VISIONASSIST

Assistive Eyewear System for Visually Impaired Individuals

Integrating Real-Time Obstacle Detection and Optical Character Recognition

Technical Report – 2026

B.G.R.V. Senanayake

Abstract

VisionAssist is an innovative assistive eyewear system designed to enhance the independence and safety of visually impaired individuals. The system integrates real-time obstacle detection using Time-of-Flight (TOF) laser sensing technology with Optical Character Recognition (OCR) capabilities, providing users with both navigation assistance and text reading functionality.

The device consists of two primary components. smart eyewear equipped with a TOF sensor and camera, and a haptic handband that delivers vibration-based alerts. This dual-component architecture enables the system to serve deaf-blind users effectively, as feedback is provided through tactile sensations rather than auditory signals.

The obstacle detection system employs a VL53L1X TOF sensor capable of measuring distances up to 4 meters, with four distinct alert zones providing graduated haptic feedback based on proximity. The OCR functionality utilizes Google Cloud Vision API to extract text from captured images, which is then converted to speech through a web-based interface.

Communication between the eyewear and handband units is achieved through ESP-NOW protocol, ensuring low-latency wireless transmission. The system demonstrates the practical application of embedded systems, computer vision, and assistive technology principles in creating accessible solutions for individuals with visual impairments.

Keywords: Assistive Technology, Visual Impairment, ESP32, TOF Sensor, OCR, Haptic Feedback, ESP-NOW, Text-to-Speech

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Visual impairment affects approximately 2.2 billion people globally, according to the World Health Organization (WHO, 2023). Among these individuals, a significant portion faces daily challenges in navigation and accessing printed information. Traditional assistive tools such as white canes and guide dogs, while effective, have inherent limitations in detecting obstacles at various heights and do not provide text reading capabilities.

The advancement of embedded systems, sensors, and cloud computing has opened new possibilities for developing intelligent assistive devices. Modern microcontrollers offer sufficient processing power to handle real-time sensor data and wireless communication, while cloud-based APIs provide sophisticated capabilities such as optical character recognition without requiring extensive local computational resources.

Time-of-Flight (TOF) sensors represent a significant advancement over traditional ultrasonic sensors used in assistive devices. TOF sensors measure distance by calculating the time taken for emitted light to return after reflecting off objects, providing more accurate and faster measurements compared to sound-based alternatives. This technology enables precise obstacle detection critical for user safety.

The integration of haptic feedback systems allows information to be conveyed through touch, making devices accessible to individuals who are both visually and hearing impaired. Vibration patterns can effectively communicate varying levels of urgency without relying on visual or auditory channels.

1.2 Problem Statement

Visually impaired individuals face two primary challenges in their daily lives:

Navigation Safety: Moving through unfamiliar or dynamic environments poses significant risks. Existing solutions such as white canes have limited range and cannot detect obstacles at head or chest height, leading to potential injuries.

Information Access: Printed text in the environment including signs, labels, documents, and displays remains largely inaccessible without sighted assistance or specialized equipment that is often expensive and not portable.

Additionally, many assistive devices rely on auditory feedback, which presents challenges for deaf-blind individuals and can be socially intrusive in quiet environments. There is a need for an affordable, portable device that addresses both navigation and text-reading needs while providing non-auditory feedback options.

1.3 Objectives

The primary objectives of this project are:

1. Design and develop a wearable obstacle detection system that alerts users to nearby objects through haptic feedback, with graduated intensity based on proximity.

2. Implement optical character recognition functionality that enables users to capture images of text and have the content read aloud through text-to-speech.

3. Create a dual-component architecture with wireless communication between eyewear and handband units for optimal sensor placement and user comfort.

4. Ensure accessibility for deaf-blind users by utilizing vibration-based alerts rather than auditory signals as the primary feedback mechanism.

5. Develop a web-based interface that allows smartphone integration for text-to-speech output and system monitoring.

1.4 Scope and Limitations

Scope

This project encompasses:

- Hardware design and integration of sensors, microcontrollers, and feedback mechanisms
- Firmware development for both eyewear and handband units
- Implementation of ESP-NOW wireless communication protocol
- Integration with Google Cloud Vision API for OCR functionality
- Development of a responsive web interface with text-to-speech capabilities
- Documentation and testing of the complete system

Limitations

The following limitations are acknowledged:

Limitation	Description
TOF Sensor Range	The VL53L1X sensor has an effective range of approximately 4 meters, with optimal accuracy within 2 meters
Battery Life	Continuous WiFi connectivity and sensor operation result in significant power consumption, limiting battery-powered operation time
OCR Dependency	Text recognition requires internet connectivity for Google Cloud Vision API access
WiFi Requirement	It is easiest to ensure reliable ESP-NOW communication when both devices are connected to the same Wi-Fi network, because this forces them onto the same Wi-Fi channel. ESP-NOW itself doesn’t require a router or SSID match, it only requires that peers operate on the same channel.
Single Direction Detection	The current design detects obstacles only in the forward direction of the sensor

Chapter 2: Literature Review

2.1 Existing Assistive Technologies

2.1.1 Traditional Mobility Aids

The white cane remains the most widely used mobility aid for visually impaired individuals. It provides tactile feedback about the immediate environment through direct contact. However, it requires extensive training, provides limited range, and cannot detect obstacles above ground level (Hersh & Johnson, 2010).

Guide dogs offer intelligent navigation assistance and can identify complex hazards. However, they require significant training for both the dog and user, involve ongoing care costs, and are not suitable for all individuals or environments (Whitmarsh, 2005).

2.1.2 Electronic Travel Aids (ETAs)

Ultrasonic-based devices such as the UltraCane and Miniguide use sound waves to detect obstacles and provide vibration or audio feedback. These devices extend detection range compared to traditional canes but suffer from limitations including:

• Interference from environmental noise

• Limited accuracy in detecting small or soft objects

• Relatively slow response times (Dakopoulos & Bourbakis, 2010)

2.1.3 Smart Glasses and Wearables

Recent developments include camera-based smart glasses that utilize computer vision for obstacle detection and scene description. Products such as OrCam MyEye and Envision Glasses provide text reading capabilities. However, these devices typically cost between $2,000 and $5,000, limiting accessibility for many users (Csapó et al., 2015).

2.1.4 Smartphone-Based Solutions

Applications such as Seeing AI (Microsoft) and Lookout (Google) leverage smartphone cameras and cloud processing for text recognition and object identification. While cost-effective, these solutions require users to hold and aim their phones, which can be impractical during navigation (Ahmetovic et al., 2016).

2.2 Comparison of Solutions

Solution	Detection Range	Text Reading	Feedback Type	Approximate Cost	Deaf-Blind Accessible
White Cane	1-1.5m	No	Tactile	$20-50	Yes
Guide Dog	Variable	No	Behavioral	$25,000+ (training)	Yes
UltraCane	2-4m	No	Vibration	$700-1,000	Yes
OrCam MyEye	N/A	Yes	Audio	$3,500-4,500	No
Envision Glasses	Limited	Yes	Audio	$2,500-3,500	No
Smartphone Apps	N/A	Yes	Audio	Free (requires phone)	No
VisionAssist	Up to 4m	Yes	Vibration + TTS	<$100	Yes

Table 2.1: Comparison of Existing Assistive Devices

2.3 Research Gap

The literature review reveals several gaps that VisionAssist addresses:

Cost Barrier: Most comprehensive solutions combining navigation and text reading are prohibitively expensive for many users in developing regions.

Deaf-Blind Accessibility: The majority of text-reading devices rely exclusively on audio output, excluding deaf-blind users.

Integrated Solution: Few affordable devices combine both obstacle detection and OCR functionality in a single wearable system.

Graduated Feedback: Many electronic travel aids provide binary (obstacle present/absent) feedback rather than graduated proximity information.

VisionAssist addresses these gaps by providing an affordable, integrated solution with haptic feedback suitable for deaf-blind users and graduated alert zones for intuitive distance perception.

Chapter 3: System Design

3.1 System Architecture

VisionAssist employs a distributed architecture consisting of three primary components:

Eyewear Unit (ESP32-S3): Worn on the user's head, housing the TOF sensor for forward obstacle detection, camera for image capture, and touch sensor for user input.

Handband Unit (ESP32-C3): Worn on the user's wrist or hand, containing the vibration motor for haptic feedback delivery.

Smartphone Interface: Provides text-to-speech output through a web browser and displays system status information.

This distributed design offers several advantages:

Optimal Sensor Placement: The TOF sensor and camera are positioned at head height for detecting obstacles at face level and capturing text at natural reading positions.

Effective Feedback Delivery: Vibration on the hand is more perceptible than on the head and does not interfere with other senses.

Reduced Eyewear Weight: Offloading the vibration motor to a separate unit keeps the eyewear lightweight and comfortable.

