid false alerts. These false alerts could also pose a risk to the safety of the user by directing them away from a safe path into an unsafe one. Therefore, it is important to tune the feedback of our cane to provide high accuracy feedback to avoid these possible concerns.

Another consideration posed by the safety of users is the battery life of the cane. The cane needs to be able to successfully navigate users to their destinations without a constant need to be charged. This would clearly cause a disruption in users daily lives making the cane ineffective. We therefore need to find a battery solution that alleviates this issue while selecting a battery to fit the dimensions of the cane.

As clearly stated above, the safety of our cane provides constraints upon our design but remains the highest priority to us. We need to have higher accuracy of object detection and feedback to keep users and others out of harm's way. We recognize that some design aspects may need to be changed throughout our project in order to pivot to safer solutions. It may be hard to navigate these solutions, but we plan to do extensive research into the ways we can beat the challenges presented.

#### **4.2.4. Economic**

Another constraint for this project is the budget. Our funding comes solely from group member contributions, with an estimated budget of \$100 to \$150. As a result, we have carefully analyzed the balance between cost and performance when selecting components. We prioritize certain investments that will enhance the functionality of the final product.

Our design is so simple enough where most components are easily affordable, we have chosen to spend more money on the key components that will help the device run as well as maintain user safety. For example, we chose to pay up for a more powerful microcontroller (an ESP32) that is a little pricier than its unflashed counterparts, but that features onboard PSRAM that will be used for simultaneous sensor readings, real-time audio processing, and multiple API communications. This means that the cane will be much more capable of accurately providing feedback in dynamic environments. We will also allocate resources to using high-quality sensors, including object detection sensors and gps location sensors, as they are crucial for accurate obstacle detection and assistance in navigation. So the SmartNav Cane relies on high-quality sensor data for its functionality, and there's no point in skimping on the parts that form the backbone of the whole system. We are also adopting high-end audio

parts(a digital MEMS microphone coupled with an energy-Efficient Class-D amplifier) to ensure clear and intelligible voice output, which is essential for user feedback and safe navigation. In contrast, non-essential items like physical housing, casing, and secondary attachments will be procured at lower prices or constructed in-house; freeing up capital to invest in the more impactful items. Even with limited funding, this strategy maximizes performance, reliability, and safety making the SmartNav Cane to both our technical specifications and end-user expectations.

#### **4.2.5 Operational**

An additional constraint of this project is the cane's power source. Selecting an appropriate battery requires us to balance performance, size, and affordability. The battery must be compact enough to fit inside of the cane while supplying sufficient power to the sensors, microcontroller, and speaker. Ideally, the cane should operate for multiple days on a single charge, but budget limitations may affect this goal. Therefore, a reliable charging method is essential to ensure long-term usability and convenience for the user.

We have also considered the challenges of the cane to function reliably across various terrains and weather conditions. The intended environment, specifically the University of Central Florida, includes a variety of indoor and outdoor terrains, such as multi-floor buildings, paved walkways, and unpaved paths. The cane must be durable to withstand repeated use in these settings without affecting the sensor accuracy or mobility. Additionally, Florida's frequent rain showers and thunderstorms pose a challenge, as water exposure could impact electronics, sensors, and battery performance. To address this, the cane should incorporate water-resistant enclosures and sealed electronics to ensure continued functionality even in adverse weather.

Maintaining the accuracy and functionality of integrated mapping services, such as Google Maps, is also a critical operational consideration. Since the SmartNav Cane relies on real-time map data for navigation assistance, it is essential to ensure that the mapping software is continuously updated and maintained. Regular updates from Google Maps not only provide the latest geographic and traffic information but also help in refining route guidance algorithms and integrating new points of interest. To achieve this, the cane's firmware will be designed to periodically synchronize with the Google Maps API, ensuring that any changes in road networks, construction zones, or temporary obstructions are promptly reflected. In addition, implementing error-handling protocols and redundancy measures will help manage potential connectivity issues or discrepancies in data, ensuring a seamless and reliable navigation experience for the user. This ongoing maintenance of map data is a key element in delivering safe, efficient, and up-to-date navigation assistance, thereby enhancing the overall operational effectiveness of the SmartNav Cane.

### **4.2.6 Reliability**

A further challenge presented in our project is the reliability of the cane. It must be able to consistently and accurately detect nearby objects and provide timely alerts to nearby objects in the desired path. This requires selecting high-accuracy sensors and fine-tuning them for optimal performance. Additionally, once an object is detected, the audio alerts must be clear and immediate, allowing the user to navigate safely and efficiently.

Another piece of this constraint is the durability and longevity of the cane. The components must be able to withstand repeated use without failure over long periods of time to work sufficiently. This means the electronics and sensors should not only perform reliably during extended operation but also throughout the cane's entire lifespan. To achieve this, the design must account for physical stress, temperature fluctuations, and potential moisture exposure from the environment.

### **4.2.7 Technical Constraints**

There are a number of technical constraints that influence the performance and reliability of the SmartNav Cane, and care must be taken to balance these constraints so as to ensure safe and effective operation. A major limitation is the characteristic limitation of sensors for environmental detection. When using ultrasonic, LIDAR or infrared sensors, accuracy is affected not only by the environment, but also by external factors like rain, fog, and strongly reflecting surfaces, which can damage measurement accuracy and lead to erroneous readings. Such a discrepancy is dangerous from the user perspective so frequent calibration is necessary alongside careful calibration and error-checking to ensure accuracy of sensors affecting overall device reliability.

Battery life is another important technical limitation. Since the SmartNav Cane is an assistive device for visually impaired people, it would need to have extended periods of use without much downtime due to needing to be charged. Small battery capacity can spoil the user experience and limit the cane at the most needed moment. Hence, choosing energy-efficient components as well as optimizing power management strategies are key to maximizing running time and keeping the device reliable during a regular use day.

Processing speed is just as crucial, with a need for the microcontroller to quickly interpret real-time sensor data and response. Processing delays can get in the way of the cane's ability to quickly address obstacles and safely lead a user through an environment. In addition to sensor integration, performing complex operations like wireless communication and interaction with APIs is also integral

for providing navigation assistance; hence, in order to achieve this, the card needs to have high-speed data processing capabilities.

These technical limitations are compounded by connectivity challenges. His reliance on wireless technologies including GPS, Bluetooth or Wi-Fi means that performance can be compromised in areas with weak or erratic signal strength. If the connection is not stable, it can result in data transfer delays or major interruptions that will hurt the overall functionality and user experiences. To address these issues, several approaches have been proposed. For instance, pairing LIDAR with ultrasonic sensors provides redundancy, improving detection accuracy by covering each other's blind spots. Low-power, high-efficiency microcontrollers, along with optimized power management circuits, are employed to enhance battery life – allowing the cane to stay operational for extended periods of time. Furthermore, edge computing is employed to analyze sensor information locally, minimizing reliance on cloud-based systems and enhancing response times. Finally, we integrate robust protocols for connectivity and error handling, to guarantee minimal impact of communication disruptions on the system performance. The cane you create must be able to overcome these technical constraints in order to be effective and provide real-time navigation so with careful component selection and advanced design strategies the SmartNav Cane is designed to overcome these challenges. By employing a holistic solution with a mix of both software and hardware components, it mitigates the effects of sensor error, battery life, computation lag, and connectivity issues while maximizing user safety and utility to the visually impaired.

## **Chapter 5: Comparison of Chat GPT with Similar Platforms**

### **5.1 The Role of AI in the development of our project**

Artificial intelligence has been transforming the way researchers and engineers approach problem-solving. In the development phase of our product, AI tools such as ChatGPT, Google Gemini, and Microsoft Copilot have played a role in providing guidance, troubleshooting, and optimizing various aspects of our design. These platforms assist with coding, circuit design, and debugging sensor integration, which are crucial for implementing features like object detection and GPS navigation. However, while AI-generated solutions can speed up development, they also come with limitations, such as the potential for inaccurate responses and the inability to interact with hardware directly. With this chapter we will see how ChatGPT can be helpful, however we will showcase the limitations as we dive deep into this program and view both the positives and negatives of ChatGPT.

## 5.2 ChatGPT

ChatGPT was created by OpenAI and launched in November of 2022. Its main purpose is to assist with text-based tasks such as answering questions, generating content, and providing recommendations. This model is based on a Generative Pre-Trained Transformer (hence GPT), meaning it generates responses using knowledge acquired during its extensive pre-training. Specifically, ChatGPT is trained on large datasets, including books, websites, and articles, to learn language patterns, grammar, facts, and contextual information. This pre-trained knowledge enables it to generate relevant and accurate responses to user prompts. The model is intended to efficiently comprehend, construct, and respond to human text inputs in a conversational style.

### Key Features of ChatGPT:

- **Conversational AI:** Can engage in dialogues like humans across a variety of topics.
- **Knowledge-Based Assistance:** It offers information on Fields such as science, engineering and business.
- **Processing Text:** It generates reports, emails and other documents with efficiency.
- **Code Assistance:** It assists in programming, debugging and developing software.
- **Personalized Learning:** It helps students and professionals learn new topics.

ChatGPT sees widespread use in content generation, customer service, coding support and educational backing. This makes it A valuable card in the stack for researchers, students and businesses. The contribution this tool makes can't be overlooked.

## 5.3 Comparison of AI tools: (ChatGPT, Google Gemini, Microsoft Copilot)

When developing SmartNav, we explored various AI tools to assist with different aspects of our project, including coding, troubleshooting, and research. ChatGPT, Google Gemini, and Microsoft Copilot each offer unique features that cater to specific needs in our development process. ChatGPT has been highly effective for explaining technical concepts, generating sample code, and seems to be very effective with troubleshooting issues that will be related to our microcontroller and sensor integration. Now when it comes to Google Gemini, on the other hand this application has the advantage of multimodal capabilities, which will allow us to

analyze both text and images, which will be beneficial when working with schematics, circuit diagrams, and visual debugging. However, its responses can sometimes lack depth, particular in complex scenarios. Now with Microsoft Copilot this is directly integrated into certain applications like one we are familiar with which is visual studio code, which can make it a valuable tool for real-time code suggestions and autocomplete features. It may speed up the process while helping us with syntax issues, but the issue is this is not always optimized for embedded systems, which is crucial for our project. After Evaluating these AI tools, we decided to focus primarily on ChatGPT due to its versatility in providing in-depth explanations, helping us troubleshoot coding errors, and assisting with research throughout the development process. Unlike Copilot, which is more code-focused, or Gemini, which is useful for visual analysis but sometimes lacks technical detail. For example, it has helped us determine the best object detection sensor by comparing Time-of-Flight (ToF) sensors, LiDAR, Ultrasonic sensors, and mmWave technology, breaking down the advantages and trade-offs of each one. Additionally, ChatGPT has assisted in implementing navigation API integration by explaining the differences between Google Maps API, OpenStreetMAP API, and Mapbox Directions API, helping us come to a conclusion on what is the most suitable option for GPS navigation. Its ability to help us break down complex concepts, suggest alternative approaches, and guide us through debugging voltage regulator issues as well as optimize power efficiency has made this application essential to our project progression. While we recognize the other AI tools, ChatGPT remains our primary choice for problem-solving and decision making while creating our project SmartNav.

## **5.4 Pros of utilizing ChatGPT**

ChatGPT provides numerous benefits that contribute to the successful development of our SmartNav Cane, enhancing both the design and implementation phases. ChatGPTs ability to assist in troubleshooting technical challenges, allowing us to identify and resolve issues efficiently when working with hardware components such as GPS modules, object detection sensors, and power management circuits. By offering different approaches to problem-solving, ChatGPT enables us to define our system architecture, optimize microcontroller programming, and debug firmware issues more effectively. Just simply having the ability to analyze and simplify complex technical documentation has been particularly a huge help in understanding new components like the NEO-M9N GNSS module and the Sharp GP2Y0A710K0F infrared distance sensor, ensuring we understand the pros and drawbacks as well as guiding us to have proper integration into our smart walking stick.

### **5.5.1 Troubleshooting**

ChatGPT can be an important part in assisting through the developmental phase of building SmartNav, in helping us resolve both immediate and technical issues and challenges that arise as we progress. When we encounter problems with sensor integration, such as incorrect obstacle detection or inaccurate GPS readings, we can use ChatGPT to help us analyze potential causes as well as suggestions on possible debugging steps. Not only would it provide step by step, it will provide us with explanations for common hardware and software errors, allowing us to diagnose issues efficiently which will help us save time not having to deal with an extensive trial and error. Additionally, ChatGPT assists in resolving coding problems by giving us solutions for microcontroller programming, debugging annoying syntax errors, and keeping our code as optimized as possible. As we progress in our project through the different stages of development, ChatGPT continues to be useful by helping us anticipate potential problems before they even occur. For instance, it can suggest best practices for wiring and power management, which will reduce the likelihood of connectivity issues. It also provides real-time guidance for troubleshooting communication between components like ultrasonic sensors, GPS modules, and the MSP-EXP430G2ET microcontroller, helping us ensure a smooth system integration. By providing us with step-by-step solutions, recommending alternative solutions, and clarifying technical documentation, ChatGPT not only speeds up problem resolution but now enhances our understanding on a specific subject. With this support we are able to streamline our workflow, allowing us to focus on more pressing matters rather than being stuck on troubleshooting longer than we have too.

### **5.5.2 Technical Aspect**

Since this is our first time developing a smart walking stick with some type of embedded system and sensor integration, ChatGPT will play a crucial role in bridging the gap in our technical knowledge. It will provide instant access to explanations, tutorials, and best practices. With our lack of experience we will most certainly run into unfamiliar concepts-such as dealing with communication between the MSP-EXP430G2ET and external modules, However with the use of chatGPT it will break down complex information into simpler terms, making it easier to understand. It will also suggest ways to improve our efficiency and implement key features dealing with obstacle detection, GPS navigation, and power management. Beyond that it will also help us interpret circuit diagrams, choose the right components based on certain specifications, and help us with hardware compatibility issues. In essence this will basically act as an always-available mentor. With this being alway accessible it will speed up our learning process and help us build a more reliable and functional smart walking stick, even without prior experience.

In addition, ChatGPT will play a vital role in recommending components for our project. As students, we may not always have the technical expertise to select

components without extensive research. ChatGPT serves as a valuable starting point by providing recommendations based on project requirements and goals. This allows us to narrow our options and conduct more focused research to ensure the best choices for performance, efficiency, and cost-effectiveness.

### **5.5.3 Allocation of Different Resources**

One of the biggest advantages of using ChatGPT in our project would be the ability to provide us with quick access to a wide range of resources. Instead of spending hours searching through forums, technical manuals, or research papers, we can get relevant information almost instantly. Whether we need a datasheet for our sensors, sample code for integrating GPS and ultrasonic modules, or explanation of electrical engineering concepts, ChatGPT helps us find and understand the necessary materials. Additionally, it can summarize complex research papers, extract key information from documentation, as well as provide alternative solutions if we run into component limitations or compatibility issues. Also Beyond using technical documentation, ChatGPT also connects us to industry best practices, Since we are new and not familiar it will help lead us and provide insights to efficient circuit design, power management, and sensor placement strategies that we might have not considered. With this we can even recommend open-source projects, development tools, and software libraries that align with our design needs. By having easy access to this pool of information we will be able to bridge gaps in our knowledge and confidently deal with whatever design aspect we are having issues with or just simply trying to solve.

### **5.5.4 Refined Writing**

ChatGPT has also been a valuable resource in the writing process. While no content in this document has been directly copied from ChatGPT, it has been used to guide the flow and clarity of certain sections, ensuring a more professional and non-repetitive tone. All content in this document has been written by the authors unless otherwise noted.

### **5.5.5 Quick Responses**

One of the major advantages of using ChatGPT in our SmartNav project is its ability to provide instant responses, which significantly reduces the time spent searching for technical solutions. Since this is our first time dealing with such complex components we are going to struggle to integrate multiple components such as the ESP32 microcontroller, LiDAR and ultrasonic sensors, GNSS modules, and power management circuits. Having quick access to relevant information is crucial in allowing the project to have a continuous workflow. Additionally, ChatGPT helps us compare different hardware options in real time, helping us make decisions on selecting the most suitable component. For example, when evaluating various GNSS modules for navigation, we were able

to quickly obtain comparisons between NEO-M9N, MAX-M10S, and ZED-F9P, helping us determine which module best balances accuracy, power consumption, and cost for our project. Similarly, when figuring out the best object detection sensor for obstacle avoidance, ChatGPT provided immediate breakdowns of the strengths and weaknesses of Time-of-Flight (ToF), LiDAR, and mmWave sensors, helping us make a decision on which route to take. Beyond hardware selection, ChatGPT has proven invaluable in the software aspect as well. While we started to look into navigation API integration, such as the Google Maps API or OpenStreetMap API, we were able to quickly identify all the drawbacks we would encounter by using either one. This saved us time and accelerated our development process. Having rapid access to information ensures that our team is able to focus more on innovation rather than spending an excessive amount of time on areas that are not as necessary.

### **5.6.6 Personalized Learning and Accessibility for visually Impaired Users.**

In the Role of Interactive tutor and tool for accessibility, ChatGPT has a lot to offer. ChatGPT offers significant advantages for both research and assistive technology, particularly in our SmartNav project. We are able to enhance our voice assistant capabilities, making navigation more intuitive and accessible for visually impaired users.

- **Assisting Students & Researchers** - ChatGPT provides in-depth explanations of research topics, technical concepts, and real-world applications. Additionally, it has assisted in comparing various speech-to-text (STT) and text-to-speech (TTS) technologies, guiding our decision-making process in selecting OpenAI Whisper API, Google Speech-to-Text API, or Azure Speech Service to enhance SmartNav's voice interaction.
- **Text-to-Speech (TTS) Conversion** - One of the key accessibility features of our product SmartNav Cane is the ability to convert navigation directions, obstacle alerts, and real-time GPS updates into speech. ChatGPT can aid us in text-to-speech by offering us guidance on selecting the best audio output speakers (such as Adafruit Mini or Dayton Audio CE Series) and give us all the drawbacks with each product allowing us to make the best decision.
- **Use Case In SmartNav Cane** - Speech synthesis and response generation play a crucial role in making the SmartNav Cane more accessible and user-friendly. By using ChatGPT we can gain insight to enhance the device's real-time audio feedback, ensuring users receive clear, concise, and context-aware navigation instructions. For example, instead of simply stating "Obstacle ahead", SmartNav could provide dynamic responses like:
  - "Caution: An object is detected, please veer slightly to the right."

- “You are approaching a crosswalk. Press the button to request to cross the road”

### **5.5.7 Cost-Effective and Scalable Solutions**

As college students developing SmartNav, budget constraints are a major factor in our decision-making process. ChatGPT has been invaluable in helping us find cost-effective alternatives for various components, ensuring our project remains affordable while maintaining high functionality. A big issue we face is balancing performance with cost. When dealing with High-end navigation systems such as LiDAR and advanced GNSS modules, We are able to choose budget-friendly alternatives without compromising reliability. For example, instead of using high-cost LiDAR sensors, ChatGPT helped us explore the feasibility of using Time-of-Flight (ToF) sensors and ultrasonic sensors, which provide adequate accuracy at a lower price point. ChatGPT also assisted in comparing models like the NEO-6M versus the more expensive ZED-F9P, helping us determine the best choice based on accuracy, power consumption, and affordability.

