Course Project Report

Course: Robotics Rehabilitation

Smart Blind Assistance System Using ESP32 and Smartphone

Preparator: Amir Hossein Osooli

Winter 2026

Table of Content

Abstract

This report presents the design and implementation of a smart blind assistance system based on an ESP32 microcontroller and a smartphone application. The system uses multiple ultrasonic sensors to detect obstacles in different directions and communicates wirelessly with a mobile phone using Wi-Fi. Detected obstacles are reported to the user through real-time voice feedback generated by the phone application. The proposed system focuses on low cost, simplicity, reliability, and accessibility, leveraging built-in smartphone text-to-speech features. Experimental results demonstrate stable communication, accurate distance estimation, and effective indoor navigation assistance.

1. Introduction

Visual impairment significantly affects independent mobility and safe navigation. Traditional aids such as white canes provide tactile feedback but limited information about obstacle distance and direction. Recent advances in embedded systems and mobile computing enable the development of electronic travel aids that enhance environmental awareness.

This project aims to design a compact, low-cost blind assistance system capable of detecting obstacles in multiple directions and delivering intuitive auditory feedback through a smartphone. The system avoids complex hardware interfaces and relies on widely available devices, making it accessible and scalable.

2. Related Work

Various blind assistance systems have been proposed in the literature, for example:

Ultrasonic cane systems: Use distance sensors to detect obstacles and provide vibration or sound alerts. These systems are simple but often lack directional information. [1]

Wearable haptic devices: Use vibration motors to indicate obstacle proximity. While effective, they may require user training and custom hardware.

Camera-based vision systems: Employ computer vision and machine learning for object detection. These systems offer rich information but are computationally expensive and sensitive to lighting conditions. [2]

Smartphone navigation applications: Provide GPS-based guidance but perform poorly indoors and cannot detect nearby obstacles.

Compared to existing solutions, the proposed system emphasizes simplicity, low cost, and reliable indoor performance, while leveraging smartphone accessibility features instead of custom feedback hardware.

3. System Overview

The proposed system consists of two main components:

ESP32-based sensing unit: responsible for obstacle detection and data processing.

Smartphone application: responsible for user interaction and voice feedback.

The ESP32 operates as a Wi-Fi access point and hosts a lightweight HTTP server. The smartphone connects directly to the ESP32 and periodically requests obstacle information, which is returned as human-readable text.

4. Hardware Design

4.1 Components

ESP32 microcontroller

Three ultrasonic distance sensors (HC-SR04)

Power supply (battery or Power Bank)

Smartphone (iOS or Android)

Packaging

Cover and Bands

4.2 Sensor Placement

Three ultrasonic sensors are placed strategically:

Front sensor: Detects obstacles directly ahead

Right sensor: Detects obstacles on the right side

Back sensor: Detects obstacles behind the user

This configuration minimizes blind spots and enables directional awareness.

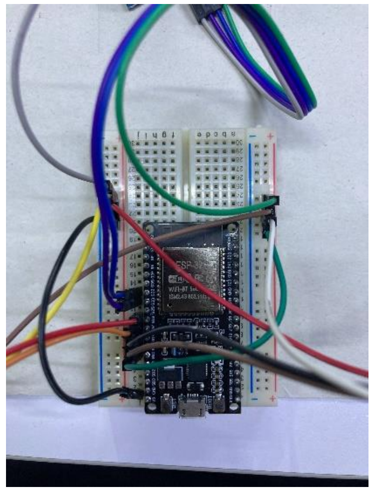

Figure 1: Assembled circuits

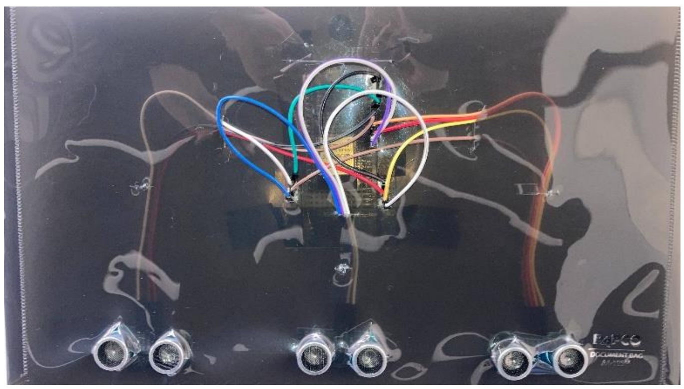

Figure 2: Packed version of device

Figure 3: Final mounted version of the device

5. Software Design

5.1 ESP32 Firmware

The ESP32 firmware is responsible for:

Configuring Wi-Fi in Access Point mode

Hosting an HTTP server

Triggering ultrasonic sensors

Processing distance measurements

Generating text-based alert messages

5.2 Distance Classification

Measured distances are classified into semantic levels:

Very close

Close

Approaching

Clear path

The closest detected obstacle and its direction are selected for feedback.