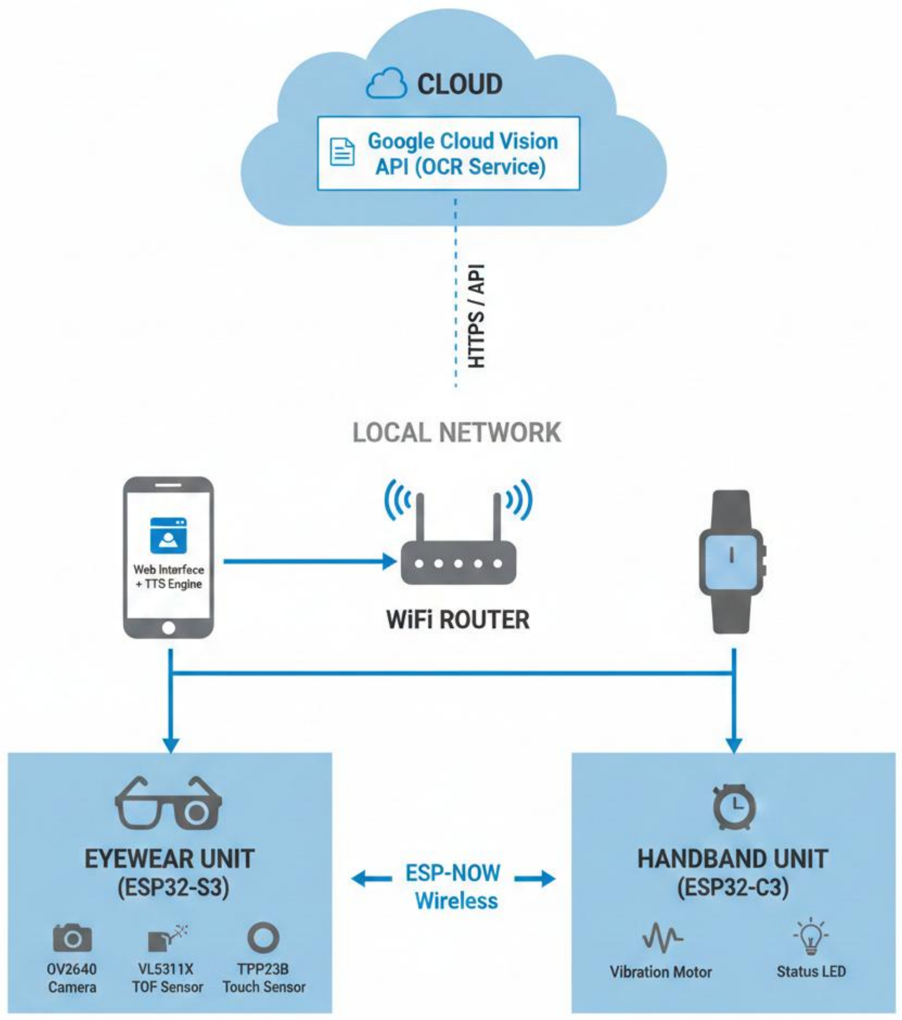


Figure 3.1: System Architecture Overview

3.2 Design Requirements

Requirement ID	Category	Description	Priority
REQ-01	Functional	Detect obstacles within 0.1m to 4m range	High
REQ-02	Functional	Provide graduated haptic feedback based on distance	High
REQ-03	Functional	Capture images on user command	High
REQ-04	Functional	Extract text from images using OCR	High
REQ-05	Functional	Convert extracted text to speech	High
REQ-06	Functional	Pause vibration alerts during text reading	Medium
REQ-07	Performance	Obstacle detection latency < 100ms	High
REQ-08	Performance	Wireless communication latency < 50ms	High
REQ-09	Usability	Single-touch activation for OCR	High
REQ-10	Usability	Intuitive vibration patterns	High
REQ-11	Hardware	Lightweight eyewear unit	Medium
REQ-12	Hardware	Comfortable handband design	Medium

Table 3.1: System Design Requirements

3.3 Block Diagram

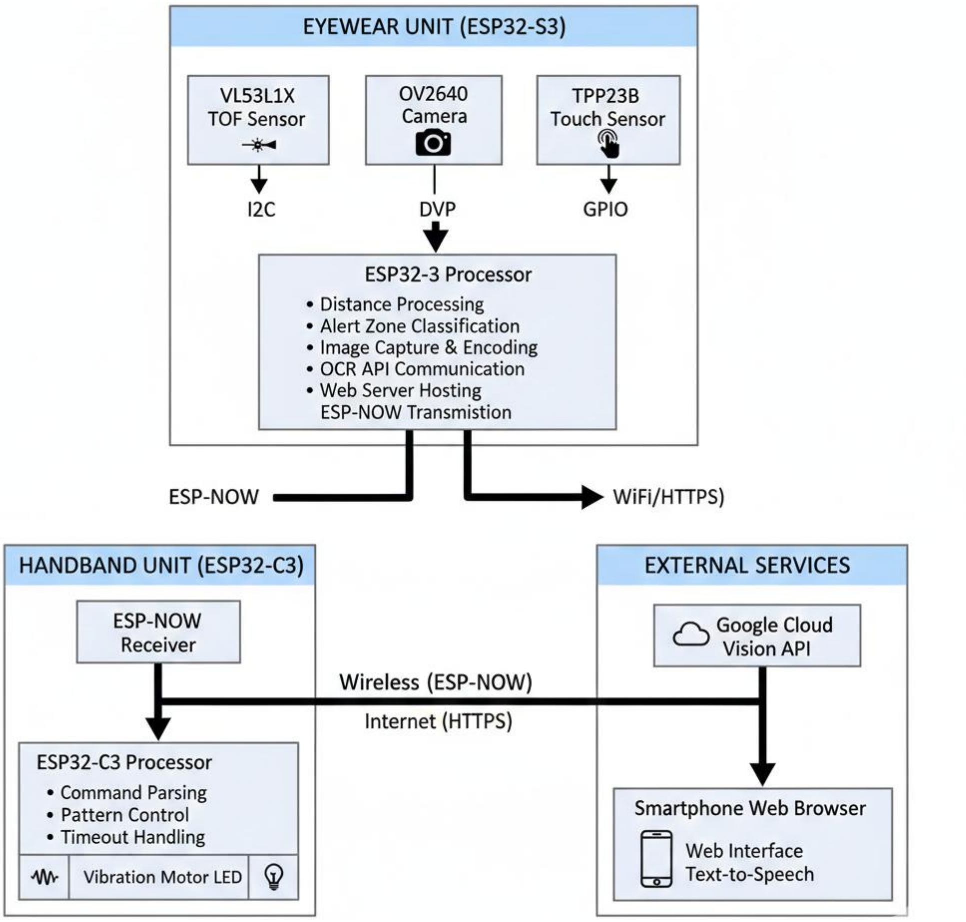


Figure 3.2: System Block Diagram

3.4 Data Flow

The system operates with the following data flows:

Obstacle Detection Flow:

1. VL53L1X sensor measures distance continuously (20ms intervals)
2. ESP32-S3 processes readings with smoothing algorithm
3. Distance is classified into alert zones
4. Alert pattern is transmitted via ESP-NOW to handband
5. ESP32-C3 activates vibration motor with corresponding pattern

OCR Flow:

1. User touches TTP223B sensor
2. ESP32-S3 captures image via OV2640 camera
3. Image is encoded to Base64 format
4. HTTP POST request sent to Google Cloud Vision API
5. API returns extracted text
6. Text is stored and flagged as new
7. Web interface polls for new text
8. Browser's Web Speech API converts text to speech

Reading Mode Flow:

1. Touch detection triggers "PAUSE" command
2. ESP-NOW transmits pause signal to handband
3. Vibration motor stops during OCR/TTS process
4. Normal operation resumes after TTS completion or timeout

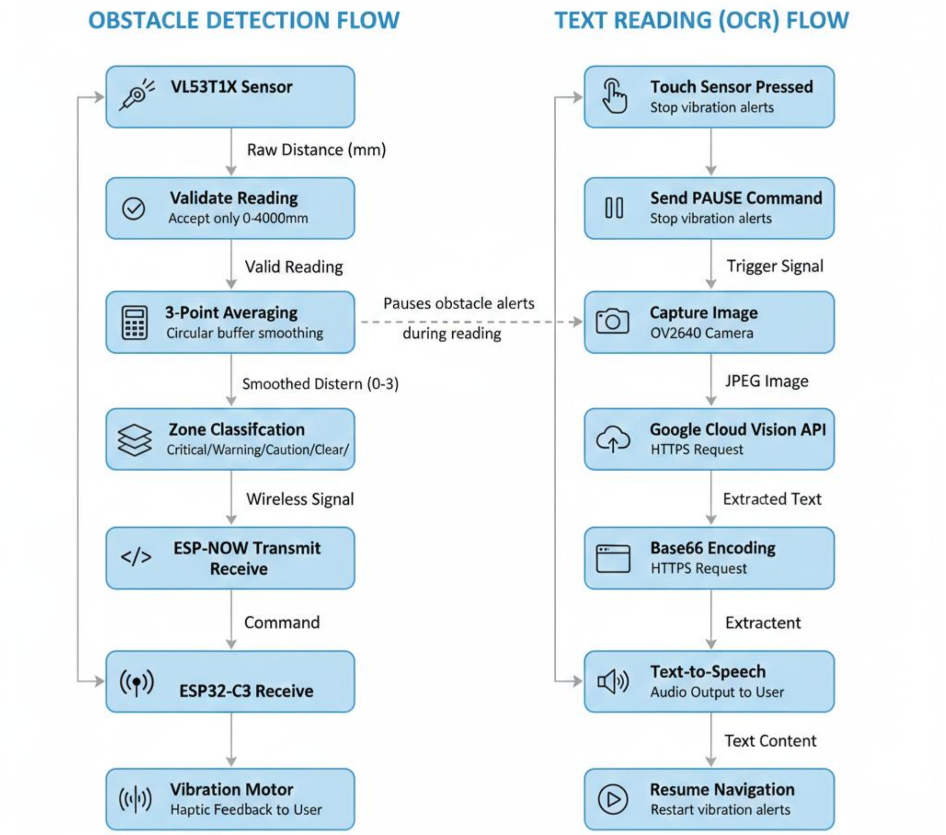


Figure 3.3: Data Flow Diagram

Chapter 4: Hardware Components

4.1 Component Overview

Component	Model	Quantity	Function
Microcontroller (Eyewear)	Seeed Studio XIAO ESP32S3 Sense	1	Main processing, camera, WiFi, ESP-NOW
Microcontroller (Handband)	ESP32-C3 Super Mini	1	Vibration control, ESP-NOW receiver
Distance Sensor	VL53L1X TOF Laser Module	1	Obstacle detection
Touch Sensor	TTP223B Capacitive Touch Switch	1	User input for OCR trigger
Vibration Motor	Coin Motor Module (3-pin PCB)	1	Haptic feedback
Camera	OV2640 (Built into XIAO)	1	Image capture for OCR

Table 4.1: Component List

4.2 Seeed Studio XIAO ESP32S3 Sense

The Seeed Studio XIAO ESP32S3 Sense serves as the primary processing unit for the eyewear component. This compact development board integrates an ESP32-S3 microcontroller with an OV2640 camera module, making it ideal for vision-based applications.