## **5.6 Negative aspect of using ChatGPT**

Despite the many benefits ChatGPT offers, there are also several limitations that must be considered. While this AI tool has helped us explore ideas, troubleshoot problems, and better understand unfamiliar technologies, it cannot fully replace the precision, depth, and critical thinking required in technical design. ChatGPT operates in a virtual environment, meaning it lacks physical awareness and firsthand interaction with the hardware components we’re using- such as sensors, microcontrollers, and GPS modules. With this issue we can run into issues where the guidance we receive is too generalized or just simply not applicable to real-world scenarios. These drawbacks highlight the importance of using ChatGPT as a supplementary tool, rather than a sole source.yes

### **5.6.1 Virtual Limitation**

One of the biggest drawbacks of using ChatGPT while we continue with our project SmartNav, would be its virtual limitation. It can provide guidance, explanations, and even code suggestions however, it lacks the ability to interact with physical hardware. Unlike a human mentor who can test circuits, inspect wiring, or physically debug sensor malfunctions, ChatGPT can only offer theoretical solutions based on the information we provide. This limitation means that if we encounter an issue that requires hands-on troubleshooting, such as loose connections or unexpected hardware behavior, we must rely on our own problem-solving skills and testing methods. This means certain errors may only

become apparent during implementation. So when we are in the field ChatGPT may lack the ability to see how certain environmental factors may affect the performance of our device. Because of this as we progress we will need to carefully verify the accuracy of our product responses and conduct thorough testing on our own. Continuing with environmental factors, ChatGPT will not always account for the variability of real-world conditions, such as interference with GPS signals or sensor inaccuracies. While it is a useful tool for guidance, it cannot replace hands-on experience and physical testing that comes with working directly with the hardware. This just highlights the importance of balancing ChatGPTs assistance with practical experimentation to ensure the reliability and functionality of SmartNav.

### **5.6.2 Possible Incorrect Information**

Another Challenge we've faced while using ChatGPT during SmartNav development is the risk of receiving incorrect or outdated information. Because ChatGPT generates responses based on patterns in the data, it could sometimes provide answers that are either oversimplified, not fully accurate, or no longer applicable. For instance, while researching different GPS modules (like the NEO-M9N or ZED-F9P) we noticed that ChatGPT occasionally mixed up specifications. Relying on this information without double-checking could lead to selecting components that don't meet our design requirements as well as our budget constraints. Similarly when configuring software environments (such as PlatformIO or ESP-IDF), ChatGPT sometimes omitted key steps or referenced older library versions. In a project like ours where precision and reliability directly affect user safety, it is essential that we validate all AI-generated suggestions.

## **5.7 Case Studies**

This section provides case studies in an effort to show specific examples of AI being used in our project. Specifically, these studies aim to represent all of the positive and negative qualities listed above in a more direct correlation with our project. We have written a short description below each to discuss these topics.

**Case study 1:** Question “What are some different object detection technologies we can research for our smart cane?”

### **1. Ultrasonic Sensors (SONAR)**

- How it Works: Uses sound waves to detect objects by measuring the time delay of the reflected signal.
- Pros:
  - Affordable and widely available
  - Works well in various lighting conditions
  - Effective for detecting objects in close range (0.2m–4m)

- Cons:
  - Limited accuracy for small or thin objects (e.g., wires)
  - Struggles with certain materials (e.g., sound-absorbing fabrics)

Example Sensor: HC-SR04, MaxBotix MB1240

## 2. LiDAR (Light Detection and Ranging)

- How it Works: Uses laser pulses to measure distances and create a depth map of the environment.
- Pros:
  - High precision (millimeter-level accuracy)
  - Works well in dark or low-light environments
  - Can detect objects at longer distances (up to 40m)
- Cons:
  - More expensive than ultrasonic sensors
  - Can struggle in fog, rain, or highly reflective surfaces

Example Sensor: RPLiDAR A1, Garmin LIDAR-Lite v4

## 3. Infrared (IR) Sensors

- How it Works: Uses infrared light to detect nearby objects based on reflected IR signals.
- Pros:
  - Low-cost and simple to integrate
  - Works well for short-range detection (5cm–80cm)
- Cons:
  - Limited range and accuracy
  - Ineffective in bright sunlight or highly reflective surfaces

Example Sensor: Sharp GP2Y0A21YK0F

## 4. Computer Vision (AI-Based Cameras)

- How it Works: Uses a camera and AI models (such as YOLO, OpenCV, or TensorFlow) to recognize objects and their distance.
- Pros:
  - Can detect and classify objects (e.g., stairs, people, obstacles)
  - Works in various environments with real-time processing
  - Can integrate with voice assistance for verbal feedback
- Cons:
  - Requires more processing power (e.g., Raspberry Pi or Edge AI modules)
  - May struggle in low-light conditions without additional IR support

Example Hardware: Raspberry Pi with OpenCV, Intel RealSense D435

## 5. mmWave Radar (Millimeter-Wave Radar)

- How it Works: Uses radio waves in the millimeter frequency range (e.g., 24GHz or 60GHz) to detect objects, distance, and motion.
- Pros:
  - Works in all weather and lighting conditions
  - Can detect multiple objects and moving targets
  - High accuracy for short-range detection (up to 10m)
- Cons:
  - More complex to integrate and process data
  - Higher cost compared to ultrasonic or IR sensors

Example Sensor: Texas Instruments IWR6843, Infineon XENSIV BGT60TR13C

### Takeaway:

This response proved to be very beneficial in the early stages of research. It provided us with multiple potential options of technologies, advantages / disadvantages of each, and possible sensors we could use. Although this response was helpful in the early stages, more extensive research had to be done to determine which technology would best suit our project. Upon this determination, we began research on sensors this response had provided, but continued research on other sensors to get a more complete idea of our options. This research was done mostly through online google searches that provided scholarly articles on these sensors. To obtain more specific data, the sensor datasheets were consulted.

**Case study 2:** Question “What are some low-power components and ways we can extend battery life?”

## Low-Power Components

### *Microcontrollers*

- **MSP430, STM32L, nRF52, ESP32-S2/S3** – all support low-power and sleep modes.

### *Sensors*

- **Ultrasonic:** MaxBotix
- **IMUs:** MPU6050, BNO055
- **IR/Proximity:** VL53L0X
- **GPS:** u-blox Neo-M8N (supports low-power mode)

### *Communication*

- **Bluetooth Low Energy (BLE)**
- **LoRa** for long-range, low-data apps

### *Feedback/UI*

- Low-power **LEDs** and **e-Paper** or **OLED displays**

### *Battery Life Strategies*

- **Sleep Modes:** Put MCU/sensors to sleep when idle
- **Duty Cycling:** Activate components only as needed
- **Efficient Code:** Use interrupts, avoid polling
- **Smart Power Circuitry:** Use LDOs, battery management ICs, fuel gauges
- **Eco Features:** Power saver modes, minimal feedback when inactive

### **Takeaway:**

In addressing the question “What are some low-power components and ways we can extend battery life?”, we engaged LL ChatGPT as a resource to generate potential solutions for our SmartNav project. While the tool provided a comprehensive list of low-power components such as the MSP430, STM32L, nRF52, and ESP32-S2/S3 microcontrollers, various sensors like the MaxBotix ultrasonic sensor, MPU6050, BNO055, VL53L0X, and even the u-blox Neo-M8N GPS module, along with communication options like BLE and LoRa, and feedback methods including low-power LEDs and OLED displays it ultimately fell short of fully integrating these suggestions with the specific power demands and operational intricacies of our SmartNav setup. The recommendations, although thorough in a general sense, did not account for the unique interplay between our sensor fusion requirements, real-time navigation demands, and the custom power management architecture we have developed. This limitation became apparent after multiple prompts when LL ChatGPT was unable to precisely connect the proposed low-power components and battery extension strategies to our detailed design parameters. The experience highlighted the need for a more tailored, in-depth analysis that considers our product's specific power profiles, duty cycles, and environmental factors. In conclusion, while LL ChatGPT provided an excellent starting framework for low-power design considerations, the specialized nature of our SmartNav project necessitates further expert intervention to align these components with our exact operational and performance requirements.

**Case Study #3:** “How do I regulate voltage for multiple components in an embedded system?”

Our SmartNav Cane integrates multiple components that require operating voltages:

- GPS module - 3.3V
- Ultrasonic Sensors - 5V
- MSP430 microcontroller - 3.3V
- Voice Module (DFPlayer Mini) - 3.3V to 5V range

Now we need a way to power all the components we may use from a single battery source while maintaining stable and isolated voltage lines.

We were recommended:

- Using buck converters (e.g., LM7805 for 5V, AMS1117 for 3.3V)
- Verifying current requirements for each device
- Including bypass capacitors to reduce voltage ripple
- Considering heat dissipation in linear regulators

How this helped:

- Helps us understand which regulators are commonly used in hobby projects
- Identify that linear regulators like AMS1117 were easy to implement for 3.3V devices
- Choose modules that have been built-in voltage regulation when possible

But it didn't provide:

- Pinout diagrams or wiring examples
- Insight into real-world voltage drops under load
- Suggestions on where to purchase the best regulator boards

#### Further Research Using Other Tools:

To ensure the accuracy of our voltage regulator strategy for our project, we expanded beyond ChatGPT and turned to several other tools for verification. While ChatGPT helps in our understanding of general concepts like regulating 5V down to 3.3V, we needed more specific reliable data for our components. For example:

- Official datasheets (e.g., from Texas instruments for the MSP430) confirmed exact voltage tolerances and current requirements.
- Google and technical forums like Stack Exchange and EEVblog provided insights on voltage regulator types, power distribution, and practical circuit layout issues.
- Vendor websites such as SparkFun, Adafruit, and Digi-key offered detailed product pages, pinouts, and usage notes for the GPS module, ultrasonic sensor, and voltage regulator. This combination of sources allowed us to cross-check ChatGPT's suggestions. Relying solely on AI would have not accounted for the variations in hardware, so supplementing with research gave us greater confidence in our final component selection.



## 6.2 Power Delivery

Our SmartNav design requires a power supply unit as a critical subsystem for reliable, stable operation of use all hardware components. The system is based on one cell Li-ion battery with a nominal voltage of 3.7V, which is appropriate for proper energy density and portability. In fact, various parts of the SmartNav have different operating voltages, for example the microcontroller (MCU) is 3.3V and other sensors like the Sharp GP2Y0A710K0F require a regulated 5V supply. To support these needs, we incorporated specific DC-DC converters into our design: a buck converter to reduce battery voltage to 3.3V for the MCU and other low-voltage peripherals; as well as a boost converter using the Pololu 5V Step-Up Voltage Regulator to increase the battery voltage to 5V as required. Our power management scheme, demonstrated in the subsystem block diagram, plays a critical role in ensuring efficiency and sustaining voltage stability throughout the device an important factor for the general performance and reliability of the in-house produced SmartNav.

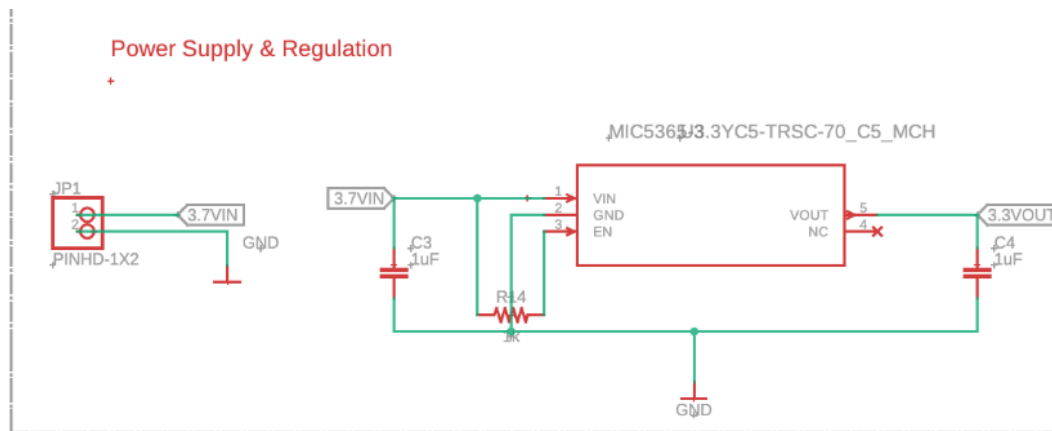


**Figure 6.2.** 3.3V Regulator Schematic.

The boost converter is a key component in our SmartNav design, responsible for stepping up the battery voltage from 3.7V to the required 5V for certain sensors and modules. By efficiently converting lower voltage from our Li-ion battery to a stable, higher output, the boost converter ensures that all critical components receive the correct power levels despite battery discharge. Its high efficiency and robust performance under moderate loads makes it ideal for portable applications, though careful attention must be paid to managing any switching noise.

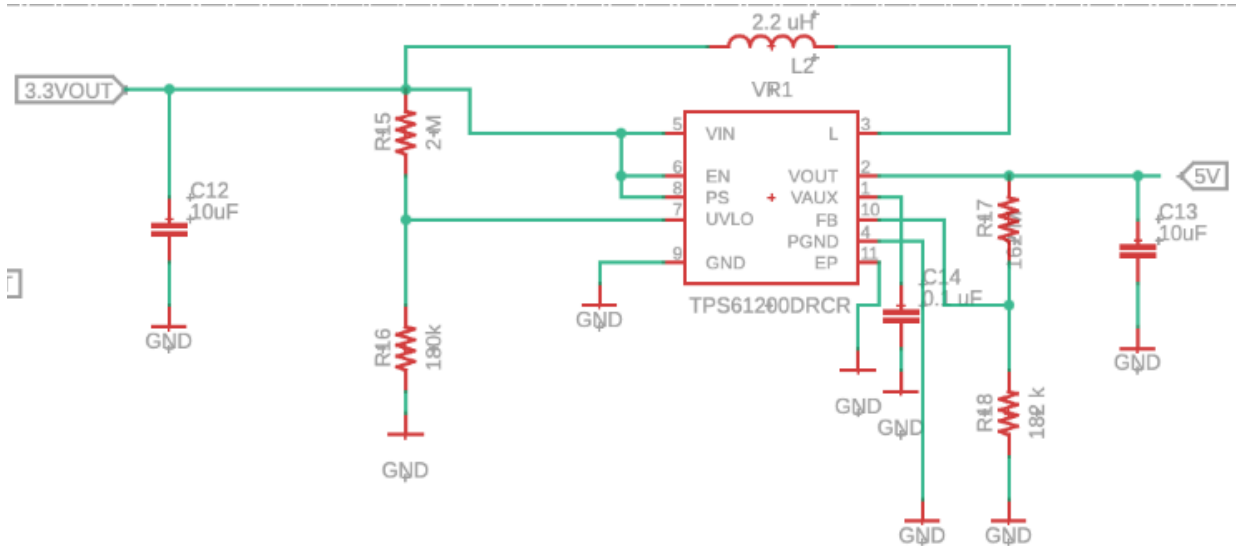


**Figure 6.3. 5V Step-Up Voltage Regulator Schematic.**

The image above shows the Pololu schematic for the step-up voltage regulator. It uses the TPS6120X boost converter chip along with various external components such as inductors, capacitors, and resistors to create a fully functional circuit that efficiently steps up the lower battery voltage to a stable 5V output.

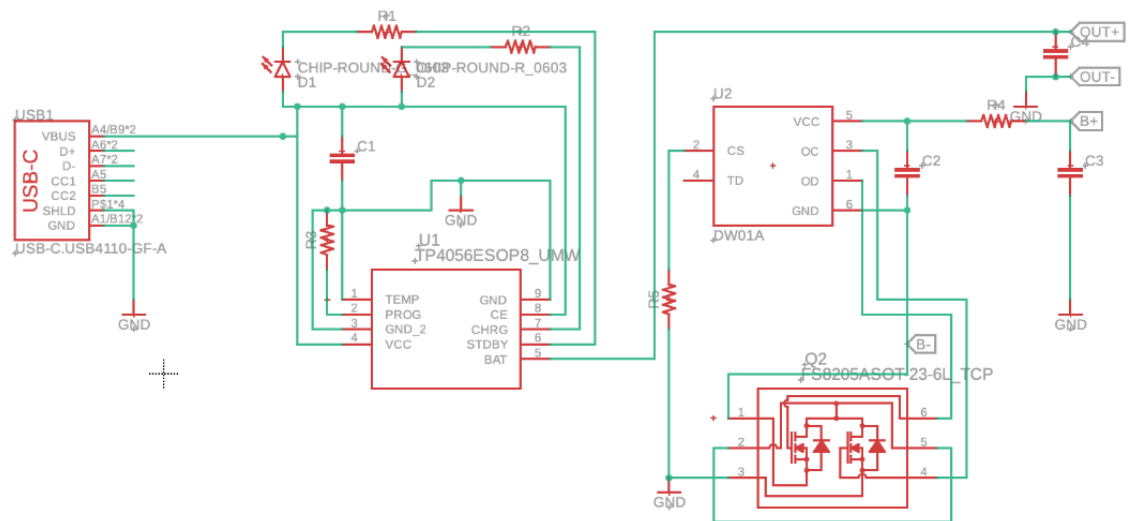


**Figure 6.4. TP4056 with External Modules Schematic.**

The TP4056 schematic we incorporated into our design plays a pivotal role in managing the charging of our single-cell Li-ion battery. This chip is responsible for ensuring safe, efficient, and controlled charging by regulating the current and voltage from a USB power source, thereby protecting the battery from overcharge and deep discharge. Integrated with essential external components such as resistors, capacitors, and a USB connector, the TP4056 circuit provides a complete charging solution on our custom PCB. By converting the 5V input from the USB into the appropriate charging conditions for our 3.7V Li-ion battery, it enables continuous, reliable operation of the SmartNav device. This ensures that the power supply remains robust and efficient, which is critical for maintaining the performance of all sensor, MCU, and communication modules within our portable system.

### 6.3 USB to UART Bridge

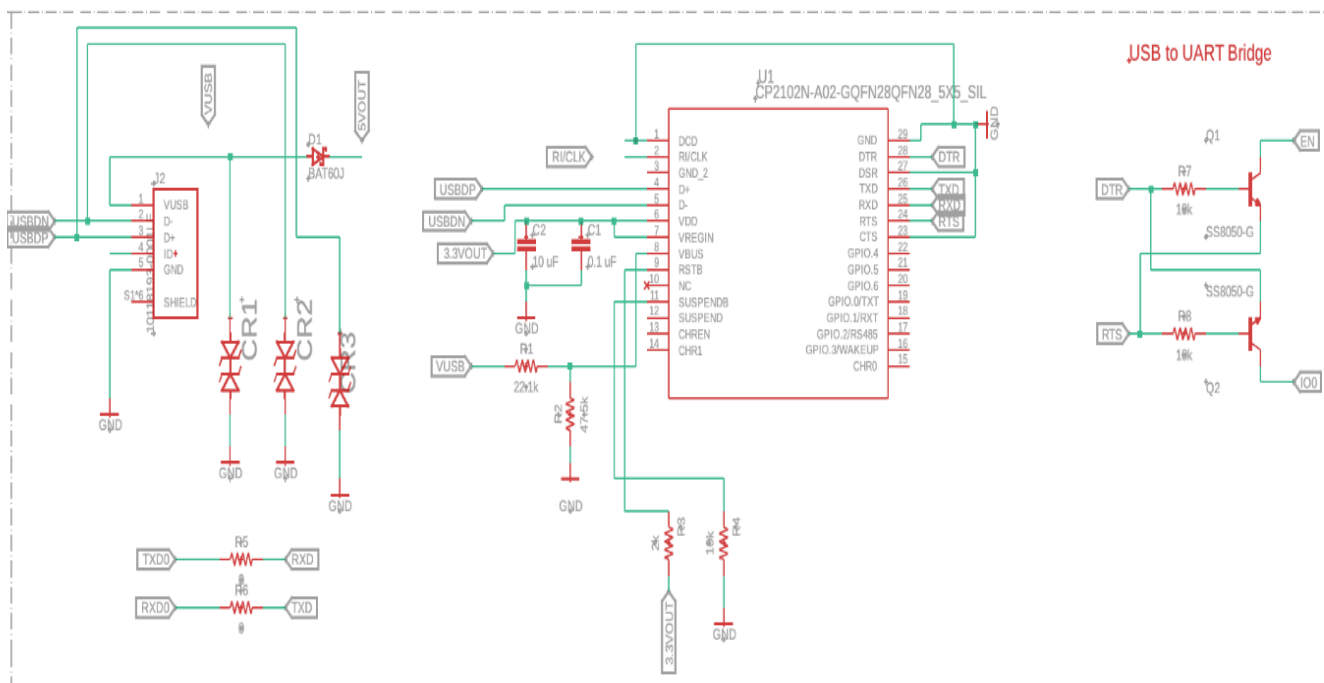


**Figure 6.5.** *USB to UART Bridge with External Modules Schematic.*

The USB-to-UART bridge schematic featuring the CP2102 is a critical component of the SmartNav PCB; our ESP 32 wroom does not contain an internal bridge so it was necessary to implement one using the ESP 32 wroom schematic as reference. This enables seamless communication between the board's microcontroller and external systems via USB. This schematic integrates the CP2102 along with necessary supporting components such as decoupling capacitors, ESD protection diodes, and filters to ensure robust and stable serial

data transfer. By converting USB signals to UART, it not only facilitates programming and debugging of the ESP32 MCU but also supports ongoing communication during operation. The design intentionally leaves optional hardware flow control pins, such as CTS and DSR, unconnected since software flow control suffices for our application, simplifying the overall design. Overall, this CP2102-based bridge plays a pivotal role in maintaining reliable connectivity and ease of development for our SmartNav project.