6. Mobile Application

The mobile application connects to the ESP32 Wi-Fi network and sends periodic HTTP requests to retrieve obstacle information. The received text messages are converted into speech using the phone's built-in text-to-speech engine.

Key features include:

Voice-based feedback

Accessibility-friendly design

No requirement for external servers

The application was developed using a AI app builder, demonstrating that advanced functionality can be achieved without native mobile development.

7. Communication Protocol

Communication between the ESP32 and the smartphone uses HTTP over a local Wi-Fi connection. The ESP32 exposes a single endpoint that returns plain text messages describing obstacle conditions.

This approach provides:

Low latency

Platform independence

Ease of debugging

Compatibility with accessibility tools

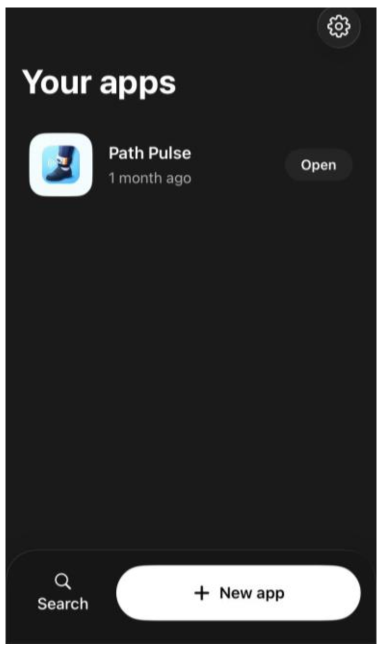

Figure 4: The Designed Application

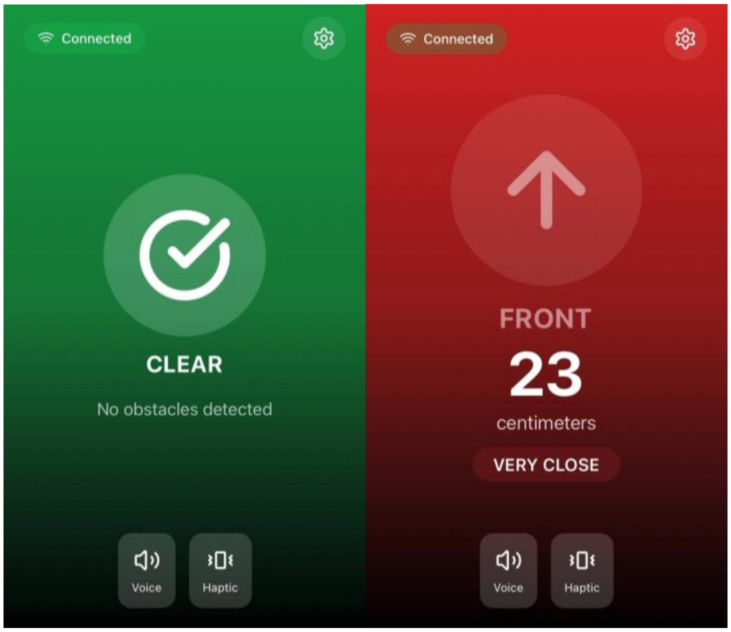

Figure 5: The Designed Application in use, clear - alarm mode

8. Experimental Results

The system was tested in indoor environments with various obstacle configurations.

Results showed:

Stable Wi-Fi connectivity

Accurate distance measurements within sensor limits

Reliable directional detection

Clear and timely voice feedback

9. Limitations

Despite its effectiveness, the system has some limitations:

Ultrasonic sensors perform poorly on soft or angled surfaces

The system is optimized for indoor environments

Requires the user to carry a smartphone

No object classification capability

10. Future Work

Potential improvements include:

Replacing the ultrasound sensors with cameras

Using machine learning for object recognition

Integrating GPS for outdoor navigation

11. Conclusion

This project successfully demonstrates a smart blind assistance system using an ESP32 and a smartphone. By combining low-cost hardware with smartphone accessibility features, the system provides effective real-time obstacle awareness. The modular design and simple communication protocol make it a strong foundation for further research and real-world deployment.

12. References

[1]. https://youtu.be/7hmTb7Iqm4s?si=_gJMG1NSumhHkq_j

[2]. https://youtu.be/yWwEnwK0olg?si=dlNe0nVouvwwiKgA