Figure 4.1: Seeed Studio XIAO ESP32S3 Sense

Key Features

Specification	Value
Processor	ESP32-S3R8 (Xtensa LX7 dual-core, 240MHz)
SRAM	8MB PSRAM
Flash	8MB
Wireless	WiFi 802.11 b/g/n, Bluetooth 5.0 (BLE)
Camera	OV2640 (2MP)
GPIO Pins	11 usable
Interface	I2C, SPI, UART, I2S
Dimensions	21 x 17.5 mm
Power Input	5V via USB-C or 3.7V battery

Table 4.2: ESP32-S3 XIAO Specifications

Selection Rationale

The XIAO ESP32S3 Sense was selected for the following reasons:

Integrated Camera: Eliminates need for separate camera module and wiring

Compact Size: Essential for wearable eyewear application

PSRAM: 8MB PSRAM enables handling of image data for OCR

Dual-Core Processor: Allows concurrent handling of sensor reading, web server, and communication

ESP-NOW Support: Enables low-latency communication with handband unit

Reference Documentation

[Official Product Page]

[ESP32-S3 Datasheet]

4.3 ESP32-C3 Super Mini

The ESP32-C3 Super Mini serves as the controller for the handband unit, receiving wireless commands and driving the vibration motor.

Figure 4.2: ESP32-C3 Super Mini

Key Features

Specification	Value
Processor	ESP32-C3 (RISC-V single-core, 160MHz)
SRAM	400KB
Flash	4MB
Wireless	WiFi 802.11 b/g/n, Bluetooth 5.0 (BLE)
GPIO Pins	13 usable
Interface	I2C, SPI, UART
Dimensions	22.52 x 18 mm
Power Input	5V via USB-C or 3.3V direct

Table 4.3: ESP32-C3 Super Mini Specifications

Selection Rationale

Ultra-Compact Size: Minimal footprint ideal for wristband integration

Low Power Consumption: RISC-V architecture offers efficiency advantages

ESP-NOW Compatibility: Seamless communication with ESP32-S3

Cost-Effective: Significantly lower cost than larger development boards

Sufficient GPIO: Adequate pins for motor control and status LED

Reference Documentation

[ESP32-C3 Datasheet]

4.4 VL53L1X TOF Laser Sensor

The VL53L1X is a Time-of-Flight (TOF) laser-ranging sensor that provides accurate distance measurements using infrared light.

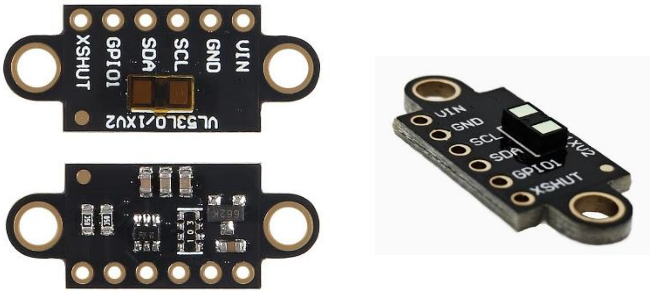


Figure 4.3: VL53L1X TOF Laser Sensor Module

Operating Principle

The sensor emits modulated infrared laser light (940nm wavelength, Class 1 eye-safe) and measures the time taken for the light to reflect back from objects. This time is converted to distance using the speed of light:

$$\text{Distance} = (\text{Speed of Light} \times \text{Time}) / 2$$

The division by 2 accounts for the round-trip journey of the light.

Key Specifications

Specification	Value
Ranging Distance	40mm to 4000mm
Accuracy	±3% (typical)
Field of View	27°
Ranging Frequency	Up to 50Hz
Interface	I2C (up to 400kHz)
Operating Voltage	2.6V to 3.5V
Laser Wavelength	940nm (invisible IR)
Laser Class	Class 1 (eye-safe)

Table 4.4: VL53L1X Sensor Specifications

Advantages Over Ultrasonic Sensors

Why TOF Instead of Ultrasonic?

Traditional distance measurement projects commonly use ultrasonic sensors (such as HC-SR04) due to their low cost and availability. However, for wearable assistive devices, ultrasonic sensors present significant drawbacks:

Size and Portability:

The HC-SR04 ultrasonic sensor measures approximately 45mm × 20mm × 15mm with two large cylindrical transducers. This bulky form factor is unsuitable for integration into eyewear, where aesthetics and comfort are critical. The VL53L1X TOF sensor, in contrast, measures only 4.9mm × 2.5mm × 1.56mm (small enough to be nearly invisible when mounted on eyewear frames).

Weight:

Sensor	Approximate Weight
HC-SR04 Ultrasonic	8-10 grams
VL53L1X TOF Module	1-2 grams

For a head-mounted device worn for extended periods, every gram matters. The TOF sensor's minimal weight contributes to user comfort and reduces fatigue.

Power Consumption:

Parameter	VL53L1X (TOF)	HC-SR04 (Ultrasonic)
Operating Current	20mA (typical)	30-50mA (during measurement)
Standby Current	5μA	2mA
Peak Current	40mA	50-100mA

The TOF sensor's lower power consumption extends battery life, a critical factor for portable assistive devices.

Performance Comparison:

Parameter	VL53L1X (TOF)	HC-SR04 (Ultrasonic)
Measurement Speed	Up to 50Hz	Up to 20Hz
Accuracy	±3%	±3mm + 1%
Minimum Range	40mm	20mm
Maximum Range	4000mm	4000mm
Beam Width	27°	15-30°
Environmental Interference	Low	High (affected by sound sources, wind)
Detection of Soft Materials	Good	Poor (sound absorption)

Additional Advantages:

- **No acoustic interference:** Ultrasonic sensors can interfere with each other and are affected by ambient noise. TOF uses invisible infrared light.
- **Better detection of soft objects:** Ultrasonic waves are absorbed by clothing and soft surfaces, whereas infrared light reflects reliably.
- **Eye-safe laser:** The 940nm Class 1 laser is completely safe for use near the eyes.

These factors made the VL53L1X TOF sensor the clear choice for VisionAssist, despite its higher cost compared to ultrasonic alternatives.

Reference Documentation

[VL53L1X Datasheet]

[API User Manual]

4.5 OV2640 Camera Module

The OV2640 is a 2-megapixel CMOS image sensor integrated into the XIAO ESP32S3 Sense board.

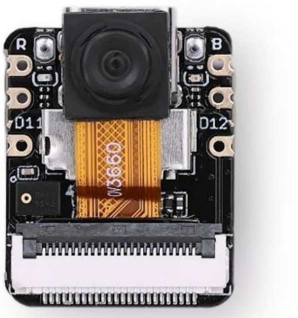


Figure 4.4: OV2640 Camera Module (Built-in)

Key Specifications

Specification	Value
Resolution	2MP (1600 x 1200 max)
Output Formats	JPEG, RGB565, YUV422
Frame Rate	15fps @ UXGA, 30fps @ SVGA
Field of View	65°
Interface	DVP (Digital Video Port)
Operating Voltage	2.5V to 3.0V

Configuration Used

For OCR functionality, the camera is configured with the following settings:

Resolution: SVGA (800 x 600) - Balance of quality and processing speed

Format: JPEG - Reduced data size for API transmission

Quality: 12 (scale 0-63, lower = higher quality)

Reference Documentation

[OV2640 Datasheet]

4.6 TTP223B Capacitive Touch Sensor

The TTP223B is a digital capacitive touch sensor module that provides simple touch detection without mechanical buttons.

Figure 4.5: TTP223B Capacitive Touch Sensor

Key Specifications

Specification	Value
Operating Voltage	2.0V to 5.5V
Output Type	Digital (HIGH/LOW)
Response Time	60ms (typical)
Detection Mode	Toggle or Momentary (configurable)
Sensing Area	Approximately 10mm diameter

Operating Principle

The sensor detects changes in capacitance when a conductive object (human finger) approaches the sensing pad. The TTP223B IC compares the measured capacitance against a threshold and outputs a digital signal when a touch is detected.

Integration Notes

The sensor is connected to GPIO7 on the ESP32-S3

A software debounce of 1000ms prevents multiple triggers

Touch detection is disabled during active OCR processing

Reference Documentation

[TTP223B Datasheet]

4.7 Vibration Motor Module

The vibration motor module consists of a coin-type (pancake) vibration motor mounted on a small PCB with integrated driver circuitry.

Figure 4.6: Vibration Motor Module

Module Description

The module includes:

• Coin/pancake vibration motor (approximately 10mm diameter)

• PCB with integrated driver components

• Three-pin interface (IN, VCC, GND)

• Onboard resistors and capacitors for signal conditioning

Key Specifications

Specification	Estimated Value
Operating Voltage	3.3V to 5V
Motor Type	Eccentric Rotating Mass (ERM)
Motor Diameter	~10mm
Input Signal	Digital (3.3V logic compatible)
Current Consumption	50-100mA (typical)

Integration Benefits

The pre-built module eliminates the need for:

• External transistor driver circuit

• Flyback diode for motor protection

• Current limiting resistors

This simplifies the handband design and reduces component count.