## 6.4 Button Interface

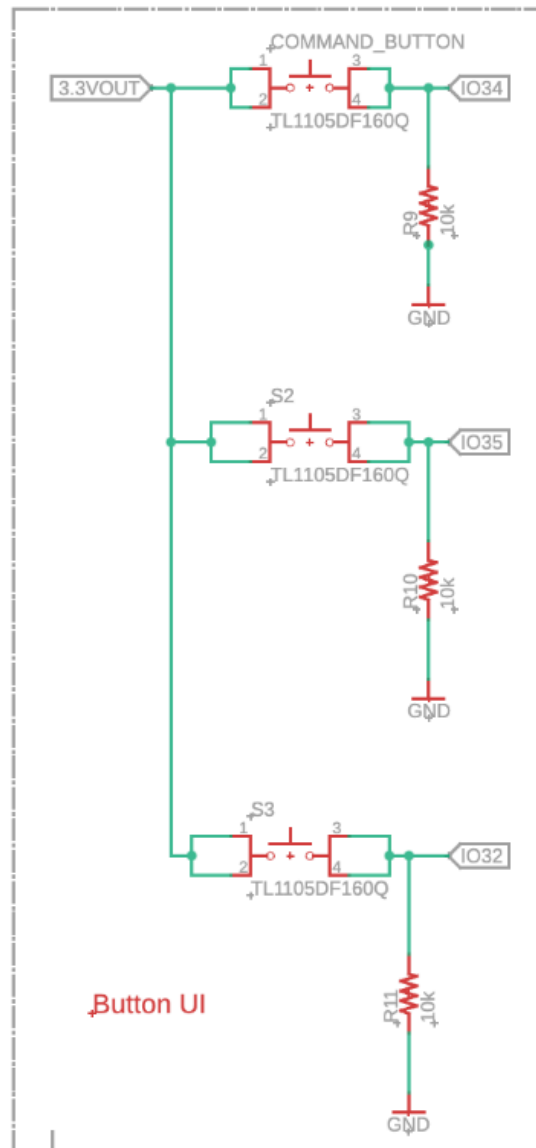


**Figure 6.6.** Buttons and resistors for UI commands Schematic.

In our SmartNav design, the button interface is carefully engineered to facilitate key functions such as displaying battery level and capturing audio command inputs. Each button is connected to a dedicated MCU input, with an external pull-down resistor ensuring a stable low logic state when not activated. When a button is pressed, it pulls the input high, triggering the corresponding function in our system. This approach provides reliable, noise-free signal detection while maintaining a consistent interface across the device.

## 6.7 Sensors

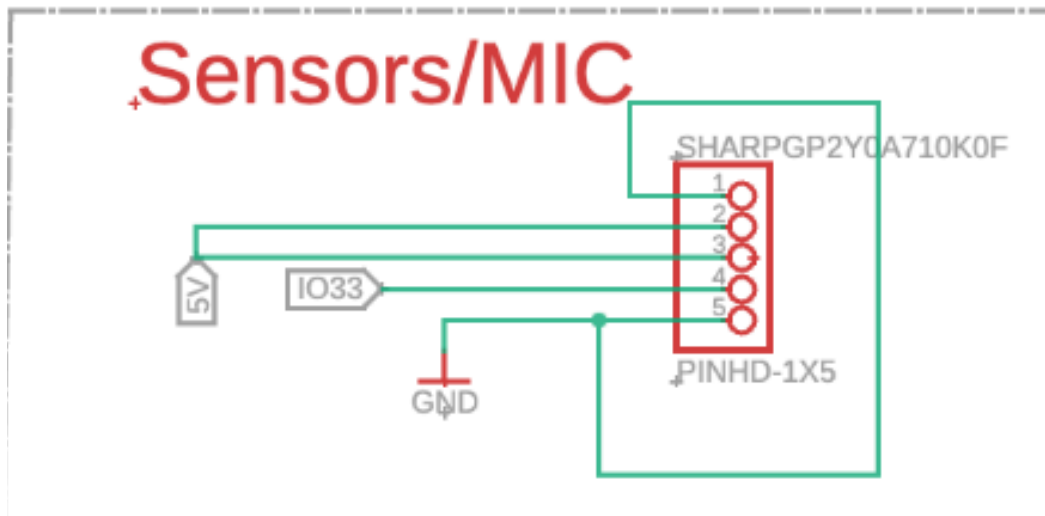


**Figure 6.7.** Sharp GP2Y0A710K0F Sensor Pin Header Schematic.

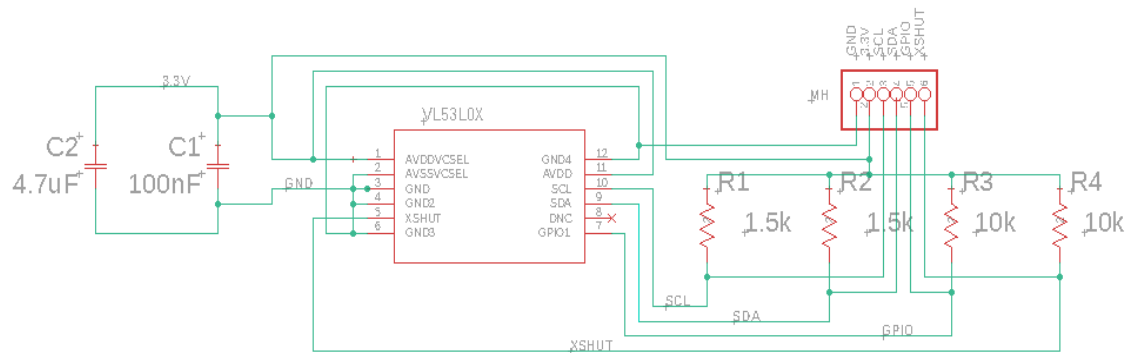


**Figure 6.8.** VL53L0X Time-of-Flight Sensor Schematic.

In our SmartNav schematic, we integrated dedicated pin headers that interface directly with the custom PCB for the STMicroelectronics VL53L0X and the breakout board for the Sharp GP2Y0A710K0F sensors. We chose to custom build one sensor PCB and utilize one off-the-shelf module to challenge us but maintain reasonable goals for our project. While the custom PCB required much research, time, and effort, the off-the-shelf module provides access to all the essential supporting components such as power conditioning, filtering, and level shifting ensuring reliable operation while simplifying our design. The VL53L0X module is strategically mounted at the bottom of the cane to capture precise, ground-level obstacle data with its narrow, high-accuracy field of view, while the Sharp GP2Y0A710K0F module is positioned at the top to provide a broader detection range for more distant hazards. Although these sensors are placed at different locations from the main PCB to optimize accuracy, they are permanently connected via soldered header pins and carefully routed wiring. This ensures robust, stable electrical connectivity and proper signal integrity. Overall, this modular design approach allows us to leverage proven sensor technologies while precisely positioning each sensor to achieve the high level of performance required for the SmartNav system.

Specifically, the custom PCB for the STMicroelectronics VL53L0X includes a 6-pin header for ground, input voltage, I2C serial clock (SCA) and data lines (SDA), digital input (XSHUT), and digital output (GPIO). This header connects to the main PCB located at the top of the cane that houses the 3.3 voltage regulator and MCU. Two 1.5K Ohm resistors have been placed onto the serial clock and data lines while two 10K Ohm resistors have been placed onto the digital input and output lines. In addition, a 4.7uF and 100 nF capacitor have been placed as close as possible to the input voltage and ground of the sensor to provide stability and noise filtration.

## 6.8 MIC and Speaker Schematics

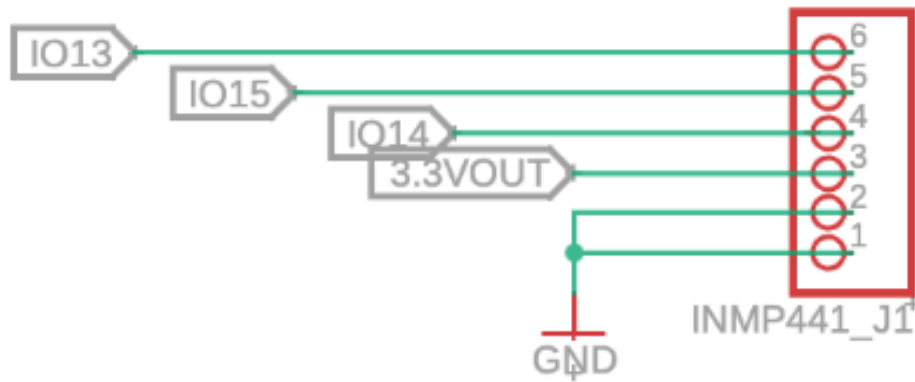


**Figure 6.9.** Pin header for the NMP441 Microphone Module.

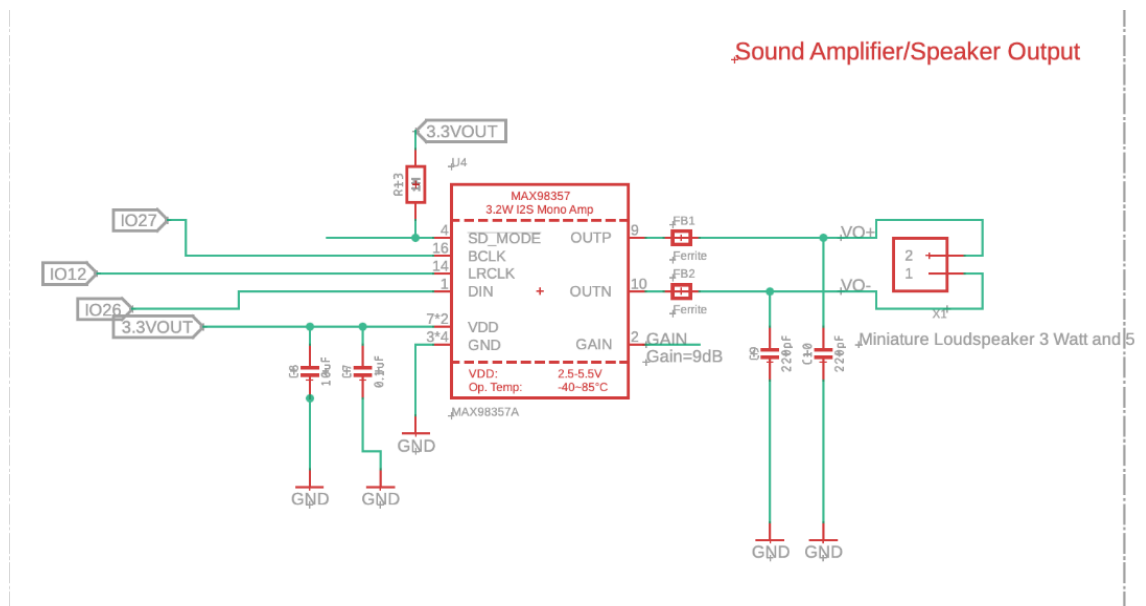


**Figure 6.10.** MAX98357A Amplifier connected to Speaker Schematics.

In our Smart NAV stick PCB, we integrated two crucial schematics to deliver high-quality audio functionality in a compact and modular design: one for the INMP441 microphone and one for the MAX98357 amplifier paired with a PUI Audio 3W 4Ω Mini Loudspeaker. The INMP441 mic schematic features a



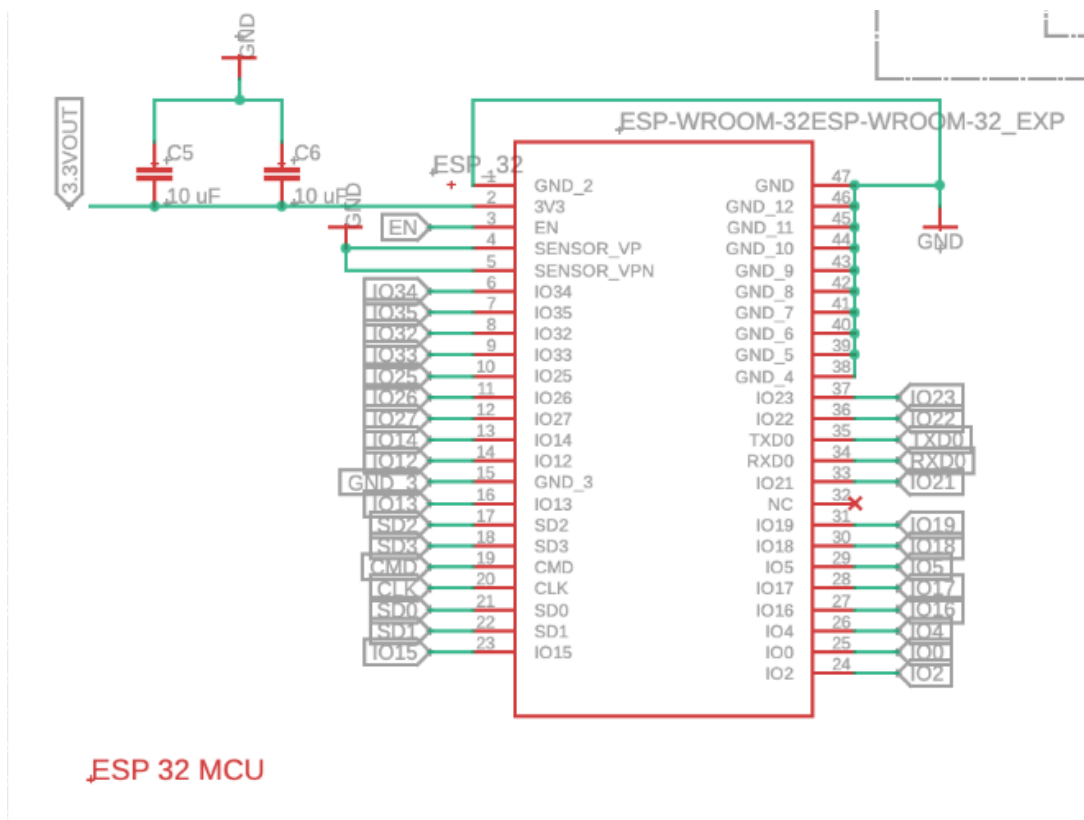
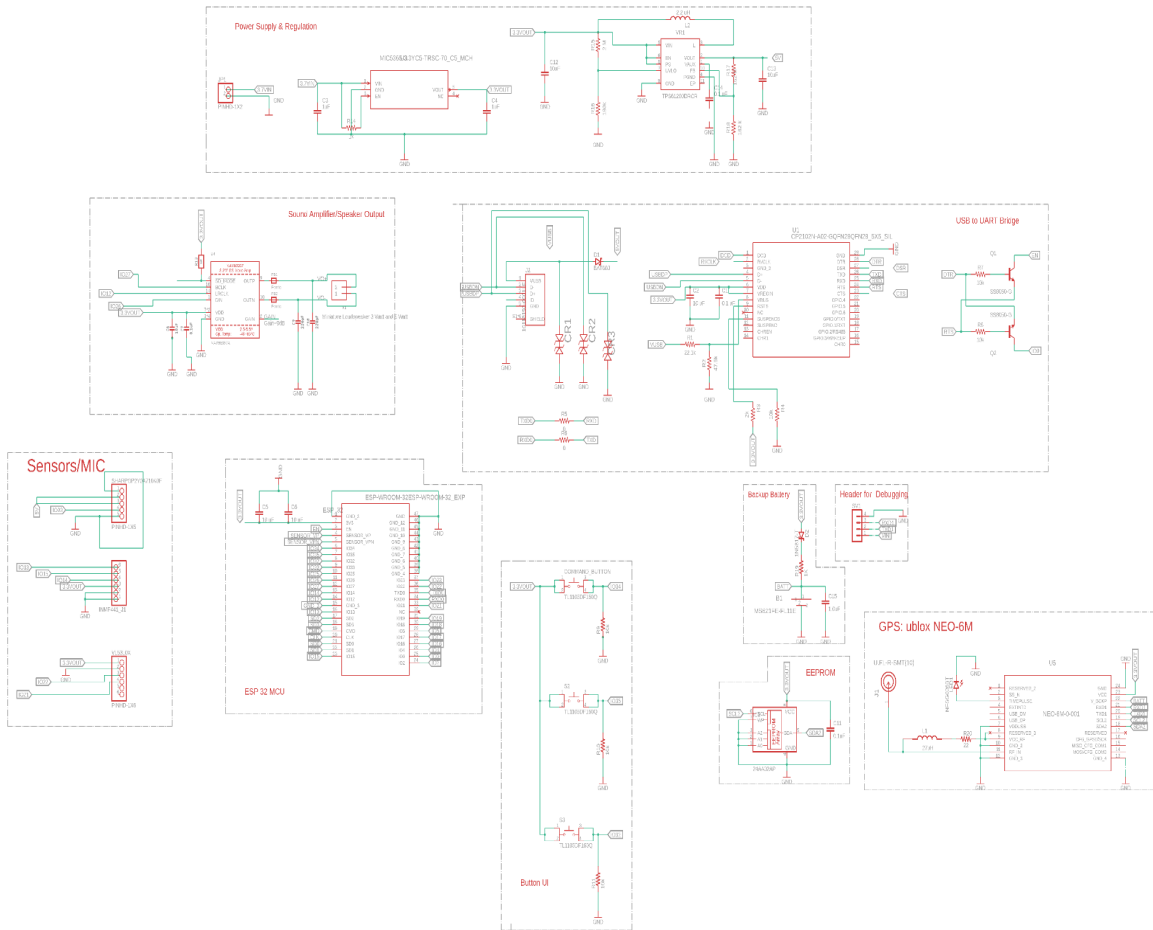


**Figure 6.12.** ESP 32 Wroom MCU Schematic.

The schematic for the ESP32-WROOM MCU in our Smart NAV stick design serves as the central nervous system, integrating all critical power management, communication interfaces, and peripheral connections. It outlines how the MCU interacts with components such as the CP2102N USB-to-UART bridge for programming and debugging, as well as other modules like the INMP441 microphone and MAX98357 amplifier for audio processing. This well-organized layout ensures stable power distribution and reliable signal integrity, forming the backbone of our smart navigation system.