4.8 Pin Connections and Wiring

ESP32-S3 XIAO (Eyewear Unit) Connections

Component	Function	GPIO Pin	Notes
VL53L1X	SDA	GPIO 5	I2C Data
VL53L1X	SCL	GPIO 6	I2C Clock
VL53L1X	VIN	3.3V	Power supply
VL53L1X	GND	GND	Ground
TTP223B	Signal	GPIO 7	Digital input
TTP223B	VCC	3.3V	Power supply
TTP223B	GND	GND	Ground
OV2640	All pins	Internal	Built-in camera

Table 4.5: ESP32-S3 Pin Connections

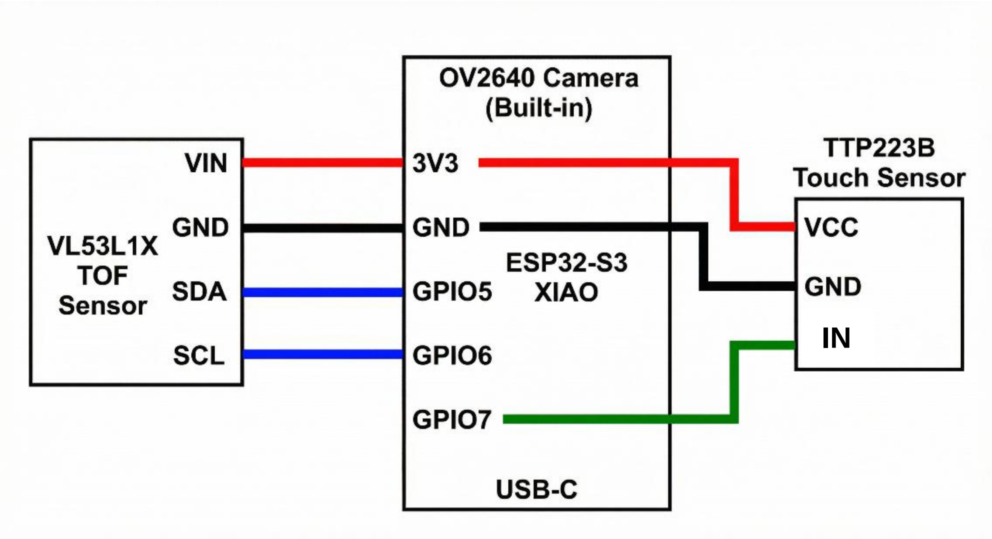


Figure 4.7: Eyewear Unit Wiring Diagram

ESP32-C3 Super Mini (Handband Unit) Connections

Component	Function	GPIO Pin	Notes
Vibration Motor Module	IN (Signal)	GPIO 4	PWM capable (digital used)
Vibration Motor Module	VCC	3.3V or 5V	Check motor specs
Vibration Motor Module	GND	GND	Ground

Table 4.6: ESP32-C3 Pin Connections

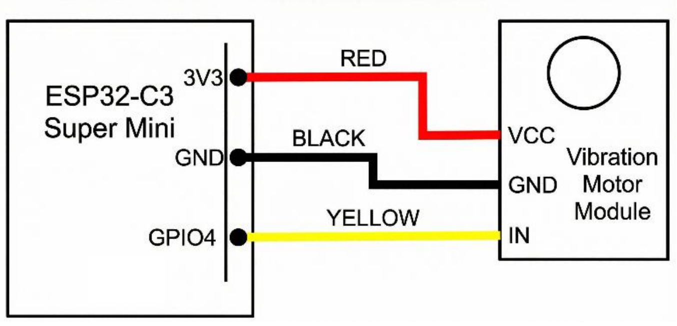


Figure 4.8: Handband Unit Wiring Diagram

I2C Configuration Notes

The VL53L1X sensor uses I2C communication with the following configuration:

Default Address: 0x29

Clock Speed: 400kHz (Fast Mode)

Pull-up Resistors: Typically built into VL53L1X module; if absent, add 4.7kΩ resistors on SDA and SCL lines to 3.3V

Chapter 5: Software Implementation

5.1 Development Environment

PlatformIO IDE

The firmware for both units was developed using PlatformIO, an open-source ecosystem for embedded development that integrates with Visual Studio Code.

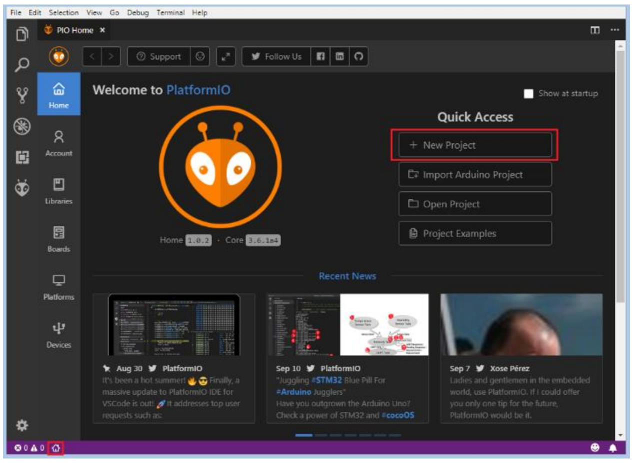


Figure 5.1: PlatformIO IDE Interface

Development Tools

Tool	Version	Purpose
Visual Studio Code	1.85+	Code editor
PlatformIO Core	6.1+	Build system and library manager
Espressif32 Platform	6.4.0	ESP32 toolchain and framework
Arduino Framework	-	Hardware abstraction layer

Libraries Used

Eyewear Unit (ESP32-S3):

Library	Version	Purpose
Adafruit VL53L1X	3.1.0	TOF sensor driver
ArduinoJson	Latest	JSON parsing (if needed)
esp_camera (built-in)	-	Camera driver
WiFi (built-in)	-	WiFi connectivity
WebServer (built-in)	-	HTTP server
esp_now (built-in)	-	ESP-NOW protocol

Handband Unit (ESP32-C3):

Library	Version	Purpose
WiFi (built-in)	-	WiFi connectivity
esp_now (built-in)	-	ESP-NOW protocol

Reference Documentation

[PlatformIO Documentation]

[ESP-IDF Programming Guide]

[Arduino-ESP32 Documentation]

5.2 Eyewear Firmware (ESP32-S3)

The eyewear firmware handles multiple concurrent tasks:

- TOF sensor reading and distance processing
- Touch sensor monitoring
- Camera capture and OCR processing
- ESP-NOW transmission
- Web server for smartphone interface

Firmware Architecture

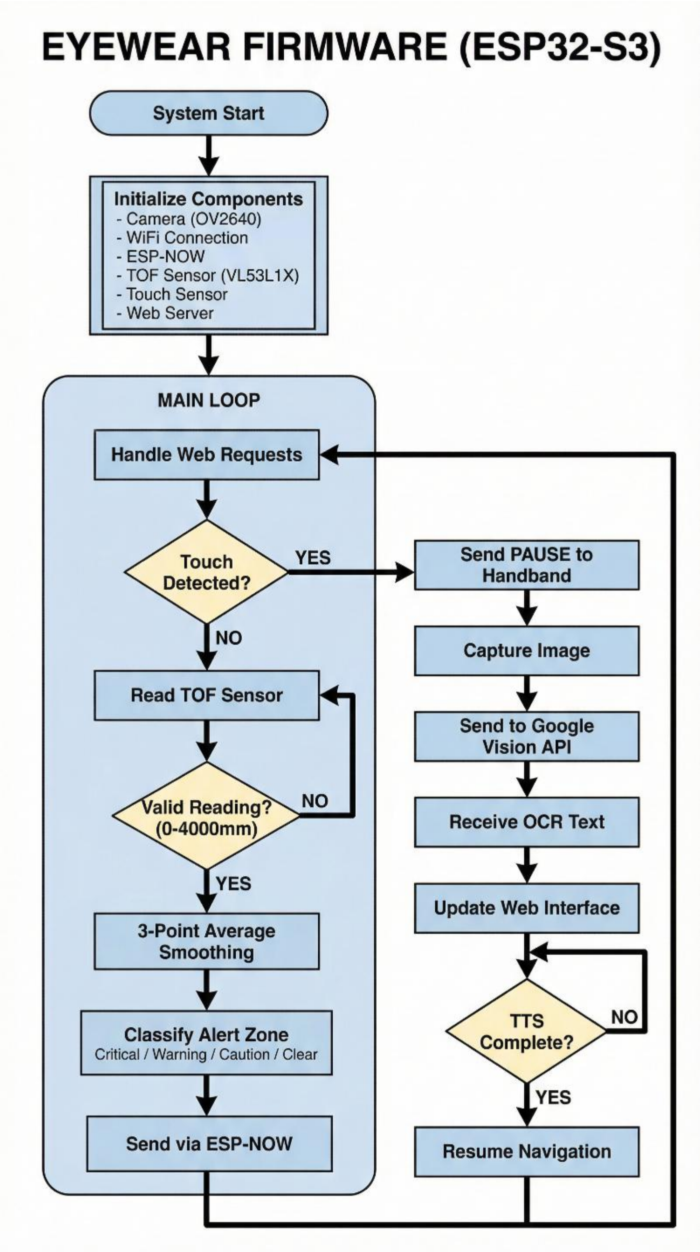


Figure 5.2: Eyewear Firmware Flowchart

Distance Processing Algorithm

The VL53L1X TOF sensor occasionally produces erroneous readings due to environmental factors such as reflective surfaces, ambient light interference, or objects outside the detection range. When the sensor fails to receive a valid reflection, it typically returns a value of approximately 5000mm. Using such erroneous values directly would cause incorrect distance calculations and unreliable haptic feedback. To address this challenge, the firmware implements a multi-stage filtering and smoothing algorithm. Alert Zone Configuration

Stage 1: Invalid Reading Rejection

The first stage acts as a gatekeeper, preventing erroneous readings from entering the processing pipeline. When a new distance reading arrives from the sensor, the firmware immediately checks whether the value falls within a valid range:

```
if (rawDistance > 0 && rawDistance < 4000) {

   // Process this reading

}
```

This condition rejects two categories of problematic readings. Readings of zero or negative values indicate sensor communication errors or initialization issues. Readings of 4000mm or above typically represent the sensor's error state (commonly 5000mm) or measurements beyond the reliable detection range. The 4000mm threshold was chosen because the VL53L1X datasheet specifies this as the maximum reliable ranging distance under ideal conditions, and the furthest alert zone (CAUTION) only extends to 2000mm, making readings beyond 4000mm irrelevant for obstacle detection purposes.

By rejecting invalid readings at this stage, the erroneous 5000mm values never proceed to the averaging calculation. This is the critical insight of the algorithm, rather than attempting to detect and remove outliers from an average, the system prevents outliers from ever being included.

Stage 2: Circular Buffer Smoothing

Even valid readings can exhibit minor fluctuations due to sensor noise, surface texture variations, or slight movements. To produce stable distance values, the firmware maintains a circular buffer containing the three most recent valid readings:

```
static int fastReadings[3] = {5000, 5000, 5000};

static int fastIndex = 0;

fastReadings[fastIndex] = rawDistance;

fastIndex = (fastIndex + 1) % 3;

smoothedDistance = (fastReadings[0] + fastReadings[1] + fastReadings[2]) / 3;
```

The circular buffer operates by overwriting the oldest reading with each new valid measurement. The index variable tracks the current position in the buffer and wraps around using the modulo operator. When index reaches 3, the modulo operation resets it to 0, causing the next reading to overwrite the oldest value in position 0.

The choice of three readings represents a deliberate balance between smoothing effectiveness and response speed. With the sensor operating at 50Hz (one reading every 20ms), three readings span only 60ms. This brief window provides sufficient smoothing to eliminate minor fluctuations while maintaining near-instantaneous response to actual distance changes. Using more readings would increase smoothing but introduce noticeable delay in obstacle alerts, an unacceptable trade-off for a safety-critical application.

The buffer is initialized with 5000mm values, which might seem contradictory given that 5000mm represents an error condition. However, this initialization is intentional and safe. During the first 60ms of operation, as the buffer fills with valid readings, the average will be artificially high. A high distance value corresponds to the CLEAR zone, meaning no vibration alert occurs. This represents the safest possible default behavior. The system assumes no obstacle until it has gathered sufficient valid data to prove otherwise. After three valid readings (60ms), all initialization values are overwritten and the average reflects only actual measurements.

Stage 3: Timeout Protection

The sensor might occasionally enter a state where it cannot obtain valid readings for an extended period. This could occur if the user points the sensor at the sky, at a very distant wall, or if the sensor experiences a temporary malfunction. Without timeout protection, the smoothed Distance variable would retain its last valid value indefinitely, potentially indicating a nearby obstacle that no longer exists or missing a new obstacle entirely.

```
if (now - lastGoodReading > 1000) {
    smoothedDistance = 5000;
    lastGoodReading = now;
}
```

If no valid reading has been received for 1000ms (one second), the firmware sets the smoothed distance to 5000mm and resets the timer. The 5000mm value places the system in the CLEAR zone, stopping any vibration alerts. This represents the safer choice. If the sensor cannot confirm an obstacle exists, the system should not continuously alarm the user about a potentially non-existent hazard.

The one-second timeout provides sufficient time for the sensor to recover from brief interference while preventing stale data from persisting indefinitely. The timer reset prevents the system from repeatedly triggering the timeout condition every loop iteration once activated.

Why This Approach Works

The three-stage algorithm creates a robust pipeline where each stage addresses a specific failure mode. The rejection filter eliminates gross errors before they can affect calculations. The circular buffer smooths minor variations without introducing significant delay. The timeout mechanism handles extended sensor failures gracefully.

Critically, the algorithm maintains the separation between error values and valid measurements. The 5000mm value serves two distinct purposes It is the sensor's error indication and the system's safe default state, but these uses never conflict because error readings are rejected before averaging and timeout protection explicitly assigns 5000mm only when appropriate.

This approach proved highly effective during practical testing, producing stable distance readings and reliable alert zone transitions even in challenging environments with reflective surfaces and varying lighting conditions.

Zone	Distance Range	Pattern Code	Command String
Clear	> 2000mm	0	"CLEAR"
Caution	1600mm - 2000mm	3	"CAUTION"
Warning	1300mm - 1600mm	2	"WARNING"
Critical	< 1300mm	1	"CRITICAL"
Reading Mode	N/A	0	"PAUSE"

Table 5.1: Alert Zone Configuration

Touch-Triggered OCR Process

When a touch is detected:

1. Set distancePaused = true to pause vibration feedback
2. Send "PAUSE" command to handband via ESP-NOW
3. Capture image from camera
4. Encode image to Base64
5. Send POST request to Google Cloud Vision API
6. Parse response for extracted text
7. Store text and flag as new for web interface
8. Resume normal operation after TTS completion or timeout (60 seconds)

Key Configuration Parameters

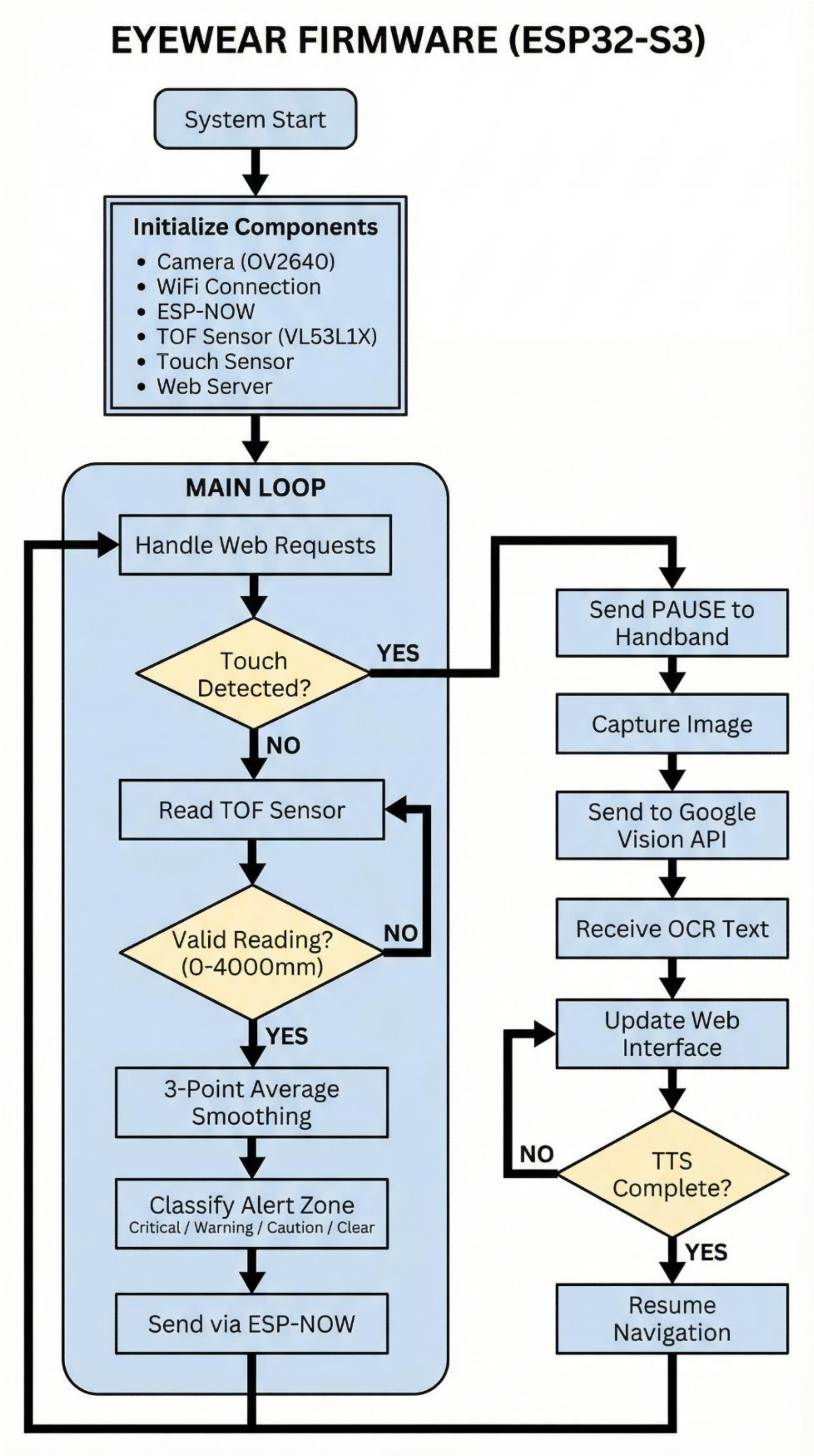
```
// Distance thresholds (millimeters)
#define CRITICAL_DISTANCE 1300
#define WARNING_DISTANCE 1600
#define CAUTION_DISTANCE 2000

// Touch sensor configuration
#define TOUCH_PIN 7
#define TOUCH_DEBOUNCE 1000  // 1 second debounce

// TTS timeout
#define TTS_TIMEOUT 60000  // 60 seconds maximum
```

5.3 Handband Firmware (ESP32-C3)

The handband firmware is simpler, focusing on receiving ESP-NOW messages and controlling the vibration motor with appropriate patterns.