## 6.10 Overall Schematic



+

**Figure 6.13.** Overall Project Schematic.

This is the overall schematic of the SmartNav Project, a modern enhancement to the traditional white mobility cane for the visually impaired. The design integrates two specialized housings to accommodate advanced electronic components while preserving the familiar structure of a standard cane. The top housing, located just below the handle, contains the main PCB and features a front-facing detection sensor and speaker for audio feedback. On the side of this housing are three tactile buttons and a small microphone port for user input and voice interaction. A second, much smaller housing is positioned near the tip of the cane, holding an additional sensor to detect low-lying obstacles. Together, these elements form a cohesive smart navigation system that improves spatial awareness and enhances the user's mobility experience.

## Chapter 7: Software Design

This chapter delves into the software architecture that forms the backbone of our SmartNav Stick project. It outlines the design and implementation of the system's software, detailing how different modules ranging from sensor data acquisition and processing to navigation, obstacle detection, and user feedback interact to deliver seamless real-time performance. Through comprehensive state diagrams, flowcharts, class diagrams, and data structure analyses, we demonstrate the logical flow and intricate communication pathways within the system. The chapter not only explains the high-level design decisions but also provides insight into the practical coding and debugging strategies employed, ensuring that our software is both robust and adaptable to evolving project requirements.

## 7.1 Functionalities Use Case Diagram

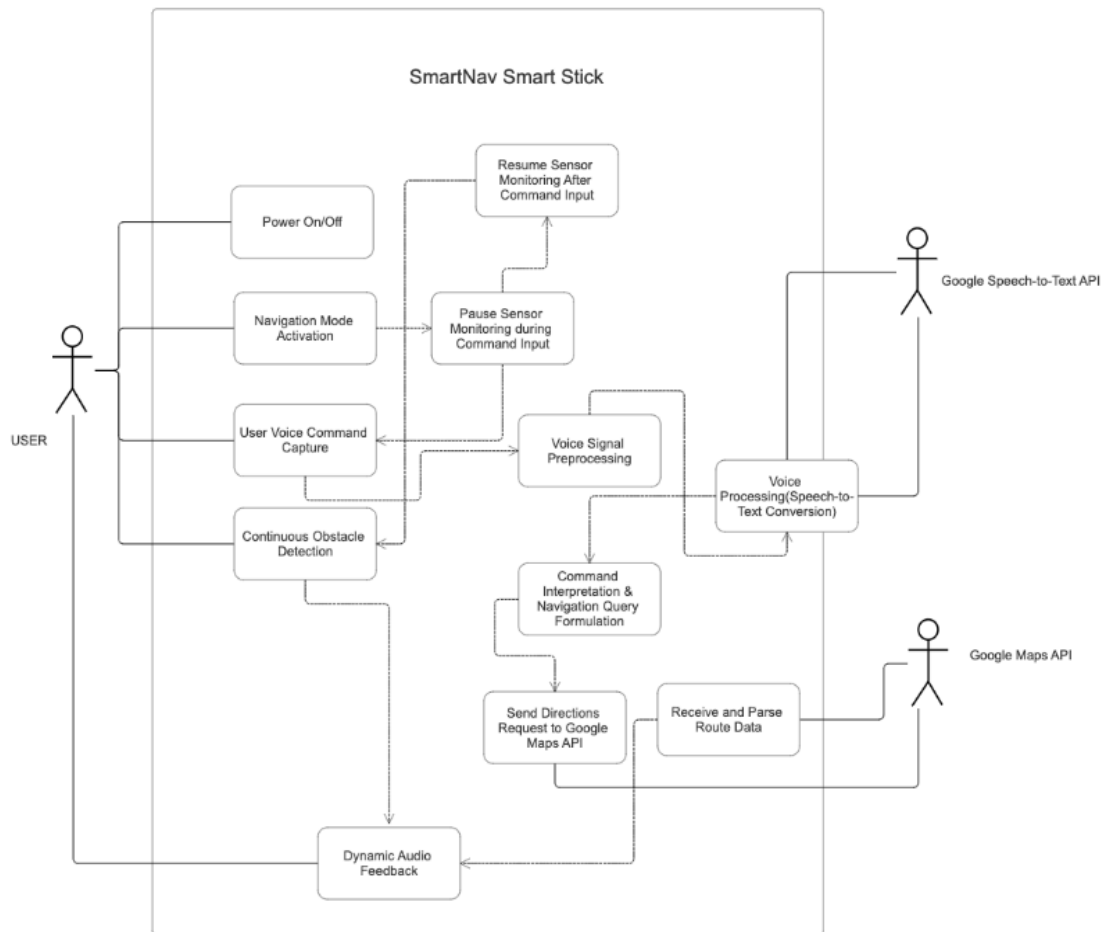


**Figure 7.1.** Use case diagram of SmartNav.

Our use case diagram for the SmartNav Stick offers a comprehensive, high-level view of how various components and functionalities interact to deliver a seamless user experience. At the core of the diagram is the SmartNav User, the primary actor who initiates and interacts with the system through physical inputs such as pressing the navigation button and receiving audio feedback. Surrounding this central role, the diagram delineates critical use cases that form the backbone of the device's operation, including power management (covering functions like power on/off and battery monitoring), navigation mode activation, and voice command capture. In addition, it illustrates the voice processing workflow that leverages external services like the Google Speech-to-Text API, enabling the system to accurately convert the user's spoken commands into text. The system then formulates a navigation query, sending this information to the Google Maps API, which returns detailed route instructions that the SmartNav Stick uses to provide dynamic audio guidance. This integration of internal functionalities with external API services is crucial for delivering real-time, context-aware navigation that adapts to both user commands and environmental data.

Simultaneously, the diagram captures the intricate interplay between concurrent system processes, such as the continuous obstacle detection managed by onboard sensors and the temporary suspension of these sensor readings during voice command input. The use case diagram clearly indicates that when navigation mode is activated, the system enters a state where sensor monitoring is paused to ensure that ambient noise and sensor data do not interfere with the accurate capture of the user's voice command. Once the command has been processed and navigation is underway, sensor monitoring resumes, allowing the device to provide immediate audio alerts if any obstacles are detected along the route. This dual-layered approach ensures that safety-critical functions like obstacle detection are maintained without compromising the clarity and accuracy of navigation commands. By mapping out these interactions and dependencies, the use case diagram not only serves as a blueprint for system functionality but also highlights the robust, responsive design that underpins the SmartNav Stick's operational effectiveness in real-world scenarios.

## 7.1.2 Software Architecture Class Diagram

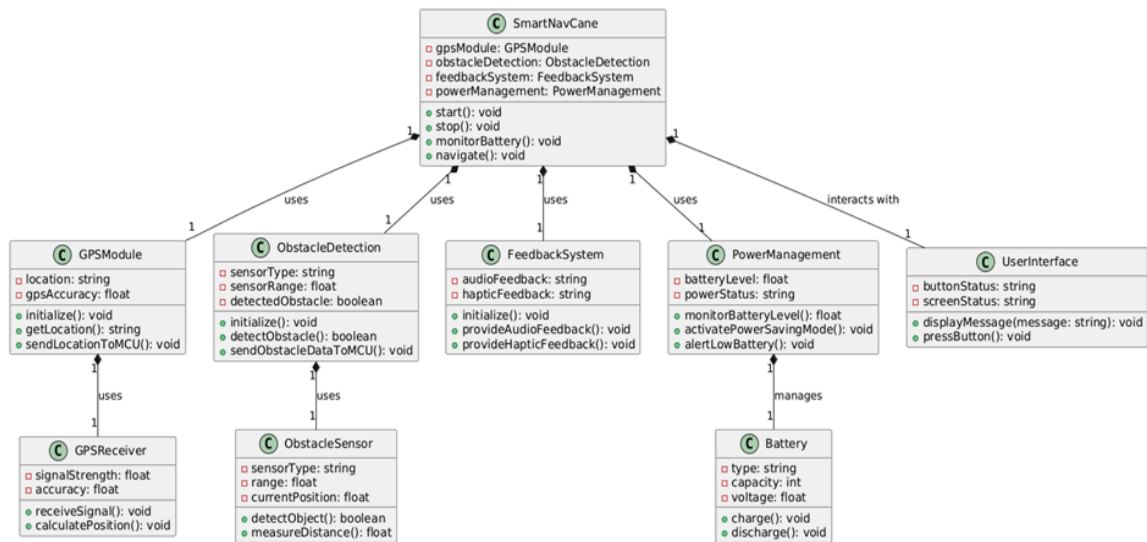


**Figure 7.2.** Software Architecture Class Diagram.

The class diagram serves as the foundational blueprint for the software architecture of the SmartNav Cane. Organized hierarchically with object classes at the top, attributes in the middle, and operations at the bottom, it provides a comprehensive visual representation of the system's structure and interactions. This diagram clearly delineates the primary components and illustrates how they interconnect, facilitating modular development and independent testing of each subsystem. By mapping out the flow of data and the relationships between classes, it demonstrates how the device achieves complex functions such as real-time navigation, precise location tracking, and robust object detection and avoidance. In essence, the diagram not only guides the software design process but also ensures that every component is aligned with the overall operational objectives of the SmartNav Cane.

Within the obstacle detection module, the diagram reveals the use of diverse data types and logic operations that are critical to the device's functionality. Here, string variables are employed to store labels or identifiers for detected objects, while float variables capture quantitative measurements like distance and size. Boolean operations play a central role by evaluating key conditions such as whether an object is within a critical proximity threshold or if its dimensions surpass defined limits to determine the appropriate system response. The outcomes of these logical evaluations are used to trigger dynamic actions, such as auditory warnings or haptic feedback, which guide the user safely around obstacles. This meticulous design underscores the importance of integrating

rigorous data handling and condition-based decision-making to enhance both the safety and usability of the SmartNav Cane.

## **7.2 Navigation / Object Detection System Architecture and State Flow**

The integrated system architecture that underpins the navigation and object detection functionalities of the SmartNav Stick, both the logical design and the dynamic state flow that govern its operation were outlined. They describe how user inputs trigger a sequence of events, from initiating voice command capture and processing via speech-to-text conversion to interfacing with the Google Maps API for real-time navigation instructions, all this occurs while concurrently managing sensor data for obstacle detection. The state flow diagrams presented here illustrate the discrete transitions between states, such as pausing sensor monitoring during navigation input, resuming obstacle tracking during active guidance, and handling interrupt-driven alerts when an obstacle is detected. This dual-layered approach ensures that critical safety mechanisms, like dynamic audio alerts for imminent hazards, seamlessly coexist with navigation commands, thereby providing a robust and responsive user experience.

## 7.2.1 Navigation System

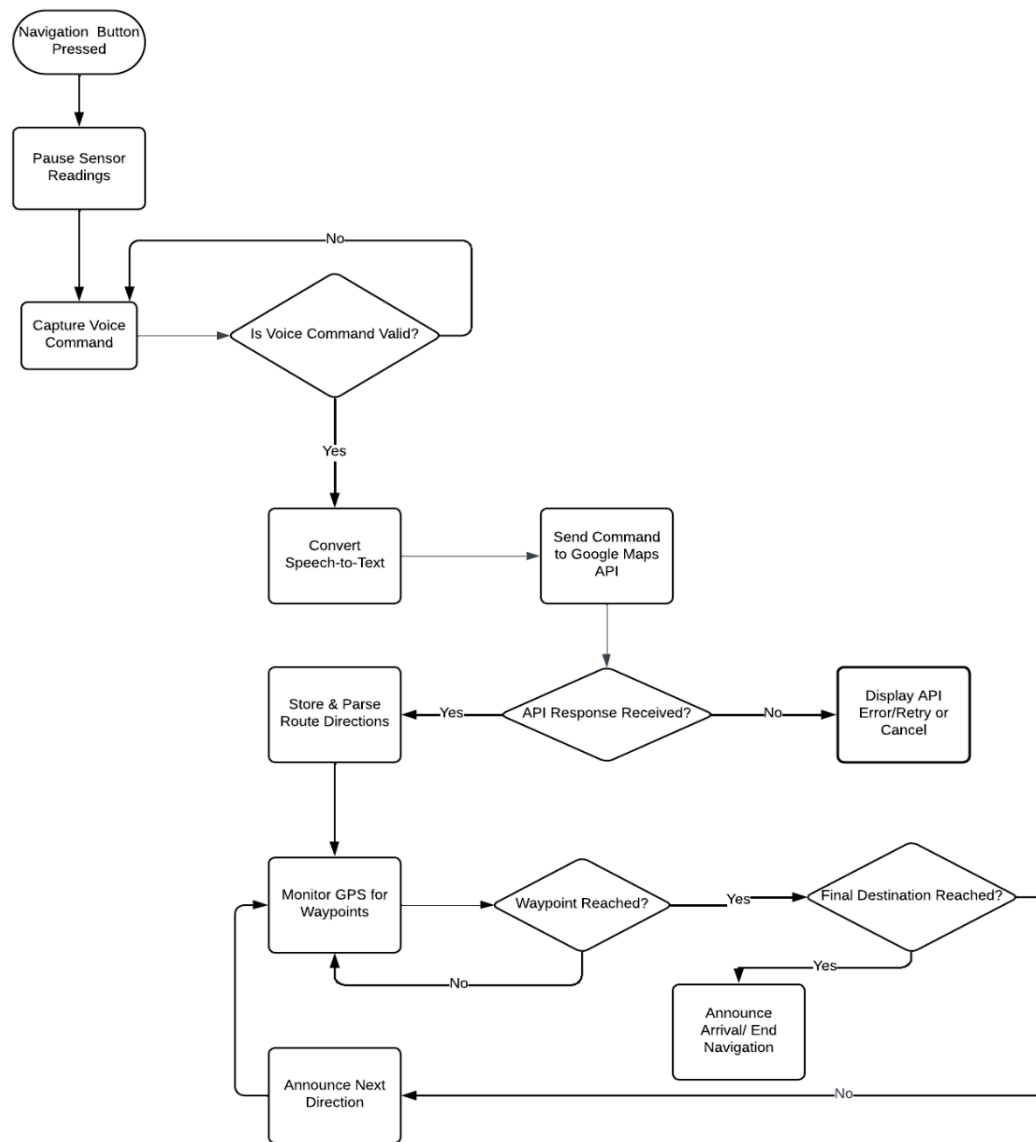


**Figure 7.3.** Navigation API Feedback System.

The Navigation Subsystem in our SmartNav project is designed to deliver a seamless and robust user feedback by integrating voice command input with the powerful capabilities of the Google Maps API. This subsystem initiates with the user pressing the navigation button, which promptly pauses ongoing sensor readings to prioritize capturing a clear voice command. The captured input is then rigorously evaluated for validity, ensuring that only accurate commands proceed further into the system. Should the initial voice command fail to meet the criteria, the system offers a retry mechanism, allowing for additional attempts before safely aborting the process and returning to an idle state. Once validated,

the command undergoes a sophisticated speech-to-text conversion process, translating the spoken input into text that is used to query the Google Maps API. Upon receiving a response, the subsystem parses the detailed routing instructions and transitions into an active guidance mode, where it continuously monitors GPS signals to track the user's progress along the determined route. This dynamic monitoring enables real-time detection of waypoint achievements, triggering timely updates and directional announcements to assist the user. The flowchart provided below encapsulates this entire process, detailing every critical decision point and error handling pathway from initial command capture, through API communication, to the final confirmation of destination arrival. This comprehensive approach not only highlights the technical rigor behind our navigation strategy but also demonstrates our commitment to integrating cutting-edge technologies to deliver an intuitive, efficient, and resilient navigational experience.

## 7.2.2 Object Detection System

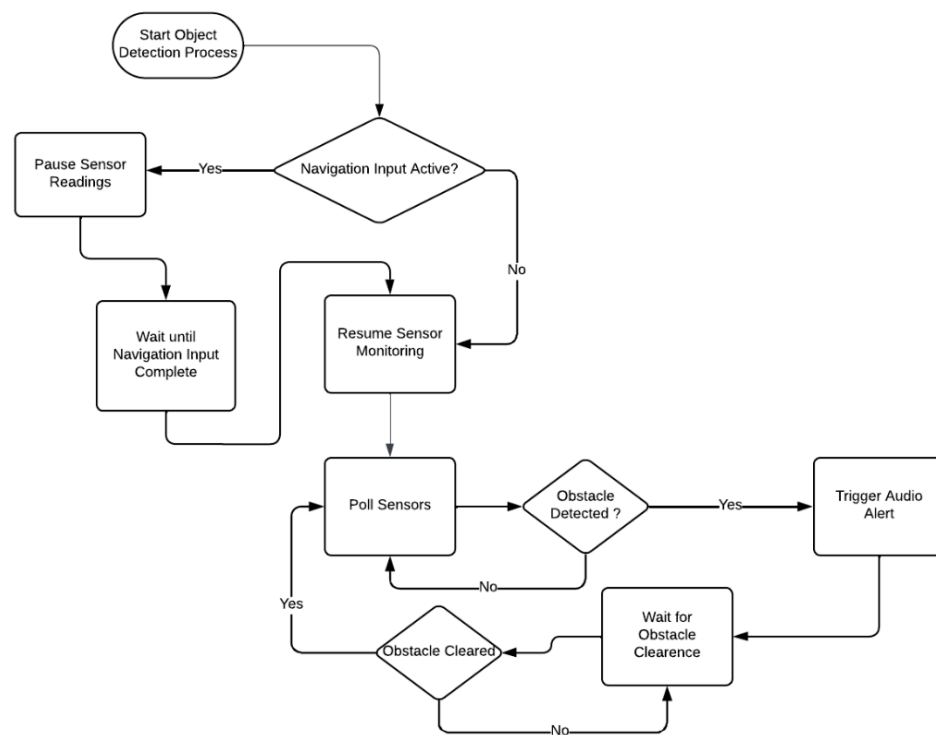


**Figure 7.4.** Object Detection Feedback System.

This flow chart outlines the object detection logic integrated within our SmartNav Stick, emphasizing its dual functionality and dynamic state transitions. The

system continuously polls the environment for obstacles using sensor inputs, but intelligently pauses this monitoring when the user is actively entering navigation details, ensuring that the voice capture and command processing occur without interference. Once the navigation input is complete and the journey commences, the sensor monitoring resumes immediately. When an obstacle is initially detected, the system enters a debounce phase to validate the reading, filtering out transient noise before confirming an obstacle. Upon confirmation, an audio alert is triggered, and the system enters an alert mode that interrupts other functions until the obstacle is cleared or acknowledged. This robust design ensures that obstacle detection operates seamlessly alongside navigation, prioritizing user safety while maintaining effective guidance throughout the journey.