Firmware Architecture

Figure 5.3: Handband Firmware Flowchart



Vibration Pattern Implementation

Pattern	Behavior	ON Time	OFF Time
0 (Clear)	Motor OFF	-	-
1 (Critical)	Continuous ON	Always	-
2 (Warning)	Fast pulse	400ms	150ms
3 (Caution)	Slow pulse	300ms	600ms

Table 5.2: Vibration Pattern Definitions

Pattern Code Implementation

```
switch(currentPattern) {
  case 1:  // CRITICAL - Continuous
    digitalWrite(MOTOR_PIN, HIGH);
    break;
  case 2:  // WARNING - Fast pulses
    if (motorState) {
      if (now - lastToggle >= 400) {
        digitalWrite(MOTOR_PIN, LOW);
        motorState = false;
        lastToggle = now;
      }
    } else {
      if (now - lastToggle >= 150) {
        digitalWrite(MOTOR_PIN, HIGH);
        motorState = true;
        lastToggle = now;
      }
    }
    break;
  case 3:  // CAUTION - Slow pulses
    if (motorState) {
      if (now - lastToggle >= 300) {
        digitalWrite(MOTOR_PIN, LOW);
        motorState = false;
        lastToggle = now;
      }
    } else {
      if (now - lastToggle >= 600) {
        digitalWrite(MOTOR_PIN, HIGH);
        motorState = true;
        lastToggle = now;
      }
```

```
        }
        break;
    default:  // CLEAR - OFF
        digitalWrite(MOTOR_PIN, LOW);
        break;
}
```

Connection Timeout Handling

If no message is received for 1500ms, the handband assumes connection loss and:

- Stops the vibration motor
- Resets pattern to 0
- Waits for reconnection

5.4 ESP-NOW Communication Protocol

ESP-NOW is a wireless communication protocol developed by Espressif that enables low-latency, connectionless data transmission between ESP devices.

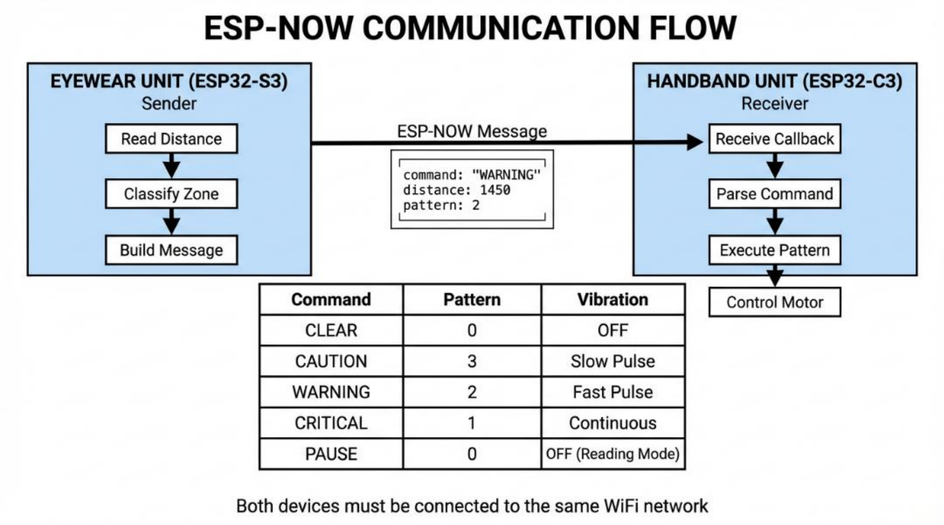


Figure 5.4: ESP-NOW Communication Flow

Message Structure

```
typedef struct {
    char command[32];  // Command string: "CLEAR", "CAUTION", etc.
    int distance;      // Distance in millimeters
    int pattern;       // Pattern code: 0, 1, 2, or 3
} Message;
```

Field	Type	Description
command	char[32]	Command string: "CLEAR", "CAUTION", etc.
distance	int	Distance in millimeters
pattern	int	Pattern code: 0, 1, 2, or 3

Table 5.3: ESP-NOW Message Structure

Communication Requirements

Critical Requirement: Both ESP32-S3 and ESP32-C3 must operate on the same WiFi channel for ESP-NOW communication to function. ESP-NOW does not require both devices to be on the same WiFi network or connected to any network at all, it is a peer-to-peer protocol independent of WiFi infrastructure. Using the same WiFi network is simply the easiest method to guarantee both devices remain on the same channel.

Sender Implementation (S3)

```
// Initialize peer
memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = WiFi.channel();  // Must match WiFi channel
peerInfo.encrypt = false;
peerInfo.ifidx = WIFI_IF_STA;
esp_now_add_peer(&peerInfo);

// Send message
strcpy(outgoingData.command, "CRITICAL");
```

```
outgoingData.distance = smoothedDistance;
outgoingData.pattern = 1;
esp_now_send(broadcastAddress, (uint8_t*)&outgoingData, sizeof(outgoingData));
```

Receiver Implementation (C3)

```
// Callback function registered during setup
void OnDataRecv(const uint8_t *mac, const uint8_t *data, int len) {
  memcpy(&incomingData, data, sizeof(incomingData));
  currentPattern = incomingData.pattern;
  lastReceived = millis();
}

// Registration
esp_now_register_recv_cb(OnDataRecv);
```

Reference Documentation

[ESP-NOW User Guide]

5.5 Google Cloud Vision API Integration

The OCR functionality utilizes Google Cloud Vision API's DOCUMENT_TEXT_DETECTION feature to extract text from captured images.

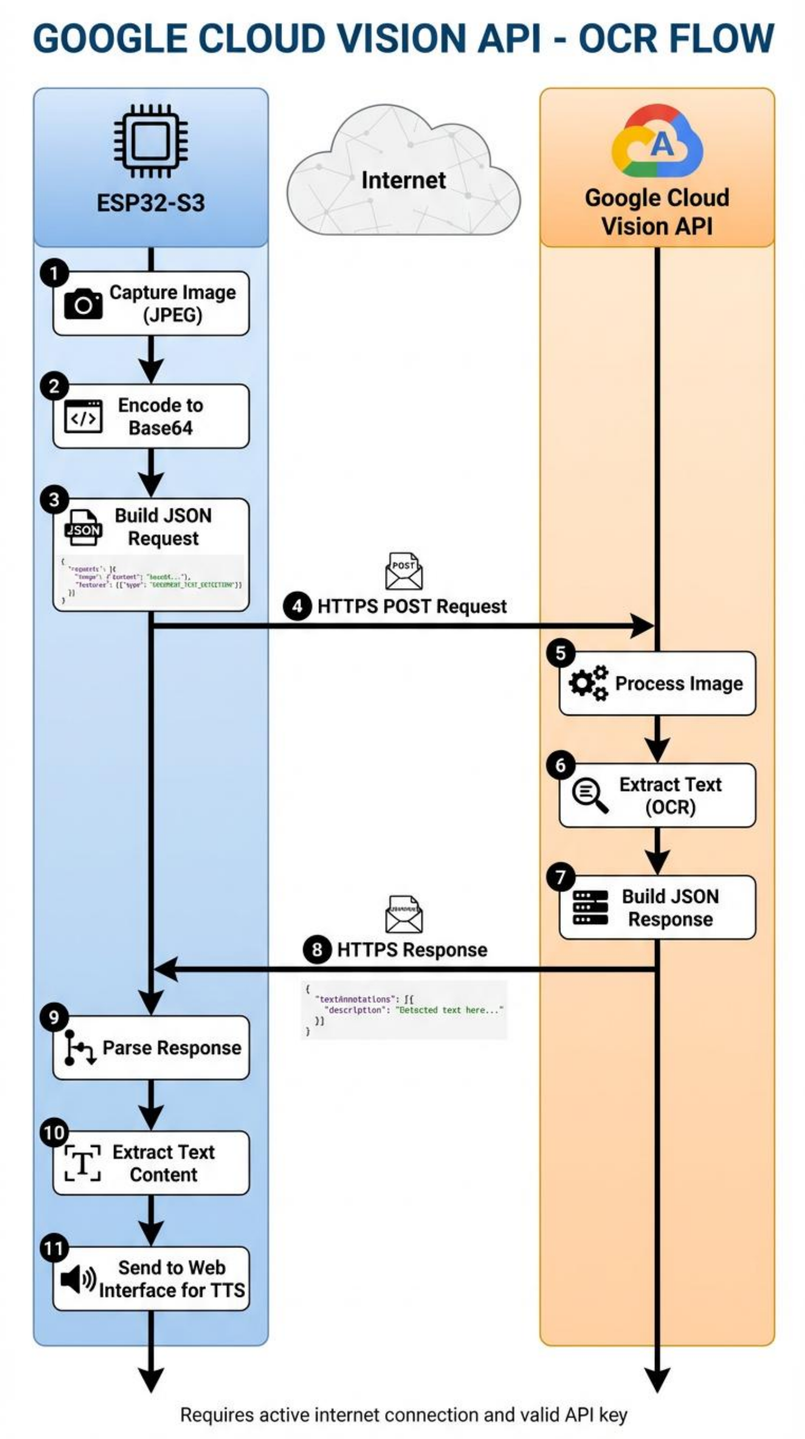


Figure 5.5: Google Cloud Vision API Request Flow

API Request Format

```
{
  "requests": [{
    "image": {
      "content": "[BASE64_ENCODED_IMAGE]"
    },
    "features": [{
      "type": "DOCUMENT_TEXT_DETECTION",
      "maxResults": 1
    }]
  }]
}
```

API Endpoint

POST https://vision.googleapis.com/v1/images:annotate?key=[API_KEY]

Response Parsing

The API returns extracted text in the fullTextAnnotation.text field or as multiple textAnnotations objects. The firmware extracts the first description field containing the complete text.