## 7.4 GPS Module Initialization, Data Processing and Feedback

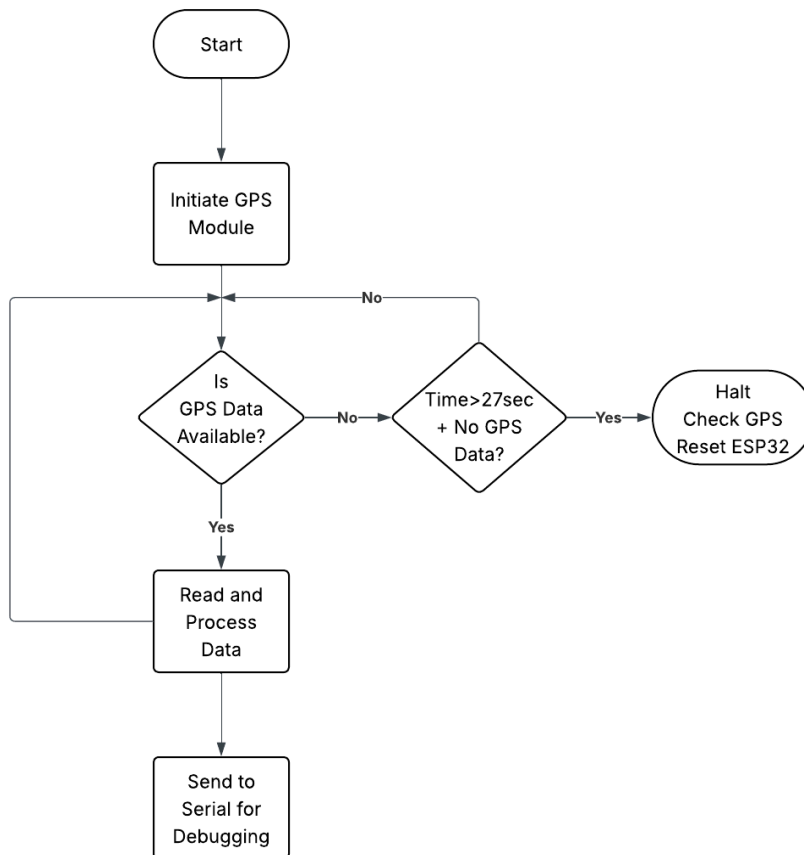


**Figure 7.5.** GPS data processing from GPS module.

Here the GPS module is initialized and checked for new data. If data is found the data is read and processed which includes parsing of the GPS data. The data is also sent to serial1 for interfacing with the CPU during debugging. If no new data is found after 27 seconds, the time specified for a time first fix, the program is halted and will restart after a reset. The user could be indoors and doesn't have a clear view of the sky or there could be an issue with the GPS module itself and the program doesn't need to continually run during this time.

## **7.5 Data Transfer**

In the SmartNav Stick, data transfer between sensors, processing units, and output devices has to be robust and efficient, as this is key to the system's overall performance and reliability. Through this chapter, we explore the multiple ways that the robot is interconnected (I2S, I2C, ADC and UART) allowing digital and analog information to be acquired, processed, and relayed to provide audio feedback, obstacle detection, and navigation in real-time. For each protocol, we discuss the hardware interfaces, software libraries, and design considerations that make this communication possible in the system.

### **7.5.1 I2S Data Transfer**

The SmartNav Stick uses the I2S (Inter-IC sound) interface as the main support of high-fidelity audio processing and transfer. Using the I2S (Inter-IC Sound) protocol, the core of the INMP441 MEMS microphone receives audio from the environment, converting this analog information into a high-resolution digital stream. This data is passed through specialized I2S lines consisting of data, word select, and clock signals directly to the ESP32 microcontroller, which on the other side processes the signals live with the help of the I2S utilized within the Arduino IDE environment. h library. This allows sound to be captured instantly, without delays or distortion, thanks to its pronounced configuration: sampling rate, bit depth, and data format. The resulting digital audio data passes via the same set of I2S connections to the MAX98357 Class-D amplifier, which reconstructs the original audio waveform into an analog signal suitable for driving a mini-speaker. That entire digital-to-analog-to-digital conversion chain is critical for the delivery of clear and immediate audio cues, like obstacle warnings and navigation prompts, to the user. Attention to detail in the configuration of the I2S bus in the areas of clock synchronization, and error-handling routines, for instance ensures low latency and high signal integrity, as is needed to support the system's real-time operational requirements (Intersil, "INMP441 MEMS Microphone Datasheet"; Maxim Integrated, "MAX98357 I2S Class-D Amplifier Datasheet").

### **7.5.2 I2C Communication**

The I2C protocol is implemented within the SmartNav Stick to facilitate communication with peripheral sensors that require a simple, efficient two-wire interface. The VL53L0X Time-of-Flight sensor, which is pivotal for precise distance measurement and obstacle detection, utilizes the I2C bus to transmit its data. With only two wires SDA for data and SCL for the clock this protocol minimizes wiring complexity while allowing the ESP32 microcontroller to reliably query the sensor for distance readings at regular intervals. Within the Arduino IDE, the Wire.h library is employed to initialize the I2C bus, manage data transfer, and handle potential errors, ensuring that sensor measurements are both timely and accurate. The integration of I2C not only simplifies the connection between the microcontroller and the sensor but also facilitates a modular design where additional I2C-compatible devices could be incorporated with minimal disruption. Furthermore, the inherent addressing capabilities of I2C allow for multiple devices to share the same bus, making it an ideal choice for compact, multi-sensor configurations. By ensuring robust signal integrity with properly sized pull-up resistors and appropriate bus speed settings, the SmartNav Stick can continuously monitor its environment and adjust navigation cues based on real-time sensor data (STMicroelectronics, "VL53L0X Time-of-Flight Sensor Datasheet").

### **7.5.3 ADC Data Acquisition**

One of the most important building blocks for data acquisition is Analog-to-Digital Conversion (ADC); this converts an analog signal, output from a sensor such as the SHARPGP2Y0A710K0F infrared distance sensor, into a stream of digital values for processing by the ESP32. This type of sensor generates an analog voltage, which takes place in a continuous way related to the distance of an obstacle, and it is read from the ESP32 Arduino IDE using the analogRead() function on its ADC channels. These digital values are then meticulously calibrated and connected to specific distance values, meaning that even slight changes in the output of the sensor are recognized and interpreted accurately. Since ADC is sensitive to noise and signal errors, designers have to put extra effort in their design (e.g., precision voltage references, filters, and signal conditioning circuits) to preserve the quality. Such a rigorous implementation of ADC allows the SmartNav Stick to make real-time decisions like enabling audio alerts when an object falls within a critical distance or range. The ADC subsystem serves an essential function in ensuring the overall safety and responsiveness of the navigation system by integrating analog sensor outputs into the digital processing workflow.

### **7.5.4 UART Communication**

UART (Universal Asynchronous Receiver/Transmitter) communication is not the main channel for sensor or audio data but works as a support mechanism within the SmartNav Stick for diagnostics, firmware updates, and external communications. This USB-UART bridge, CP2102N is used to provide a serial interface between the ESP32 microcontroller and the external USB part (computer or peripheral module). To facilitate transfers of debug information, system logs, and status messages, the UART (Universal Asynchronous Receiver-Transmitter) interface is programmed using the Arduino IDE's Serial library, which allows for asynchronous operations at defined baud rates. This serial connection is particularly useful in the development and testing periods when real-time observations of device operation and things like issues can be addressed live. For that reason, and its flexibility, a UART interface can also be used as a backup channel for communication in case any other interfaces get problems. While UART itself may not directly be a functional part of navigating an environment avoiding obstacles, its implications for the core design to provide reliable rearrangements to the system and bug-fixing, it is a crucial part of the overall design plan.

## **7.6 Task Management Hierarchy and Interrupt-Driven Responsiveness**

### **7.6.1 Task Prioritization**

In a system as dynamic and safety-critical as the SmartNav Stick, proper task prioritization is essential to ensure that the most time-sensitive and mission-critical operations are executed with minimal latency. The architecture of the firmware, whether structured around a real-time operating system (like FreeRTOS) or a tightly controlled loop() in the Arduino IDE, is designed with layered task importance in mind. At the highest priority level are tasks like obstacle detection, which poll sensors such as the VL53L0X ToF and SHARP GP2Y0A710K0F at high frequency to monitor the user's environment in real time. These tasks are non-negotiable; they must run continuously or at predictable intervals (e.g., every 50–100 ms) to detect imminent hazards and trigger safety alerts. Below this level are navigation-related tasks, which include GPS coordinate tracking, direction updates, and audio feedback. These are important but can afford short delays, as they generally operate on slower-changing data (e.g., a new direction every few meters). Background tasks, such as battery level monitoring and system diagnostics, are given the lowest priority and scheduled at longer intervals or triggered only when needed. By structuring the firmware with clearly defined task layers using time-based execution in loop() or priority levels in FreeRTOS (xTaskCreate with assigned weights) we ensure that the system remains responsive to critical events while still delivering complex multi-functionality such as speech-based navigation and sensor fusion. The task hierarchy is further enforced by flag-based control flow and conditional branching

that prevents lower-priority tasks from executing when higher-priority conditions (like an obstacle warning) are active.

## **7.6.2 Interrupt Service Routines**

To complement task scheduling, the SmartNav Stick relies on hardware interrupts to instantly respond to user interactions and time-critical events without waiting for the current execution thread to complete. The most prominent example is the navigation button press, which is managed via an external interrupt tied to a digital GPIO pin on the ESP32. When the user presses the button, the system immediately suspends ongoing lower-priority operations (such as routine sensor polling or audio playback), triggering an Interrupt Service Routine (ISR) that flags the microcontroller to enter a new state typically, voice command capture mode. This mechanism ensures that user input is processed with sub-millisecond latency, a requirement for smooth interaction and responsive behavior. Unlike regular polling methods, ISRs eliminate the need to constantly check for pin states, reducing unnecessary CPU load and power consumption. In our implementation, the ISR is minimal its role is to set a flag or enqueue an event, which is then handled in the main loop or a dedicated task. This approach avoids the common pitfalls of complex ISR logic, such as race conditions or stack overflows. Furthermore, the ESP32's multi-core architecture allows certain ISRs or high-frequency tasks to be pinned to a specific core using `xTaskCreatePinnedToCore()`, improving concurrency and ensuring that time-critical operations (like audio alerts or command capture) do not interfere with background processing. By combining responsive interrupt-driven logic with a well-structured task priority system, the SmartNav Stick maintains both safety and performance, ensuring that the user always receives timely feedback even in the presence of multiple concurrent system demands.

## **7.7 API Integration and Protocols**

API calls are a crucial component of the SmartNav Stick, enabling it to interface seamlessly with external services that greatly enhance its functionality. In our design, we leverage APIs primarily for two key purposes: converting voice commands into actionable text via the Google Speech-to-Text API and retrieving real-time routing information through the Google Maps API. These API calls allow the device to process user inputs and deliver dynamic navigation instructions, making it capable of providing a responsive and context-aware user experience. The integration of these external services is essential not only for functionality but also for maintaining the overall performance and reliability of the system.

At the technical level, API calls in the SmartNav Stick are implemented over secure HTTP(S) connections using RESTful endpoints. The ESP32 microcontroller is responsible for initiating these calls asynchronously, ensuring

that critical processes such as sensor polling for obstacle detection continue uninterrupted while awaiting responses from the cloud services. This is achieved using code libraries like HTTPClient in the Arduino IDE or PlatformIO environment, which facilitate the creation, dispatch, and management of HTTP requests and responses. Each API call is meticulously configured with the appropriate endpoint URL, request headers, and payload parameters, all of which are documented in the respective API documentation. For example, when processing a voice command, the raw audio is first captured and pre-processed before being sent to the Google Speech-to-Text API. The returned transcription is then used to formulate a navigation query that is dispatched to the Google Maps API, which, in turn, returns a detailed set of directions and waypoints. Error handling strategies such as retries, timeout management, and fallback mechanisms are implemented to ensure robustness in scenarios where network connectivity or API response times are less than ideal.

Configuration of these APIs is critical; it involves obtaining secure API keys, ensuring they are stored safely within the firmware, and setting up proper authentication protocols. The device must also parse JSON responses accurately to extract useful data, which is then integrated into the real-time decision-making algorithms that drive the SmartNav Stick's navigation and obstacle avoidance features. Furthermore, legal considerations are paramount when dealing with third-party services. Our implementation strictly adheres to the terms of service and data usage policies set forth by Google, ensuring that all user data is transmitted securely and that the device complies with privacy regulations. By maintaining compliance and robust security measures, we not only ensure the integrity of the system but also protect user data from unauthorized access or misuse. This comprehensive API integration approach underscores the SmartNav Stick's commitment to leveraging cutting-edge cloud-based services while maintaining high standards of software design and legal compliance.

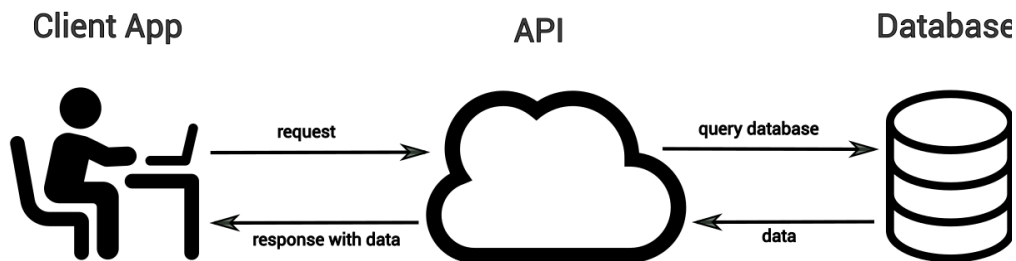


**Figure 7.6.** Conveying API Integration.

## Chapter 8: Prototype Fabrication

For the SmartNav device, we've developed a set of clustered, specialized PCBs designed around three fundamental guidelines critical to the overall project's success: Using a bare microcontroller chip with external components, using multiple custom-made peripherals, and a well structured power management system. At the heart of our design is the ESP32 microcontroller PCB, serving as the main logic hub and integrating essential power supply modules, including voltage regulators and boost converters, to ensure stable and efficient power delivery. Accompanying this main PCB, we designed an auxiliary rechargeable battery PCB, leveraging breakout modules for convenient and safe energy storage and management. Additionally, recognizing the modularity requirements of the project, we included pin headers on the main board to interface seamlessly with custom-designed peripheral boards, such as our Time-of-Flight (ToF) sensor breakout, microphone and amplifier circuits, GPS module, and other sensor peripherals. This structured yet modular design allowed us to achieve optimal performance, ease of debugging, minimal interference between sensitive signals, and streamlined assembly all while adhering to our project's design criteria of modularity, reliability, and efficient power management.

### 8.1 Main PCB Board Design

This is our main PCB, the central core of our SmartNav device, thoughtfully designed to integrate multiple critical functions essential to device operation using the bare ESP 32 Wroom 32 D chip. It hosts multiple specialized components, including audio amplification circuitry to ensure clear and effective audio feedback, a UART-to-USB bridge that facilitates efficient debugging and communication, and a comprehensive power management section featuring one voltage regulator and converters to reliably power the entire system. To accommodate expandability and modular integration, we incorporated multiple pin headers, providing simple connection points for external peripherals that we will build and custom breakout modules. By consolidating these diverse functions into a single, cohesive PCB, we streamlined the overall design, improved signal integrity, simplified debugging, and significantly enhanced reliability for the SmartNav project. Given the integration of the ESP32-WROOM module, careful attention was placed on PCB manufacturing requirements such as controlled-impedance routing for antenna traces, adherence to recommended antenna keep-out areas to maximize wireless performance, proper thermal management to handle the module's heat dissipation, and rigorous power distribution strategies to ensure stability. These considerations were instrumental

in achieving optimal ESP32 performance and ensuring our PCB is compliant with best practices for reliability and electromagnetic compatibility.

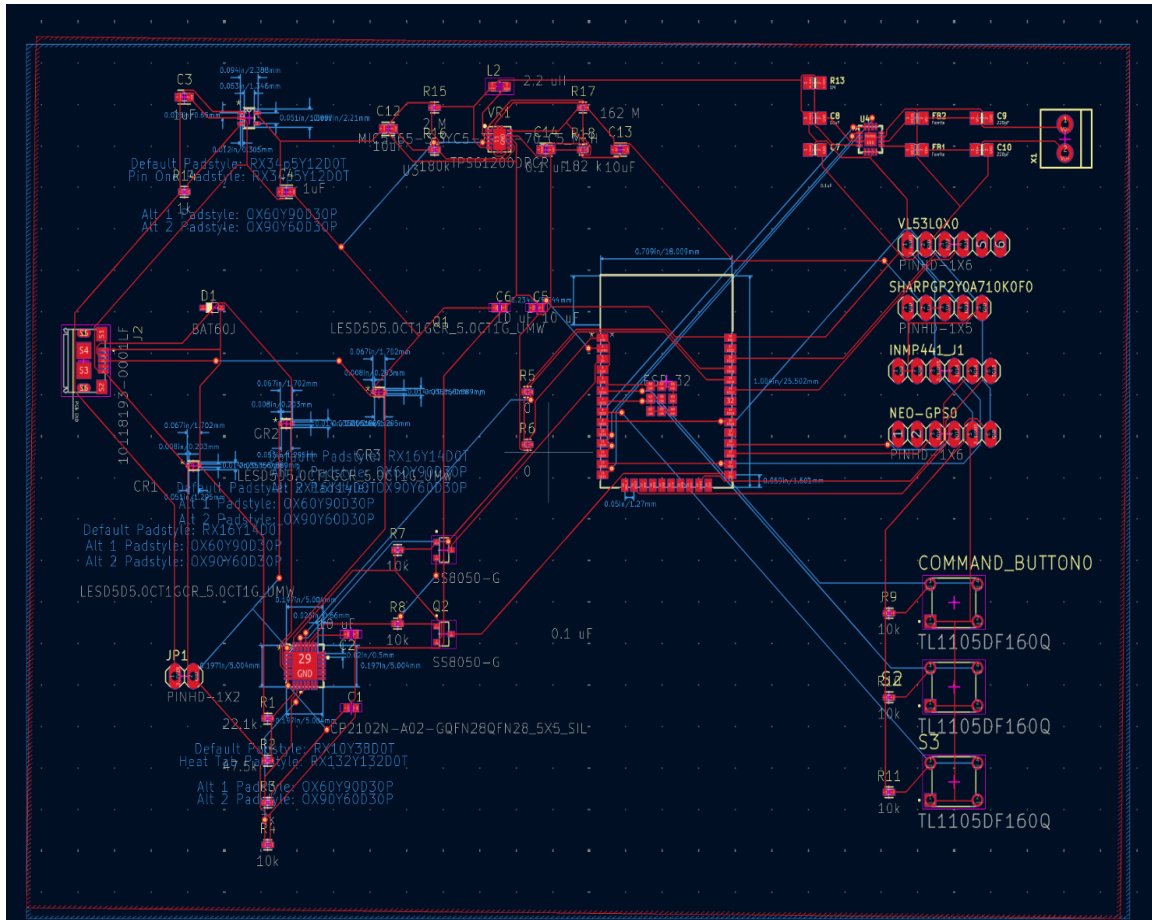


**Figure 8.1** Main PCB in KiCAD.

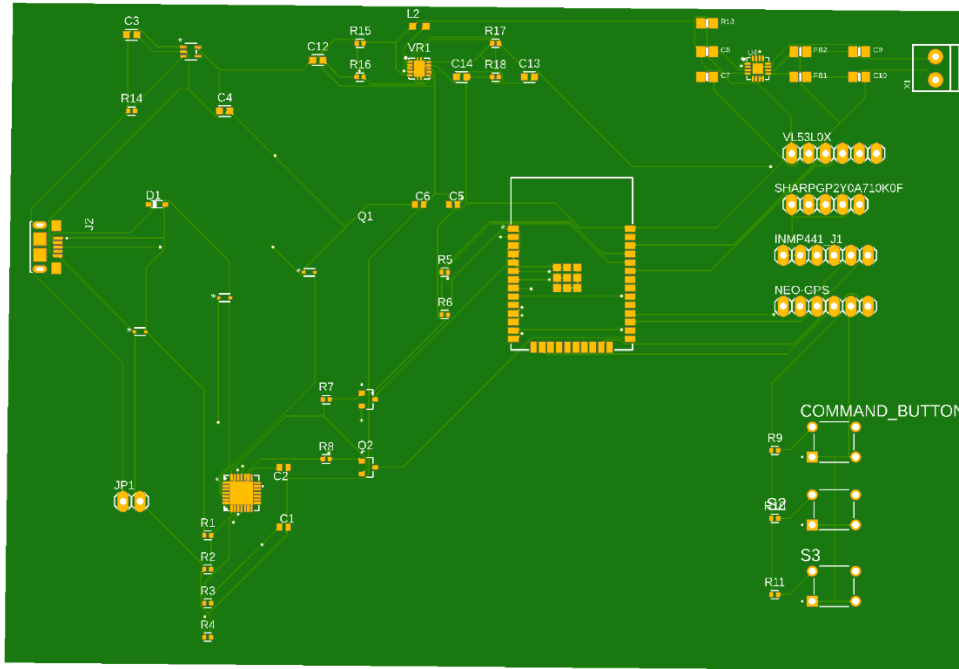


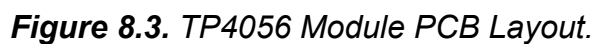
**Figure 8.2** Main PCB Manufacturing Preview.

## 8.2 Rechargeable TP4056 Module Peripheral

The Rechargeable TP4056 Peripheral Breakout Board helps consistently power our lithium ion batteries is a dedicated PCB module that manages efficient and safe charging and discharging cycles for our project's lithium-ion battery. At the heart of this peripheral is the TP4056, a specialized linear charger IC explicitly designed for lithium-ion and lithium-polymer batteries. The TP4056 is engineered to provide precise charging control through constant current/constant voltage (CC/CV) charging methods, ensuring battery longevity and safe operation.

Complementing the TP4056, we've integrated the DW01A battery protection IC, which continuously monitors the battery voltage and current, offering essential protection mechanisms against common hazards such as overcharging, over discharging, and excessive current draw. This intelligent protection IC immediately triggers protective measures in hazardous conditions, significantly increasing battery life and safety. Additionally, our breakout incorporates the FS8205A dual MOSFET in a SOT-23-6 package, serving as a high-efficiency switch controlled by the DW01A. This dual MOSFET manages the charging and discharging paths, effectively isolating the battery when protection thresholds are

We chose to create this module as a separate PCB to isolate sensitive power management circuitry from the main device logic, significantly reducing electrical noise and interference, which is crucial for precision navigation and sensor accuracy in the SmartNav system. Additionally, separating the battery management from the main PCB simplifies troubleshooting, servicing, and future upgrades, offering increased flexibility in power management solutions. The modular approach also enables us to adapt different battery capacities or charging methods without modifying the main PCB, thus streamlining our overall development and manufacturing process. Overall, the Rechargeable TP4056 Peripheral Breakout Board is a critical, well-engineered solution that ensures reliable, safe, and efficient battery operation, aligning perfectly with the modular and performance-oriented goals of our SmartNav project.



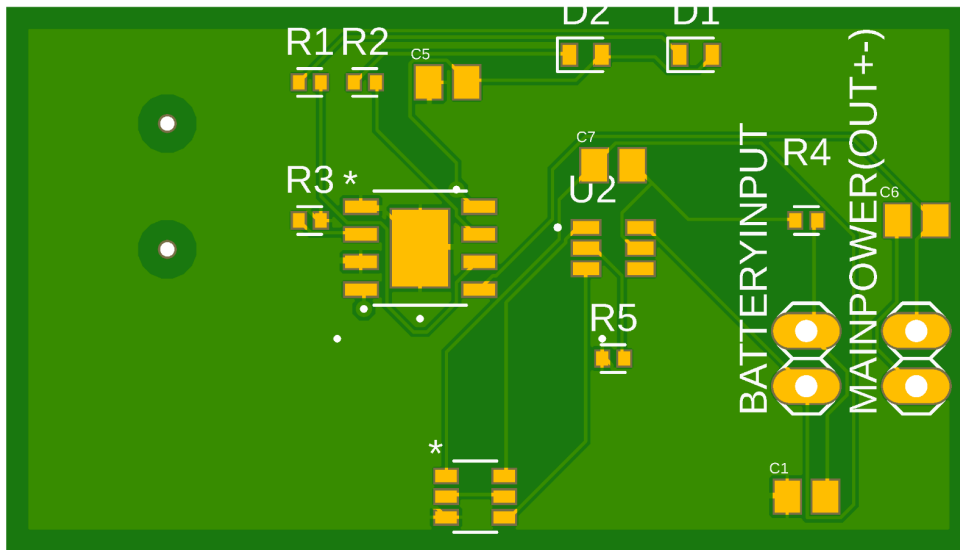


**Figure 8.4.** *TP4056 Module Manufacturing Preview.*

### 8.3 NEO-M9N GPS Module BreakOut Board

Our decision to house the Global Positioning System (GPS) module on a dedicated printed circuit board (PCB) was driven by the significant advantage this approach offers in terms of antenna placement. By employing a flexible ribbon cable to establish the connection between the standalone GPS module and the primary system PCB, we gain considerable freedom in optimizing the antenna's physical location. This is crucial for ensuring robust and reliable GPS signal reception, as the antenna can be positioned away from potential sources of electromagnetic interference or obstructions that might be present on the main board.

To ensure the proper integration and functionality of the GPS module, we meticulously reviewed the component datasheets, paying close attention to the recommended hookup diagrams and schematic representations. Furthermore, we examined existing and proven GPS module designs, drawing inspiration and best practices from reputable sources such as SparkFun's offerings and the widely adopted GY-GPS6MV2 breakout board. This comparative analysis allowed us to leverage established knowledge and avoid potential pitfalls in our design.

The process of translating our conceptual design into tangible hardware involved the use of industry-standard software tools. Specifically, we utilized Autodesk's Eagle for the creation of detailed electronic schematics, outlining the interconnections between the various components of the GPS module PCB.

Autodesk's Fusion 360 was employed to generate the initial physical layout designs for the PCB, considering factors such as component placement, trace routing, and overall board dimensions.

In formulating our GPS module PCB design, we deliberately adopted a simplistic approach. This approach is predicated on the understanding that the main system PCB has a number of essential support functions. These include providing the necessary regulated voltage to power the GPS module, managing the recharging of the onboard backup battery, and handling the operation of the onboard Electrically Erasable Programmable Read-Only Memory (EEPROM). By centralizing these functions on the main PCB, we were able to simplify the design of the GPS module PCB, focusing its resources solely on the core GPS receiver circuitry and the antenna interface.

The inclusion of a backup battery and an EEPROM on the GPS module PCB is a strategic decision aimed at enhancing the speed and reliability of GPS acquisition. The backup battery ensures that critical satellite ephemeris data and the last known position are retained even when the main power supply is disconnected. This retained information allows the GPS receiver to perform "hot starts" or "warm starts" upon subsequent power-up, significantly reducing the time required to establish a satellite fix compared to a "cold start," where the receiver must acquire satellite signals and download orbital data from scratch. This capability is particularly important for applications requiring quick and seamless location awareness.

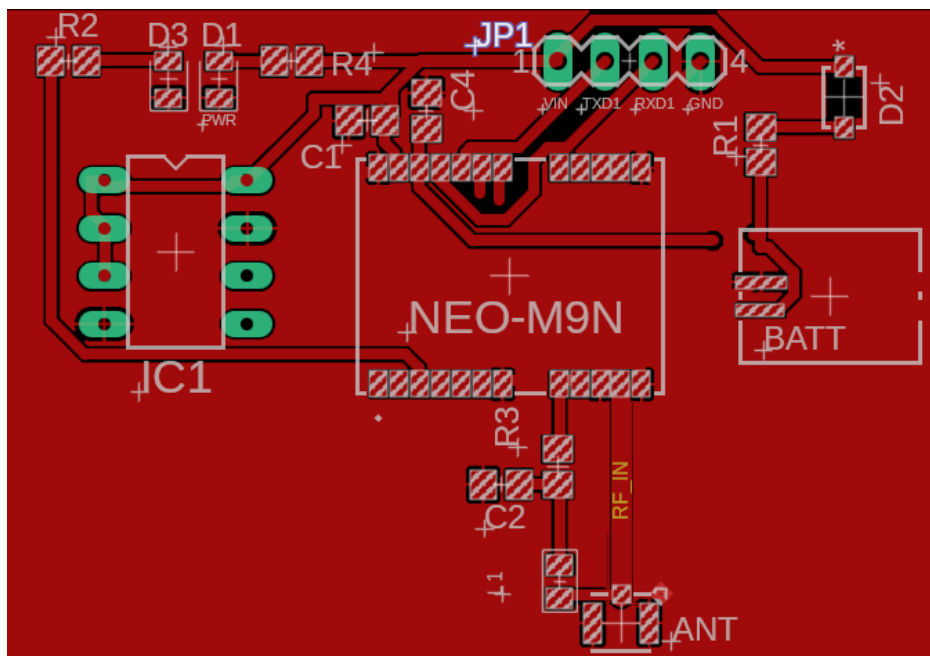


**Figure 8.5.** GPS module PCB.

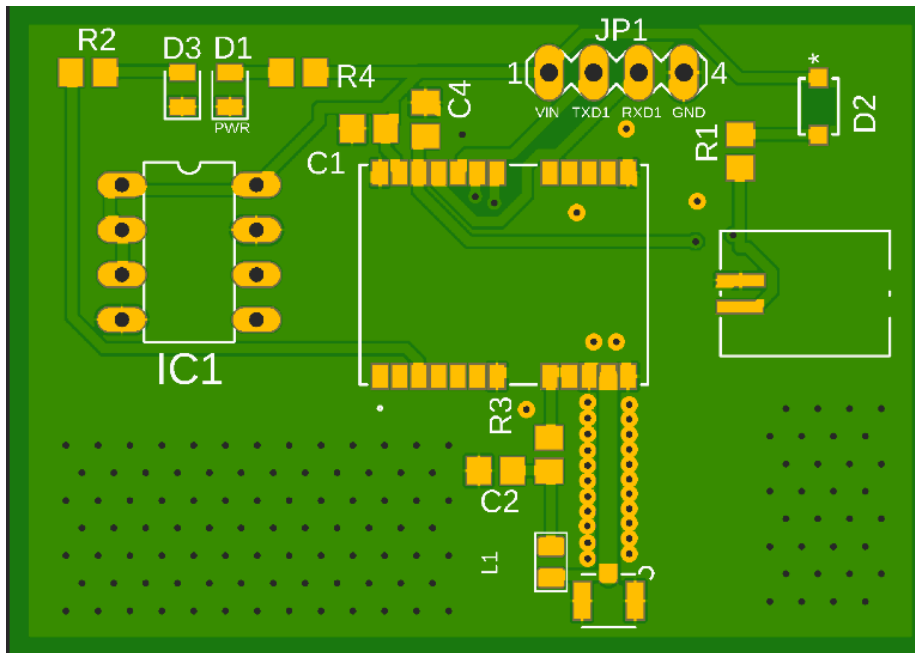


**Figure 8.6.** GPS Module PCB (Top Side)

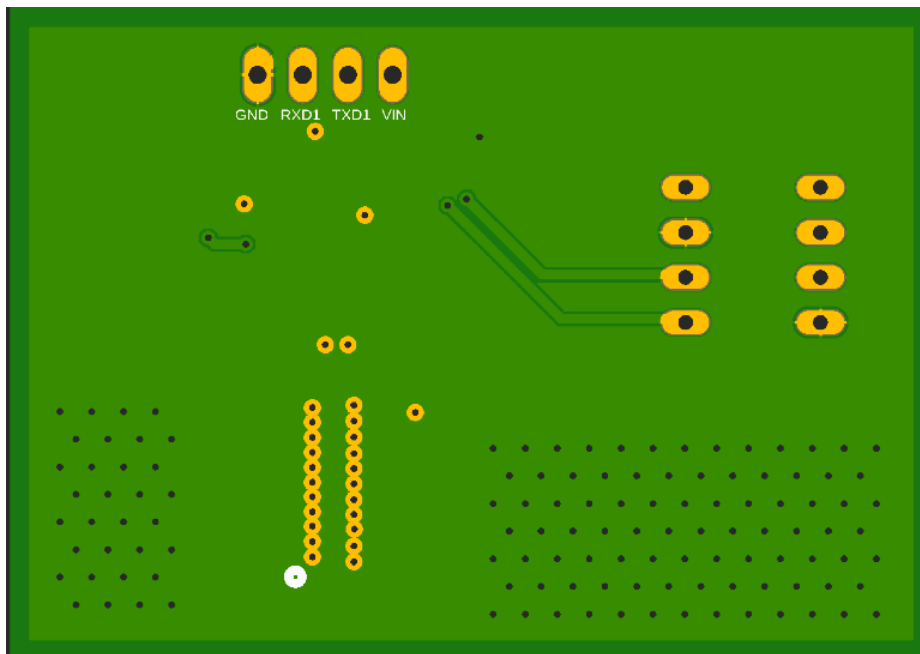


**Figure 8.7.** GPS Module PCB (Bottom Side)

## 8.4 TOF VL530X BreakOut Board

As part of the SmartNav mobility stick's peripheral suite, a custom VL53L0X Time-of-Flight (ToF) Breakout Board was designed and integrated to enable

precise distance sensing in real-time navigation scenarios. The VL53L0X, developed by STMicroelectronics, is a compact laser-ranging module capable of measuring absolute distances up to 2 meters with high accuracy, regardless of target reflectance. Unlike conventional IR sensors, this ToF technology offers consistent performance across varied surface types and ambient conditions, making it ideal for obstacle detection in dynamic environments. Our custom PCB breakout board, measuring  $3 \times 2.5 \times 0.5$  inches, was developed to support the bare chip with the advised external components recommended in ST's reference documentation. This includes decoupling capacitors, voltage regulation, and I2C pull-up resistors, all selected to ensure stable operation, signal integrity, and ease of integration with the main ESP32-based controller. The modular form factor of the board allows it to be mounted within the lower housing of the mobility stick for unobstructed downward-facing sensing, contributing to the overall spatial awareness system of the device.

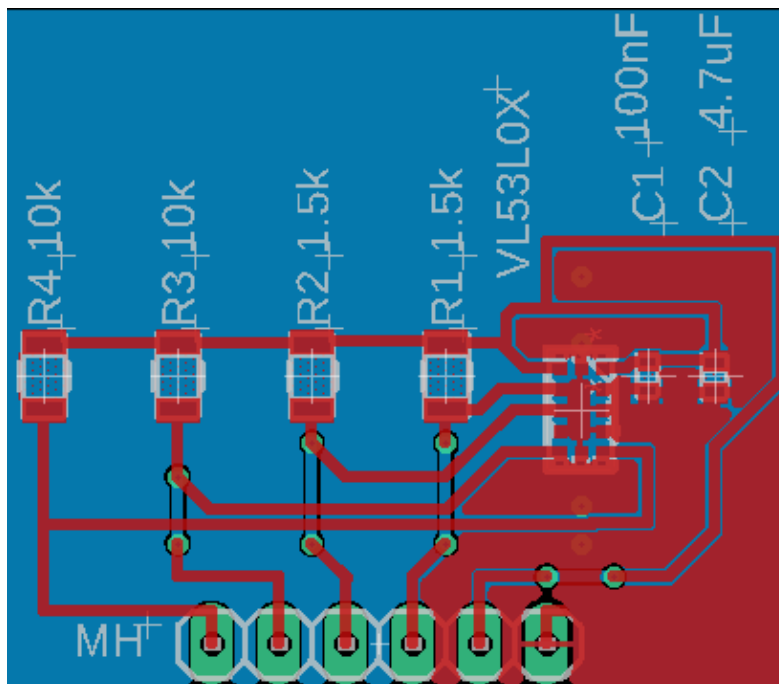


**Figure 8.8.** ToF Sensor PCB.

## Chapter 9: System Testing

### 9.1 Hardware Testing

Thorough testing of both hardware components and software is critical to ensuring proper functionality and reliability in the SmartNav Cane, it helps us understand how exactly our peripherals work and how to properly integrate them

into the smart Nav system. The sections below provide details of how each component has been tested.

### **9.1.1 ESP32-WROOM-32D**

To test the functionality of the ESP32 development board, we simply connected the board to a computer using a USB→USB-C cord to confirm it was receiving the power. The powering of the LED on the board verified that the device was receiving voltage. The voltages of the input/output pins on the board were tested using a digital multimeter. In addition, a little time was spent testing the functionality of certain features available through this board. We tested serial communication through Arduino IDE and wrote a simple script to print to the serial monitor. We also confirmed that the board had functioning Wi-fi by writing another short script and checking the detected networks in the serial monitors. These procedures were found in the ESP32 Technical Reference Manual.

### **9.1.2 MIC5365-3.3 Linear Voltage Regulator**

To test the MIC5365-3.3 linear voltage regulator, we constructed a breadboard following the “typical application” schematic in the datasheet. This schematic and datasheet provides all information needed to construct this board including the components and their values. After completing this board, we connected it to a power supply varying the voltage between 0 - 5.5 volts to confirm we were receiving the desired output voltage of 3.3 volts. This output voltage was measured using a digital multimeter.

### **9.1.2 TPS61200 Step-Up Voltage Regulator**

To evaluate the performance of the TPS61200 step-up voltage regulator, we constructed a functional prototype on a breadboard based on the “Typical Application Circuit” provided in the official Texas Instruments datasheet. This reference schematic included all necessary component values such as the input/output capacitors, inductor specifications, and feedback resistors ensuring that our test circuit would replicate the intended operating conditions of the regulator. Careful attention was given to matching the recommended component ratings, particularly for the inductor and output capacitor, as these directly impact voltage stability and efficiency.

Once the circuit was assembled, we powered it using a variable bench power supply. The input voltage was gradually adjusted across a range from 0V to 5.5V to observe the regulator’s ability to maintain a steady 5V output under varying supply conditions. The goal was to confirm that the TPS61200 would properly boost lower input voltages such as those from a depleted battery up to a regulated 5V output suitable for powering sensitive components like

microcontrollers and sensors. Throughout the test, a digital multimeter was used to monitor the output voltage with precision, and the readings consistently aligned with the expected 5V output once the input voltage crossed the minimum threshold required for regulation. This validation confirmed that the regulator was operating correctly and would be suitable for integration into the SmartNav system's power management design, particularly in situations where battery voltage drops below the required operating level of key components.

### **9.1.3 ToF VL53L0X Object Detection Sensor**

To test the ToF VL53L0X sensor, we constructed a breadboard following the “typical application” schematic in the datasheet. This schematic and datasheet provides all information needed to construct this board including the components and their values. After completing this board, we connected it to a power supply using 3.3 volts. We then connected this board to our ESP32 development board and installed the VL53L0X library to Arduino IDE. We wrote a short script and used the serial monitor to observe the distance measurements.