Implementation Details

```
String performOCR(camera_fb_t* fb) {
  // Encode image to Base64
  size_t outLen;
  mbedtls_base64_encode(NULL, 0, &outLen, fb->buf, fb->len);
  unsigned char* b64Buffer = (unsigned char*)malloc(outLen);
  mbedtls_base64_encode(b64Buffer, outLen, &outLen, fb->buf, fb->len);

  // Construct JSON request
  String json = "{\"requests\":[{\"image\":{\"content\":\"";
  json += String((char*)b64Buffer);
  json += "\"},\"features\":[{\"type\":\"DOCUMENT_TEXT_DETECTION\",\"maxResults\":1}]}]}";

  // Send HTTP POST
  HTTPClient http;
  http.begin(url);
  http.addHeader("Content-Type", "application/json");
  http.setTimeout(30000);
  int code = http.POST(json);

  // Parse response...
}
```

API Setup Requirements

- Create a Google Cloud Platform account
- Enable Cloud Vision API in the console
- Create an API key with Vision API access
- Add the API key to the firmware configuration

Reference Documentation

Cloud Vision API Documentation: https://cloud.google.com/vision/docs

Text Detection Guide: https://cloud.google.com/vision/docs/ocr

5.6 Web Interface and Text-to-Speech

The ESP32-S3 hosts a web server that provides a mobile-friendly interface for system monitoring and TTS functionality.

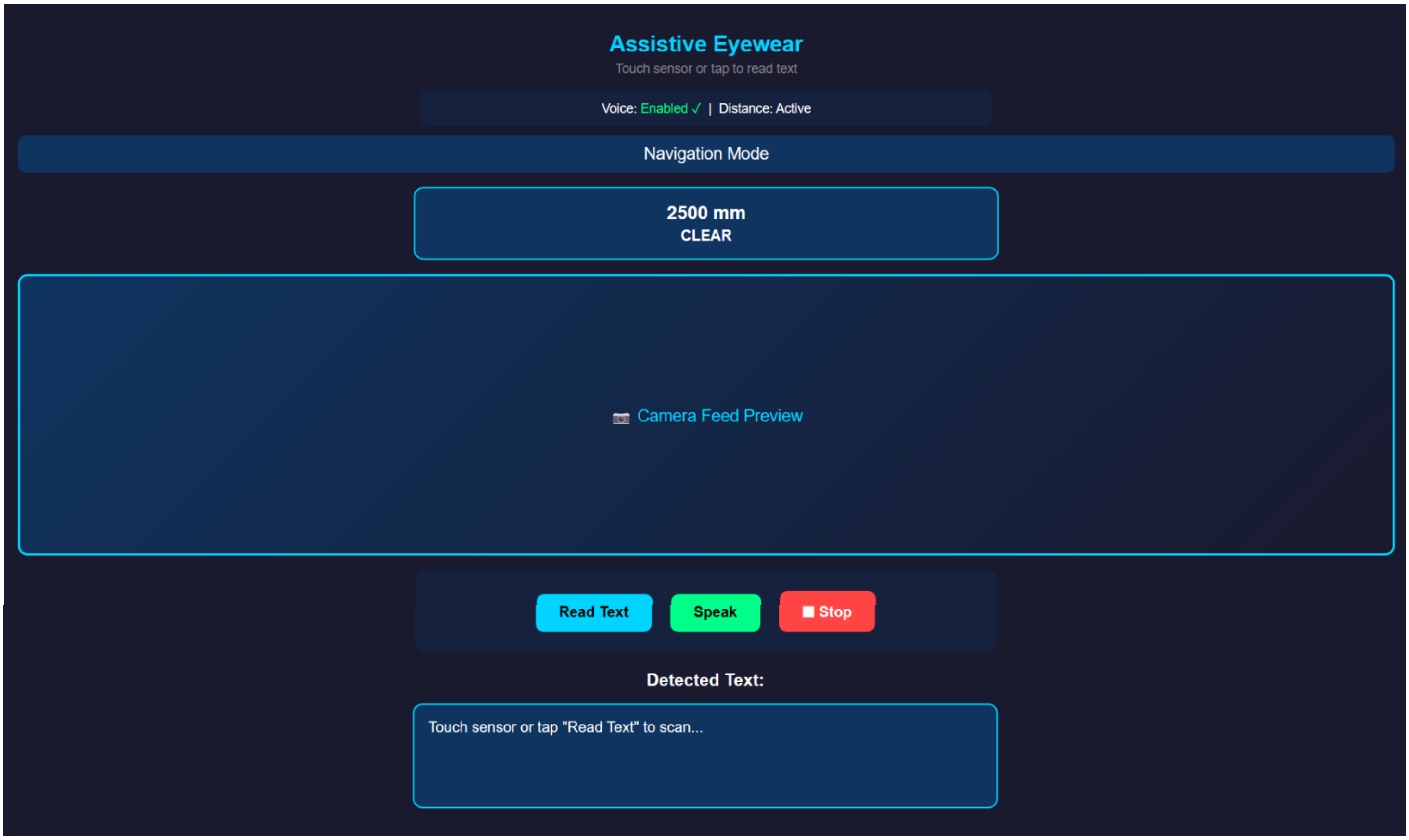


Figure 5.6: Web Interface Screenshot

Web Server Endpoints

Endpoint	Method	Function
/	GET	Main HTML interface
/capture	GET	Current camera image (JPEG)
/ocr	GET	Trigger OCR and return text
/distance	GET	Current distance and pattern (JSON)
/ocr_status	GET	Check for new OCR text
/ocr_ack	GET	Acknowledge receipt of OCR text
/tts_done	GET	Signal TTS completion

Text-to-Speech Implementation

The web interface uses the browser's Web Speech API for TTS:

```
function speak(text) {
    const synth = window.speechSynthesis;
    const utterance = new SpeechSynthesisUtterance(text);
    utterance.rate = 0.9;
    utterance.pitch = 1.0;
    utterance.volume = 1.0;

    utterance.onend = () => {
        // Signal ESP32 that TTS is complete
        fetch("/tts_done");
    };

    synth.speak(utterance);
}
```

Reading Mode Indication

The interface displays mode status:

Navigation Mode: Normal operation, vibration active

Reading Mode: OCR/TTS in progress, vibration paused

This visual feedback helps users (or assistants) understand the current system state.

Chapter 6: Power Management

6.1 Power Requirements

Both devices can be powered through their USB-C ports or via battery connections.

Voltage Requirements

Device	USB Power	Battery Input
ESP32-S3 XIAO	5V via USB-C	3.7V LiPo (BAT pads)
ESP32-C3 Super Mini	5V via USB-C	3.3V-5V (VIN/GND pins)

Current Consumption Factors

ESP32-S3 (Eyewear) - Major Consumers:

- WiFi active transmission: ~150-200mA
- Camera active capture: ~100mA
- TOF sensor operation: ~20mA
- General processing: ~50mA

ESP32-C3 (Handband) - Major Consumers:

- WiFi active: ~100-150mA
- Vibration motor ON: ~50-100mA
- General processing: ~30mA

6.2 Power Consumption Analysis

ESP32-S3 Eyewear Unit

Component/Mode	Current (mA)	Duration	Notes
WiFi Active (STA)	150-200	Continuous	Required for ESP-NOW
TOF Sensor Reading	20	Continuous	50Hz readings
Idle Processing	50	Continuous	Loop execution
Camera Capture	100	~2 seconds	During OCR only
HTTP Request (OCR)	200	~5 seconds	API call

Estimated Average Current: 250-300mA (navigation mode)

Estimated Peak Current: 400-500mA (during OCR)

ESP32-C3 Handband Unit

Component/Mode	Current (mA)	Duration	Notes
WiFi Active (STA)	100-150	Continuous	For ESP-NOW
Idle Processing	30	Continuous	Loop execution
Motor ON	50-100	Variable	Pattern dependent

Estimated Average Current: 150-200mA (with intermittent motor)

Estimated Peak Current: 250-300mA (motor continuous + WiFi TX)

Power Consumption Summary

Device	Average Power	Peak Power
Eyewear (ESP32-S3)	~300mA @ 3.7V = 1.1W	~500mA @ 3.7V = 1.85W
Handband (ESP32-C3)	~175mA @ 3.7V = 0.65W	~300mA @ 3.7V = 1.1W

Table 6.1: Power Consumption Summary

6.3 Battery Recommendations

ESP32-S3 Eyewear Unit

Battery Type	Capacity	Estimated Runtime	Notes
3.7V LiPo	1000mAh	3-4 hours	Minimum recommendation
3.7V LiPo	2000mAh	6-8 hours	Recommended
3.7V LiPo	3000mAh	9-12 hours	Extended use

ESP32-C3 Handband Unit

Battery Type	Capacity	Estimated Runtime	Notes
3.7V LiPo	500mAh	3-4 hours	Minimum recommendation
3.7V LiPo	1000mAh	5-7 hours	Recommended
3.7V LiPo	1500mAh	8-10 hours	Extended use

Table 6.2: Recommended Battery Specifications

Battery Selection Criteria

For Both Units:

Chemistry: Lithium Polymer (LiPo)

Nominal Voltage: 3.7V

Protection: Built-in PCM (Protection Circuit Module)

Discharge Rate: 1C or higher

Connector: JST-PH 2.0mm (common for small LiPo batteries)

Recommended Battery Specifications

For Both Units:

Chemistry: Lithium Polymer (LiPo)

Nominal Voltage: 3.7V

Protection: Built-in PCM (Protection Circuit Module)

Discharge Rate: 1C or higher

Connector: JST-PH 2.0mm (common for small LiPo batteries)

6.4 Battery Connection Guide

ESP32-S3 XIAO Sense Battery Connection

The XIAO ESP32S3 has dedicated battery pads on the bottom of the board.