### **9.1.4 IR Sharp GP2Y0A710K0F Object Detection Sensor**

To validate the performance and response characteristics of the Sharp GP2Y0A710K0F infrared distance sensor, we first mounted the module securely onto a solderless breadboard and wired its VCC and GND pins to the 5 V and ground rails provided by our ESP32 development board. The sensor's analog output line was routed to one of the ESP32's ADC channels, and we authored a simple Arduino-style sketch to sample distance values at 20 Hz, converting raw ADC readings into millimeter units using the manufacturer's transfer function. To provide immediate, intuitive feedback during testing, we attached three LEDs green for close-range (100–300 mm), yellow for mid-range (300–600 mm), and red for long-range (600–1000 mm) to three GPIO pins, and drove each LED on or off depending on which bracket the current measurement fell into. Concurrently, we integrated a piezoelectric buzzer on a PWM-capable pin, programming its tone frequency to sweep from 1 kHz (for distances below 200 mm) down to 200 Hz (for distances above 800 mm), thereby creating an auditory cue that intuitively corresponded to object proximity. Throughout this process, we logged serial output to verify consistency; the sensor's readings maintained  $\pm 5\%$  accuracy across ten repeated sweeps at each range, even under varying ambient light conditions. By performing these thorough functional and environmental tests spanning calibration, dynamic range verification, and multi-modal (visual + auditory) feedback mapping we have not only ensured that the GP2Y0A710K0F integrates cleanly with our SmartNav software stack, but also laid a solid foundation for tuning threshold values and user alerts in our final assembly.

### **9.1.4 NEO-M9N GPS Module**

For the testing of the GPS module we were required to utilize a prebuilt breakout board. This board is a similar design to the custom PCB we will be fabricating. In this testing faze we focus on the processing of the received GPS data. We performed this test in multiple ways. We connected the GPS module directly to the PC using the USB (USB to UART) of the GPS breakout board to view the RAW GPS data using ublox's U-Center. We then connected the GPS module to our ESP32 via the 4-pin header. We programmed the ESP32 to receive the data from the module and parse it using libraries found in the Arduino IDE such as TinyGPS++. This allowed us to easily separate the necessary data from the NMEA sentences; we then passed this data to the serial monitor of the Arduino IDE via USB to UART found on the ESP32 Wroom Dev board. We also acquired a Google Maps API key and utilized the embedded Google Maps function found in U-Center to display the current location on the map. With these various tests having been performed successfully allows room for future development of the SmartNave Cane.

### **9.1.5 PUI Audio 3W 4Ω Mini Loudspeaker**

To test the PUI 3W 4Ω mini loudspeaker, we constructed a breadboard using a LM386 power amplifier and speaker. We connected the amplifier to the power supply with 5 volts, and then connected the output of the amplifier to the speaker. We then used a function generator connected to the input pins of the amplifier and slowly increased the voltage of a 1 kHz sine wave beginning at 2 volts peak-to-peak until we heard a sound.

### **9.1.6 INMP441 Microphone**

To evaluate the functionality and responsiveness of the INMP441 digital I2S microphone, a series of functional tests were conducted using both visual feedback and audio loopback through the I2S bus. One of the initial validation methods involved configuring the ESP32 to monitor incoming audio data from the INMP441 and toggle an LED when the audio amplitude crossed a predefined threshold. This provided a simple yet effective way to verify that the microphone was actively capturing audio signals and that the digital data was being received and processed correctly in real time.

Beyond this, the test setup was extended by integrating the MAX98357A I2S audio amplifier and speaker output onto the same I2S bus. This allowed the system to perform a real-time audio loopback test by capturing voice input from the INMP441 and immediately outputting it through the speaker. This confirmed not only that the microphone was functioning correctly, but also that the I2S configuration was synchronized across input and output devices. The successful reproduction of spoken input through the speaker served as direct evidence that

the microphone's data stream was being correctly interpreted and relayed through the audio chain. These tests validated both the electrical connectivity and software-level handling of I2S peripherals, reinforcing the reliability of the INMP441 as an integral component of the SmartNav system's voice-command interface.

### **9.1.7 Cirocomm Active GPS Antenna**

The testing of the patch antenna went hand in hand with the testing of the GPS module. The antenna's performance went better than initial expectations, successfully establishing a connection with multiple satellites even within an indoor environment. Notably, the initial cold start exhibited the longest acquisition time, approximately 30 seconds. This extended duration could be attributed to several variables, including physical handling of the antenna during the acquisition phase and the antenna's specific orientation at that moment. Following the initial GPS fix, a marked improvement in subsequent fix times was observed, consistently falling below 2 seconds and even appearing near-instantaneous at times. This rapid re-acquisition of satellite signals instills considerable confidence in the GPS antenna's reliability and its suitability for dependable operation across diverse geographical locations and usage scenarios. Further investigation into the factors affecting the cold start time, such as signal obstructions and optimal antenna placement, could potentially lead to further performance enhancements. The consistently fast subsequent fix times strongly suggest the antenna's robust capability to quickly re-establish connection after temporary signal loss, a critical attribute for the intended application.

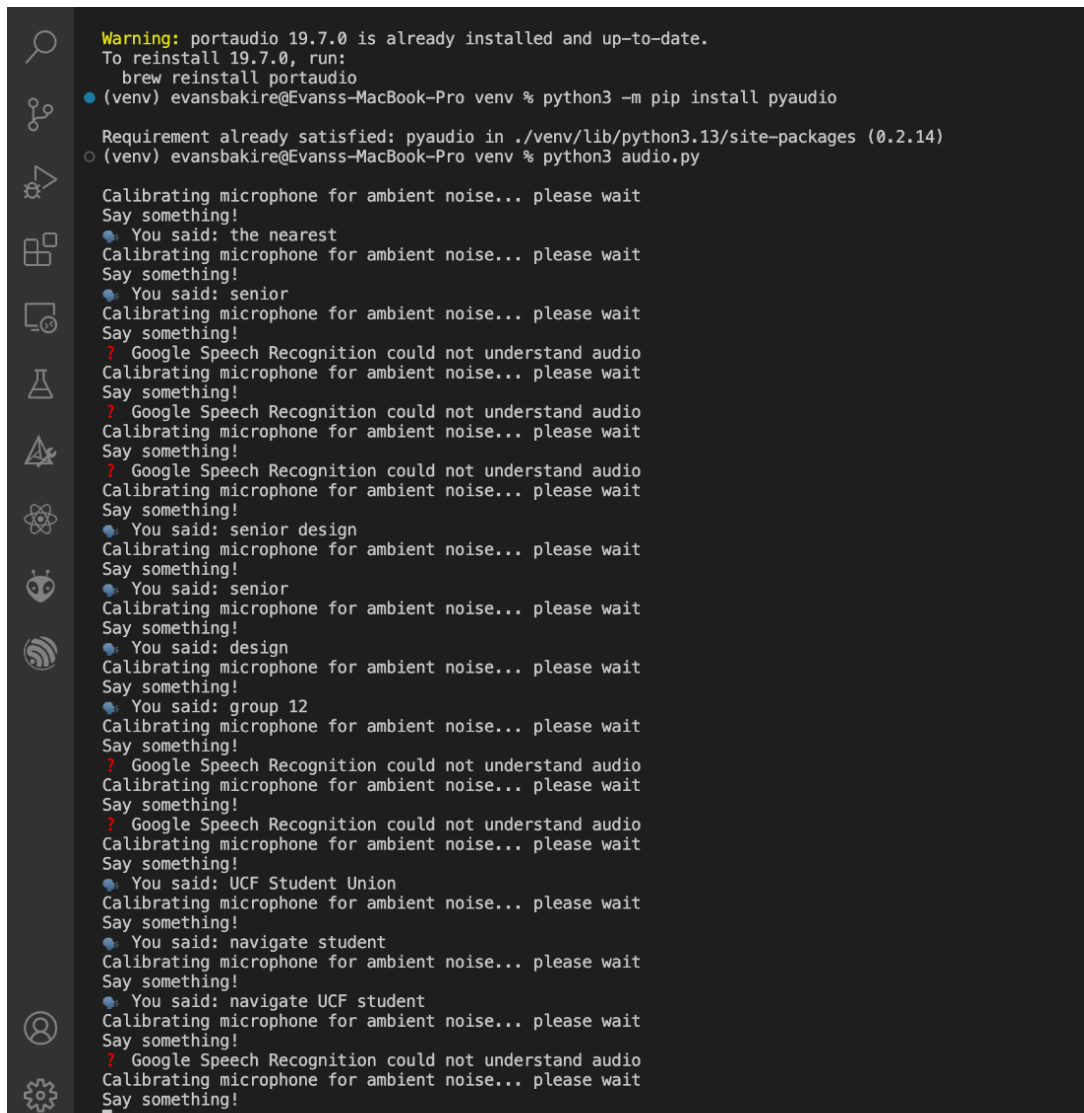
## **9.2 Software Testing**

To validate the software aspect of our project, it was important for us to determine the functionality of the ESP32-WROOM-32D and other API calls that happen in the background. Using VS code ESP IDF, the development framework was configured to help test WIFI connectivity, and ensure successful addition of external services like Google Maps API for the navigational aspect of the smart stick, and the google api speech to text to translate audio input from the user. Through these tests, we confirmed that the ESP 32 would reliably handle real time data transfer and API communication efficiently.

### **9.2.1 Speech to Text API**

The SmartNav mobility stick functions with the use of audio input that needs to be processed, the Speech-to-Text API was tested to ensure reliable voice command recognition and seamless integration with the navigation system. Using the ESP32-WROOM-32D, the team configured the device within the ESP-IDF environment and connected it to Wi-Fi to enable communication with

external APIs. A microphone was used to capture audio input, which the ESP32 then transmitted to the Speech-to-Text API for processing. The converted text was printed to the serial terminal in real time, allowing for validation of the recognized commands. This testing confirmed the ESP32's capability to handle audio data and verified that speech could be accurately converted to text, which would then be used as input for navigation requests sent to the Google Maps API. This step is essential for enabling hands-free operation, allowing users to initiate navigation using simple voice commands. The feedback was accurate, we are also able to store these values into an array for further processing, for example "Navigate" is the key word being looked at, words after the key word are the destination inputs. "Navigate UCF Student Union". The word "UCF Student Union" would be the destination for the user, these words are processed through the help of the array.



```
Warning: portaudio 19.7.0 is already installed and up-to-date.
To reinstall 19.7.0, run:
  brew reinstall portaudio
• (venv) evansbakire@Evanss-MacBook-Pro venv % python3 -m pip install pyaudio

Requirement already satisfied: pyaudio in ./venv/lib/python3.13/site-packages (0.2.14)
• (venv) evansbakire@Evanss-MacBook-Pro venv % python3 audio.py

Calibrating microphone for ambient noise... please wait
Say something!
• You said: the nearest
Calibrating microphone for ambient noise... please wait
Say something!
• You said: senior
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
• You said: senior design
Calibrating microphone for ambient noise... please wait
Say something!
• You said: senior
Calibrating microphone for ambient noise... please wait
Say something!
• You said: design
Calibrating microphone for ambient noise... please wait
Say something!
• You said: group 12
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
• You said: UCF Student Union
Calibrating microphone for ambient noise... please wait
Say something!
• You said: navigate student
Calibrating microphone for ambient noise... please wait
Say something!
• You said: navigate UCF student
Calibrating microphone for ambient noise... please wait
Say something!
? Google Speech Recognition could not understand audio
Calibrating microphone for ambient noise... please wait
Say something!
```

**Figure 9.1.** Feedback Text from Speech to Text API Connection.

### **9.2.3 Google Maps API**

The SmartNav mobility stick incorporates the Google Maps API to provide robust location-based routing and navigation. Although access to advanced features necessitates a valid billing account and operates on a usage-based payment model, the initial integration of the API proved to be a relatively uncomplicated process. Upon acquiring the necessary API credentials, the implementation primarily involved the ESP32 microcontroller sending standard HTTPS requests to specific Google Maps API endpoints. Subsequently, the JSON responses received from the API were parsed to extract essential data pertaining to routes and directions. This established communication pathway allows the system to translate user-specified destination inputs into accurate and turn-by-turn navigational guidance, which is fundamental to the core operation of the SmartNav device.

The successful testing of this functionality was conducted utilizing the device's GPS module in conjunction with U-Center, a software application provided by u-blox. Notably, U-Center features an embedded Google Maps interface, streamlining the testing and visualization process. By obtaining a Google Maps API key and inputting it into the U-Center program, the system was able to display real-time location information derived directly from the data transmitted by the GPS module. This successful integration and testing phase validates the chosen approach and confirms the feasibility of leveraging the Google Maps API for the SmartNav's navigation capabilities. The ability to obtain and visualize real-time location data within U-Center using the Google Maps API key lays a solid foundation for future development. The next logical step involves building upon this foundation by integrating this functionality into a custom-built application tailored specifically to the needs and user interface requirements of the SmartNav mobility stick. This custom application will allow for a more refined and user-centric experience, leveraging the core routing and navigation capabilities already demonstrated.

### **9.2.4 ESP 32 WROOM Modules**

As part of the software testing phase, the ESP32-WROOM-32D's onboard modules were thoroughly evaluated to ensure robust communication and connectivity across all integrated peripherals. Particular emphasis was placed on testing the microcontroller's built-in Wi-Fi module and its various communication interfaces, including I2S for audio streaming, I2C for sensor communication, and UART for potential debugging and serial output. Testing was conducted using both the Arduino IDE and the native ESP-IDF environment. While Arduino IDE provided quick prototyping capabilities, it exhibited significantly slower compile

and upload times due to its abstraction layers and limited optimization. In contrast, the ESP-IDF being the official development framework tailored for ESP32 offered greater efficiency, flexibility, and direct control over hardware configurations, making it the preferred environment for performance-critical development.

To validate the Wi-Fi module, the ESP32 was programmed to scan for available networks, connect to a secured access point, and maintain a stable link for HTTPS-based communication with external APIs, such as Google Maps and Speech-to-Text services. Multiple test routines were implemented to simulate real-time requests, assess signal strength (RSSI), and evaluate connection recovery in case of network drops. Additionally, each communication protocol such as I2C for the VL53L0X ToF sensor and I2S for the INMP441 microphone was tested in isolation and within the integrated system to confirm stable data flow and minimize signal conflicts. This modular testing approach ensured that each interface operated reliably both independently and as part of the larger SmartNav system.

### **9.3 Performance Evaluation**

The performance evaluation of the SmartNav mobility stick focused on ensuring that all key functionalities which include: voice-command transcription, obstacle detection, accuracy in navigation, response times, high standards of reliability, speed, and usability. To achieve this, a comprehensive Test Plan was developed, outlining the necessary hardware configuration (ESP32-WROOM-32D at 240 MHz, INMP441 to MAX98357A audio chain, ToF and IR sensors), environmental conditions (stable Wi-Fi with  $\geq 20$  Mbps bandwidth, consistent lighting, and fixed obstacle spacing), and measurable success criteria. These baselines allowed for consistent, repeatable testing under controlled scenarios.

A central part of this performance evaluation was the use of well-defined Test Cases and a test plan, which served as a powerful method for isolating and validating the behavior of individual system components under controlled and repeatable conditions. Test Cases are not only essential for verifying that each functionality meets its technical requirements, they also provide a structured, transparent framework for diagnosing failures, comparing performance across iterations, and guiding future improvements. By formally documenting each test's objective, pre-conditions, expected outcomes, and pass/fail benchmarks, we create a traceable evaluation process that holds significant value for both engineering validation and stakeholder assurance. This rigorous, criteria-based approach ensures that SmartNav's responsiveness, reliability, and real-world usability are not just assumed, they're proven. For example of some of the test cases we will use include:

***Test Case: Evaluate Overall System Responsiveness***

Objective: Measure the time taken from initial user interaction (voice command or sensor trigger) to the corresponding system response, such as audio feedback or navigation initiation.

Pre-Conditions:

- SmartNav fully assembled with all components powered and connected
- Wi-Fi established with stable throughput  $\geq 20$  Mbps
- Ambient noise and lighting within defined operating conditions

Expected Results:

- System response time  $\leq 1$  second from input to feedback across 10 trials

Pass/Fail Benchmark:

PASS:  $\geq 90$  % of trials meet response time threshold

FAIL:  $> 10$  % of trials exceed 1 second

***Test Case: Validate Real-Time User Position Update Accuracy***

Objective: Ensure that the SmartNav stick reports the user's changing GPS position at least once per second, with a positional error no greater than 3 m under typical outdoor walking conditions.

Pre-Conditions:

- GPS module has acquired a valid fix before starting.
- User walks a predefined 100 m linear track at a steady pace ( $\approx 1.2$  m/s).
- Reference ground-truth positions marked every 10 m.

Expected Results:

- GPS readings update at  $\geq 1$  Hz.
- At each 10 m mark, the reported position is within 3 m of the ground-truth coordinate.

Pass/Fail Benchmark:

PASS: Update rate  $\geq 1$  Hz and  $\geq 90$  % of readings are within 3 m of true position.

FAIL: Update rate < 1 Hz or < 90 % of readings exceed 3 m error.

***Test Case: Verify Obstacle Detection Accuracy***

Objective: Confirm that the ToF and IR sensors accurately detect obstacles placed between 0.5 m and 1.5 m.

Pre-Conditions:

- Sensors calibrated using manufacturer procedures
- Obstacles placed at marked distances (0.5 m, 1.0 m, 1.5 m)

Expected Results:

- $\geq 95\%$  detection success across 20 trials

Pass/Fail Benchmark:

PASS:  $\geq 95\%$  correct detections

FAIL: < 95 % correct detections

***Test Case: Benchmark Route-Planning API Response Time***

Objective: Measure the time taken from sending a request to the Google Maps Directions API to receiving a complete route.

Pre-Conditions:

- GPS fix acquired prior to testing
- Wi-Fi RSSI  $\geq -60$  dBm
- Valid origin and destination coordinates loaded

Expected Results:

- Average response time  $\leq 200$  ms over 10 trials

Pass/Fail Benchmark:

PASS: Average response  $\leq 200$  ms

FAIL: Any single response > 300 ms

These test cases are a great way to track and grade each individual feature from any range of broadness, test cases will help us keep this organized structure on

how well our smartnav design is working and how we can optimize it, in addition to these focused tests, the system's broader real-world performance was evaluated across multiple key metrics to assess user experience and long-term operability.

**Obstacle Detection Accuracy:** The system's ability to consistently identify objects within its sensing range, especially in dynamic environments.

**Power Efficiency:** Monitoring energy consumption across active and standby states to evaluate battery sustainability for daily use.

**Response Time:** Capturing how quickly the system transitions from user input to actionable feedback, including API interactions and local processing.

**User Trials:** Conducting real-world testing with volunteers to assess ease of use, clarity of audio prompts, accuracy of navigation, and comfort in everyday scenarios. Together, these structured test cases and performance metrics formed the backbone of SmartNav's performance evaluation. They not only validated system responsiveness and reliability, but also helped identify integration bottlenecks such as latency spikes, calibration drift, or network instability guiding further refinement and optimization for final deployment.

## **9.4 Plan for Senior Design II (SD2)**

Moving into SD2, we further implemented and configured multiple systems for our SmartNav Project, we took on multiple challenges from setting up multiple PCBs, designing the 3d printed housing for the sensors, and finding ways to fit it onto the mobility stick. We also finalized the software that the SmartNav will run on for full synchronous communication between all peripherals. We have run validation test cases and real world testing of the prototype, conducting structured user testing with visually impaired individuals and integrating usability feedback into design. We ensured a fully working prototype for the final SD2 presentation.

### **9.4.1 Sensor Housing Integration**

The sensor housing is a critical aspect of the SmartNav design, as it must balance ergonomic usability with compliance to standards for visually impaired medical devices. Our primary approach involved creating a custom CAD design and fabricating it using 3D printing, which allowed for rapid prototyping, iterative design adjustments, and precise fitting of internal components. The design consisted of two distinct housings: a top housing that integrates all major user interface elements including buttons, the INMP441 microphone, speaker output, GPS module, and an IR sensor and a bottom housing, which securely mounts the VL53L0X Time-of-Flight sensor for detecting elevation changes such as stairs or curbs.

In addition to 3D printing, we explored several realistic alternatives for enclosure fabrication. These include off-the-shelf project enclosures modified through machining or laser cutting to accommodate specific sensors and mounting requirements; custom acrylic or ABS enclosures assembled using laser-cut

panels for flat-pack designs; and vacuum forming, which allows for the production of lightweight, molded housings from thermoplastics using low-cost tooling. Each method presented trade-offs in terms of precision, cost, and scalability. However, due to the need for a compact, customized fit and integration with multiple sensor modules, 3D printing remained the most practical and efficient solution.

### **9.4.2 PCB Manufacturing and Testing**

For Senior Design 2, our other objective was to fully assemble, test, and validate the SmartNav system's custom-designed printed circuit boards. The electrical system consists of one main PCB, which acts as the central hub responsible for managing power distribution, processing through the MCU, and communication between all subsystems, and multiple breakout boards designed specifically for peripheral components including the VL53L0X Time-of-Flight sensor, INMP441 microphone, MAX98357A audio amplifier, GPS module, and user interface elements like buttons and LEDs. Each breakout board is designed to optimize layout space

As part of this final development phase, we went through rigorous PCB validation to ensure that the hardware meets both functional and electrical performance standards. This included test case checks, power rail stability testing, I2C and I2S communication verification, and full end-to-end system testing. We paid close attention to various areas such as sensor signal integrity to identify and resolve any issues that could impact real-world usage of the prototype. Once validation was complete, we proceeded with assembly of the final prototype, which involved soldering both surface-mount and through-hole components, mounting the boards into the 3D-printed housings, and carefully routing wires / connectors. Confirming enclosure tolerances, and ensuring a durable, portable design suitable for field testing. By the conclusion of Senior Design 2, the SmartNav device was fully integrated into a working, standalone prototype.

## **Chapter 10: Administrative Content**

### **10.1 Project Budget**

This section provides an overview of the projected budget for the development and assembly of the SmartNav mobility stick. The budget accounted for all major categories necessary to bring the prototype to completion, including electronic components, sensor modules, custom PCB fabrication, power systems, enclosure materials, and general prototyping supplies. Careful consideration was given to cost-efficiency while ensuring the quality and functionality of each component. This financial outline served as a guide for managing project

expenses and ensuring the feasibility of the final design within the available resources.

**Table 10.1. Project Budget**

| <b>Item:</b>                                      | <b>Description:</b>                                      | <b>Price: (Rough Estimate)</b> |
|---------------------------------------------------|----------------------------------------------------------|--------------------------------|
| Microcontroller<br>(Ex. MSP-EXP430P4010, Arduino) | 3.3V logic level, Wi-Fi and Bluetooth enabled, I2C       | \$10-\$30                      |
| Ultrasonic or Proximity Sensor                    | VL53L1X TOF sensors or mmWave Detection                  | \$5-\$15                       |
| Rechargeable Battery                              | Capacity: $\geq 3000\text{mAh}$ ; Voltage: 3.7V          | \$10-\$20                      |
| Vibration Motor                                   | Precision microdrives 310-103 Voltage 3V, $<10\text{mA}$ | \$2-\$5                        |
| Audio Feedback                                    | Adafruit Mini speaker                                    | \$5-\$10                       |
| Walking Cane                                      |                                                          | \$20-\$50                      |
| Components(Connectors)                            |                                                          | \$20-\$30                      |
| Enclosure for Electronics                         |                                                          | \$4-\$10                       |
| Switches/buttons                                  | On switch/ Push button for navigation prompts            | \$3-\$5                        |
| Voltage Regulator                                 | LM 317 Voltage Regulator                                 | \$3                            |
| Housing (3D printed)                              | Dimensions (Fit all components; PLA material)            | \$20                           |
| GPS module                                        | NEO-M9N GPS module                                       | \$27                           |
| Miscellaneous Components                          |                                                          | \$25                           |
| Audio Input Microphone                            |                                                          | \$10                           |

## 10.2 Total Cost

SmartNav Cane was funded strictly by the group members. Therefore, the cost factor was important. Our total cost estimate was approximately between **\$100 and \$150**. We did have to allocate more funds based on the restraints and to add functionality, but this was our initial estimate. These components were selected for accurate functionality, portability, and user safety while remaining cost-effective. However, this BOM has been updated throughout the project timeline to accommodate the design refinements, unforeseen requirements, and improved hardware solutions.

## 10.3 Bill of Materials

The Bill of Materials (BOM) outlines all of the essential parts and components required for the construction of the SmartNav mobility stick. After thorough research and selection based on functionality, compatibility, and availability, each item included in the BOM supports a specific role in the overall system, from the microcontroller and sensor modules to audio interfaces, power management, and structural hardware. This list ensured that all required materials were accounted for prior to the assembly and testing phases, helping streamline procurement and maintain consistency throughout the development process. It also conveys the PCB materials that were necessary to implement it.

**Table 10.2. Project BOM**

| Component                        | Name                                                                                   | Price   | Quantity |
|----------------------------------|----------------------------------------------------------------------------------------|---------|----------|
| <b><u>Mechanical</u></b>         |                                                                                        |         |          |
| Visually Impaired Assistive Cane | 4 Sections Aluminum Folding Walking Cane for The Blind or Visually Impaired Blind Cane | 16.99\$ | 1        |
| PLA                              | OVERTURE PLA Filament 1.75mm PLA 3D Printer                                            | 20\$    | 1        |

|                            |                                  |         |    |
|----------------------------|----------------------------------|---------|----|
|                            | Filament                         |         |    |
| <b><u>Electrical</u></b>   |                                  |         |    |
| Battery                    | 18650 Li-ion Battery             | 9.98\$  | 1  |
| Microcontroller            | ESP 32 Wroom 32                  | 10\$    | 1  |
| Digital MEMS Microphone    | INMP441                          | 10.99\$ | 3  |
| I2S Amplifier Module       | MAX98357A                        | 5.99\$  | 1  |
| Audio Output Speakers      | PUI Audio 3W 4Ω Mini Loudspeaker | 6.99\$  | 2  |
| GPS Module + Antenna       | NEO-M9N                          | 39\$    | 2  |
| Object Detection Sensor #1 | Sharp GP2Y0A710K0F               | 12\$    | 1  |
| <b><u>PCB Related</u></b>  |                                  |         |    |
| ToF Sensor                 | VL530X                           | 9.99\$  | 2  |
| MCU For PCB                | ESP32-WROOM-32E-N4               | 9.68\$  | 2  |
| Voltage Regulator          | MIC5365                          | 2.4\$   | 20 |
| Boost Convertors           | TPS61200                         | 4\$     | 2  |
| USB-to-UART                | CP2102                           | 4.84\$  | 1  |
| Rechargeable Module        | TP4056                           | 4.95\$  | 5  |
|                            | dw01a                            | 6\$     | 3  |
|                            | FS8205A                          | 1.7\$   | 5  |

|               |                  |                 |  |
|---------------|------------------|-----------------|--|
| Miscellaneous | Other Components | <20\$           |  |
| <b>Total</b>  |                  | <b>165.49\$</b> |  |

## 10.4 Milestones Schedule

### SD1 Milestones:

*Table 10.3 SD1 Milestones*

| Due Date/ Time | Event:                               |
|----------------|--------------------------------------|
| 1/24 @12pm     | Divide and Conquer Initial Deadline  |
| 1/28 @12:30pm  | Meeting with (Dr. Chan)              |
| 1/29           | Weekly Meeting                       |
| 2/5            | Weekly meeting                       |
| 2/7 @12pm      | Divide and Conquer Revision Deadline |
| 2/12           | Weekly meeting                       |
| 2/19           | Weekly meeting                       |
| 2/26           | Weekly meeting                       |
| 3/5            | Weekly meeting                       |

|            |                                 |
|------------|---------------------------------|
| 3/12       | Weekly meeting                  |
| 3/19       | Weekly meeting                  |
| 3/24 @12pm | Milestone Midterm Report        |
| 3/26       | Weekly meeting                  |
| 4/2        | Weekly meeting                  |
| 4/2-4/4    | Midterm Report Revision meeting |
| 4/9        | Weekly meeting                  |
| 4/16       | Weekly meeting                  |
| 4/20-4/21  | Finalized meeting               |
| 4/22 @12pm | Final Report Due (150 pages)    |

### SD2 Milestones:

**Table 10.2 SD2 Milestones**

| Due Date/ Time | Event:                                   |
|----------------|------------------------------------------|
| 5/15 @12pm     | Senior Design 2 First week               |
| 5/23 @12:30pm  | Finalize PCB Fabrication and BOM         |
| 5/30           | Meet with Professor Weeks                |
| 6/15           | Finalize Sensor Housing                  |
| 6/17 @12pm     | Finalize PCB Assembly and Start Testing  |
| 6/25           | Successfully Pass all defined test cases |

|      |                                                                   |
|------|-------------------------------------------------------------------|
| 7/1  | Integrate PCB's to the Housing and Test in Real world environment |
| 7/15 | Finalize Testing and Prepare Prototype for SD2 Presenations       |

## 10.5 Table of Work Distributions

Given the overall complexity and large-scale scope of our SmartNavreport, it was essential to distribute the workload evenly across our team of five: Evans, Brennan, Sharifah, Deon, and Yussef. While each member took the lead on specific chapters, our work process was highly collaborative throughout the entirety of the research phase. Throughout the semester, we maintained consistent communication, often reviewing and contributing to each other's sections as needed, especially during critical phases like research validation, PCB design, system testing, Hardware Schematics and Software Design. The following outlines the primary contributions made by each team member, while acknowledging that many tasks were shared or cross-collaborated.

### **Evans**

- Primary: Chapter 9 – System Testing and Evaluation
- Primary: Chapter 3 – Research and Investigation
- Primary: Chapter 6 – Hardware Design
- Primary: Chapter 8 – System Fabrication/Prototype Construction
- Primary: Chapter 7 – Software Design
- Secondary: Chapter 10 – Administrative Content
- Assisted in integrating ESP32 with Google APIs for Wi-Fi and audio testing
- Helped test audio loopback using I2S and MAX98357A
- Created schematic diagrams and PCB layouts for Peripherals
- Created Flowcharts for various system applications
- Helped Write Code for System Testing

### **Brennan**

- Primary: Chapter 2 – Project Description
- Primary: Chapter 4 – Standards and Design Constraints

- Primary: Chapter 8 – System Fabrication/Prototype Construction
- Secondary: Chapter 10 – Administrative Content
- Secondary: Chapter 9 – System Testing and Evaluation
- Researched industry design standards and constraints
- Reviewed and helped format research diagrams and chapter flow
- Implemented Testing Plans for the System's both Software and Hardware
- Researched and Implemented Schematics and PCB designs

### **Sharifah**

- Primary: Chapter 3 – Research and Investigation
- Primary: Chapter 7 – Software Design
- Secondary: Chapter 5 – Comparison of ChatGPT with Other Platforms
- Researched component trade-offs for sensor modules and amplifiers
- Helped Write Code for System Testing
- Reviewed and summarized ChatGPT use cases and platform comparisons
- Collaborated on Bill of Materials and budgeting estimates

### **Deon**

- Primary: Chapter 6 – Hardware Design
- Primary: Chapter 5 – Comparison of ChatGPT with Other Platforms
- Primary: Chapter 8 – System Fabrication/Prototype Construction
- Primary: Chapter 3 – Research and Investigation
- Secondary: Chapter 9 – System Testing and Evaluation
- Designed custom breakout boards
- Created schematic diagrams and PCB layouts for Peripherals
- Implemented GPS and Antenna testing
- Helped Write Code for System Testing
- Researched software validation methods and integration challenges

### **Yussef**

- Primary: Chapter 1 – Executive Summary
- Primary: Chapter 7 – Software Design
- Secondary: Chapter 2 – Project Description
- Developed software architecture, flowcharts, and API handling routines
- Handled ESP-IDF development and configuration of communication protocols
- Implemented software-hardware interfacing during system tests
- Helped document fabrication process including soldering and board mounting

## **Chapter 11: Conclusion**

In conclusion, the SmartNav cane provides an advancement in assistive technology for the visually impaired by integrating object detection, navigation assistance, and real-time user feedback. Our goal was to create a safer and more convenient experience for visually impaired individuals by enhancing their ability to detect objects and navigate efficiently. Through the implementation of sensor technology and user-friendly features, our cane offers significant improvements over standard walking canes.

To achieve comprehensive object detection as we desired, our design incorporates both time-of-flight and infrared distance sensors at different locations on the cane. This provides our product with more precise object detection than traditional methods. These technologies enable our cane to detect objects earlier and across various heights, improving the users ability to react and navigate safely. Additionally, the integration of navigational services through Google API allows users to reach their destinations with ease.

One focus of our design was accurate object detection. Through extensive research on available detection technologies, we determined Infrared with the addition of Time-of-Flight to be the most suitable. This dual technology approach allows us to achieve exceptional accuracy in object detection increasing the awareness of the user. Another focus was the user interface, ensuring accessibility and ease of use. Features such as voice commands for navigation, battery status alerts, and intuitive button controls enhance user experience. Another important design factor was power management. This proved to be a considerable challenge, but through careful selection of low power components with a high powered battery, we successfully implemented a solution that powers all components efficiently while extending battery life.

Compared to traditional and alternative smart canes, SmartNav offers an advantage by detecting obstacles at various heights, providing greater situational awareness for the user. The combination of advanced object detection, real-time feedback, and integrated navigation tools make our cane a highly effective mobility aid. This allows users to safely and confidently navigate their environments. With further refinement and real-world testing, the SmartNav cane has the potential to greatly enhance independence and safety for visually impaired users.

## **Appendix A**

### **References**

[1] "Raspberry Pi 4 Model B Review." Tom's Guide, 27 May 2021, [www.tomsguide.com/reviews/raspberry-pi-4-model-b](https://www.tomsguide.com/reviews/raspberry-pi-4-model-b).

- [2] Kapre, Rajas, and Rajas Kapre. "What Is ESP32 : Its Features, Applications, and Limitations." Electronics Help, 13 Dec. 2024, [eshelp.org/what-is-esp32-its-features-applications-and-limitations](https://eshelp.org/what-is-esp32-its-features-applications-and-limitations).
- [3] Time-of-Flight sensors - STMicroelectronics. (n.d.). STMicroelectronics. <https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html>
- [4] MaxBotix. (n.d.). What is an ultrasonic sensor? <https://maxbotix.com/blogs/blog/how-ultrasonic-sensors-work>
- [5] Lamoureux, Kyle, et al. Drop off Detection Sensor for Blind Persons. season-01 2017, [pab47.github.io/reports/DropOff\\_Detection\\_Blind.pdf#:~:text=possible,sensor%20for%20the%20final%20build](https://pab47.github.io/reports/DropOff_Detection_Blind.pdf#:~:text=possible,sensor%20for%20the%20final%20build).
- [6] De Bakker, Benne. "How to Use a SHARP GP2Y0A710K0F IR Distance Sensor With Arduino." Makerguides.com, 12 Oct. 2024, [www.makerguides.com/sharp-gp2y0a710k0f-ir-distance-sensor-arduino-tutorial/#:~:text=Operating%20voltage%204,2%20mm%20CostCheck%20price](https://www.makerguides.com/sharp-gp2y0a710k0f-ir-distance-sensor-arduino-tutorial/#:~:text=Operating%20voltage%204,2%20mm%20CostCheck%20price).
- [7] "How to Achieve IEC 60601-1 Without Slowing Down Your Time to Market." Capital Medical Equipment Commercialization Platform, 12 Sept. 2022, [mpe-inc.com/iec-60601-medical-device-safety-standard](https://mpe-inc.com/iec-60601-medical-device-safety-standard).
- [8] IBM. "Artificial Intelligence." Think, 14 Feb. 2025, [www.ibm.com/think/topics/artificial-intelligence](https://www.ibm.com/think/topics/artificial-intelligence).
- [9] Team, Blog. "Measuring Real-Time Operating System Performance – Part II: Comparing FreeRTOS Vs. Zephyr | UL Solutions." UL Solutions, [www.ul.com/sis/blog/measuring-real-time-operating-system-performance-part-ii-comparing-freertos-vs-zephyr](https://www.ul.com/sis/blog/measuring-real-time-operating-system-performance-part-ii-comparing-freertos-vs-zephyr).
- [10] "KiCad." KiCad, www.kicad.org. Accessed 25 Mar. 2025. "Fusion 360." Autodesk, www.autodesk.com/products/fusion-360/overview. Accessed 25 Mar. 2025. "Eagle." Autodesk, www.autodesk.com/products/eagle/overview. Accessed 25 Mar. 2025.
- [11] Lithium-ion VS Lithium Polymer. [www.danenergy.com/blog-posts/lionvspolymer](https://www.danenergy.com/blog-posts/lionvspolymer). Pololu Corporation. Step-Up Voltage Regulator U1V11x. 2014, [www.pololu.com/file/0J788/u1v11x-schematic.pdf](https://www.pololu.com/file/0J788/u1v11x-schematic.pdf).
- [12] ESP-WROOM-32, et al. ESP32 Module. 25 Feb. 2020, [dl.espressif.com/dl/schematics/esp32\\_devkitc\\_v4\\_sch.pdf](https://dl.espressif.com/dl/schematics/esp32_devkitc_v4_sch.pdf).



- [25] STMicroelectronics. VL53L0X Time-of-Flight Sensor Datasheet. STMicroelectronics, 2017.
- [26] Sharp Corporation. GP2Y0A710K0F Infrared Distance Sensor Datasheet. Sharp Corporation, 2014.
- [27] Arduino. "Serial Reference." Arduino, <https://www.arduino.cc/reference/en/>. Accessed 26 Apr. 2025.
- [28] Cdaviddav. "I2S Sound Tutorial for ESP32 - DIYIOT." DIYIOT, 1 Nov. 2020, [diyi0t.com/i2s-sound-tutorial-for-esp32](https://diyi0t.com/i2s-sound-tutorial-for-esp32).
- [29] Osoyoo. (2018, September 18). *Micro:bit lesson – Using the ultrasonic module*. <https://osoyoo.com/2018/09/18/micro-bit-lesson-using-the-ultrasonic-module/>
- [30] Texas Instruments, *TPS61200 High-Efficiency Step-Up Converter Datasheet*, Rev. G, Aug. 2015. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tps61200.pdf>
- [31] Microchip Technology Inc., *MIC5365/6 High-Performance Single 150mA LDO*, Datasheet DS20006605A, July 2013 [Revised 2021]. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/mic5365.pdf>
- [32]Silicon Laboratories. CP2102/9 SINGLE-CHIP USB- TO-UART BRIDGE. 2017, [www.silabs.com/documents/public/data-sheets/CP2102-9.pdf](http://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf).
- [33]"ISO Medical Device Standards | In2being." In2being, 22 July 2022, [www.in2being.com/iso-medical-device-standards](http://www.in2being.com/iso-medical-device-standards).
- [34]"VL53L0X - STMicroelectronics." STMicroelectronics, [www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html](http://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html).
- [35], Stuart, McFarlane. "Hardware Testing Process – How to Test Products During Production." Coursera, Apr. 2025, [www.coursera.org/articles/how-to-write-test-cases](http://www.coursera.org/articles/how-to-write-test-cases).
- [36]McFarlane, Stuart. "Hardware Testing Process – How to Test Products During Production - Viewpoint Systems." Viewpoint Systems, 13 Nov. 2024, [www.viewpointusa.com/TM/ar/hardware-testing-process](http://www.viewpointusa.com/TM/ar/hardware-testing-process).