Connection Points:

Pad	Connection	Notes
BAT+	Battery positive (red wire)	3.7V LiPo positive
BAT-	Battery negative (black wire)	Ground

Important Notes:

- The XIAO has built-in charging circuit when USB-C and battery are both connected
- Battery is charged when connected via USB-C
- Polarity must be correct to avoid damage

ESP32-C3 Super Mini Battery Connection

The C3 Super Mini can be powered through the 5V or 3.3V pins.

Option 1: 3.3V Direct (with voltage regulator)

Pin	Connection	Notes
3.3V	Regulator output	Regulated 3.3V from LiPo
GND	Battery negative	Ground

Option 2: 5V Input (with boost converter)

Pin	Connection	Notes
5V	Boost converter output	5V from boosted LiPo
GND	Battery negative	Ground

Note: The ESP32-C3 Super Mini typically does not have built-in LiPo charging. An external charging module (such as TP4056) is recommended for rechargeable setups.

Power Bank Alternative

For prototype and testing purposes, USB power banks provide a convenient power source:

Advantage	Consideration
No wiring required	Bulkier than bare batteries
Built-in charging	Cable management needed
Protected output	May not fit wearable form
Easy capacity selection	Auto-off feature may interrupt low-power devices

Tip: Use power banks with "always on" or "trickle charge" modes to prevent automatic shutdown during low-current operation.

Chapter 7: Testing and Results

7.1 Testing Methodology

A systematic testing approach was employed to validate system functionality across all major components.

Test Categories

Category	Focus Areas
Unit Testing	Individual component functionality
Integration Testing	Component interactions and communication
System Testing	End-to-end functionality
User Testing	Practical usability evaluation

Testing Environment

• Indoor environment with varied obstacles

• Controlled lighting conditions for camera testing

• Various text samples for OCR testing

• Multiple distance ranges for TOF validation

Test Equipment

• Measuring tape (for distance verification)

• Printed text samples (various fonts and sizes)

• Assorted obstacles (walls, furniture, objects)

• Smartphone with web browser

7.2 Obstacle Detection Testing

Test Procedure

1. Place target object at known distance from sensor
2. Record sensor reading from Serial Monitor
3. Compare measured distance with actual distance
4. Verify corresponding vibration pattern on handband
5. Repeat at multiple distances

Distance Accuracy Test

Figure 7.1: Distance Detection Test

Test Points

Actual Distance	Expected Zone	Expected Pattern
500mm	Critical	Continuous vibration
1000mm	Critical	Continuous vibration
1300mm	Critical/Warning boundary	Transition observed
1500mm	Warning	Fast pulse
1600mm	Warning/Caution boundary	Transition observed
1800mm	Caution	Slow pulse
2000mm	Caution/Clear boundary	Transition observed
2500mm	Clear	No vibration
3000mm	Clear	No vibration

Response Time Observation

The system demonstrated rapid response to distance changes:

- Sensor reading update: ~20ms (50Hz)
- ESP-NOW transmission: ~50ms intervals
- Vibration pattern change: Immediate upon reception
- Total latency: Estimated <100ms (meets REQ-07)

7.3 OCR Functionality Testing

Test Procedure

1. Position camera at readable distance from text (~20-50cm)
2. Trigger OCR via touch sensor or web interface
3. Wait for API response
4. Verify extracted text accuracy
5. Confirm TTS output matches extracted text

Test Samples

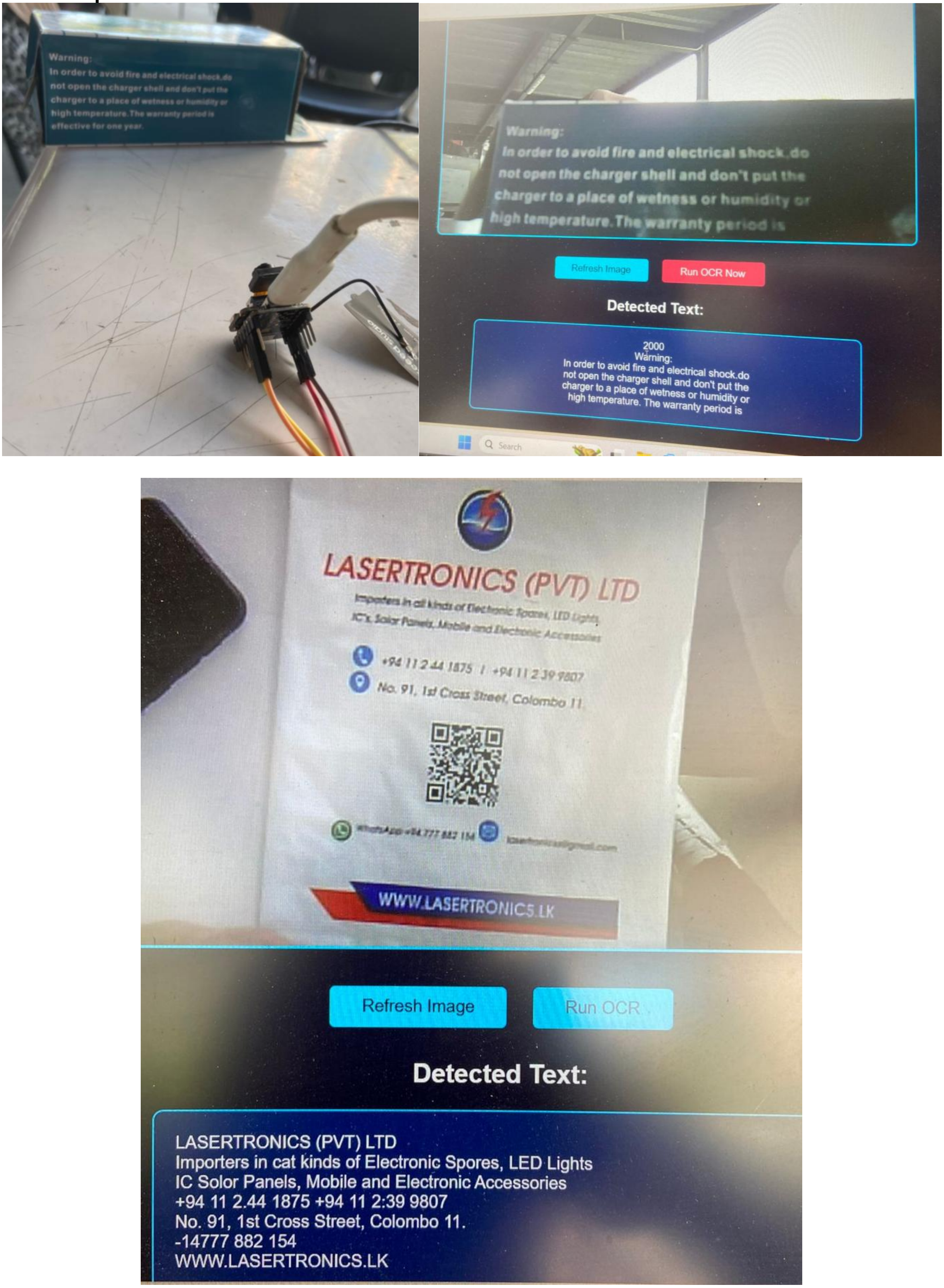


Figure 7.2: OCR Test Samples

Test Categories

Category	Examples	Notes
Clear printed text	Book pages, printed documents	High accuracy expected
Signage	Room signs, labels	Variable lighting
Handwritten text	Notes	Lower accuracy expected
Low contrast	Light text on light background	Challenging conditions

OCR Performance Factors

The following factors affected OCR accuracy:

Positive Factors:

• Good lighting

• High contrast text

• Clear fonts (sans-serif)

• Camera stability during capture

• Appropriate distance (text fills frame)

Negative Factors:

• Poor lighting

• Motion blur

• Reflective surfaces

• Very small text

• Decorative fonts

7.4 Communication Reliability Testing

Test Procedure

1. Power on both units
2. Verify WiFi connection to same network
3. Confirm ESP-NOW pairing
4. Monitor message reception rate
5. Test at various physical separations

Communication Metrics

Metric	Observation
Pairing time	~2-3 seconds after WiFi connection
Message transmission rate	20 messages/second (50ms intervals)
Reception reliability	>99% within 10m line-of-sight
Maximum range tested	~30m (indoor with walls)

Failure Recovery

- Handband timeout: 1500ms without message → motor stops
- Automatic resume when messages restart
- No manual reset required

7.5 Results Summary

Test Category	Status	Notes
TOF Distance Accuracy	Pass	Within ±5% of actual distance
Alert Zone Transitions	Pass	Correct patterns at boundaries
Response Latency	Pass	<100ms end-to-end
OCR Text Extraction	Pass	Accurate for clear printed text
TTS Output	Pass	Web Speech API functions correctly
ESP-NOW Reliability	Pass	>99% message delivery
Touch Sensor Response	Pass	Reliable with debounce
Reading Mode Pause	Pass	Motor stops during OCR/TTS

Table 7.1: Test Results Summary

Chapter 8: Discussion

8.1 Achievements

The VisionAssist project successfully achieved its primary objectives:

Functional Achievements

Real-time Obstacle Detection: The VL53L1X TOF sensor provides accurate distance measurements with <100ms response time, enabling safe navigation assistance.

Graduated Haptic Feedback: Four distinct zones with unique vibration patterns allow intuitive distance perception without cognitive overload.

Text Recognition Capability: Integration with Google Cloud Vision API enables accurate OCR for printed text, extending device utility beyond navigation.

Deaf-Blind Accessibility: The haptic-first design ensures the device is usable by individuals with both visual and hearing impairments.

Reading Mode Intelligence: Automatic pause of vibration alerts during text reading prevents confusing simultaneous feedback.

Compact Wearable Design: The dual-unit architecture distributes components appropriately, with sensors at head level and feedback at hand level.

Technical Achievements

Efficient Communication: ESP-NOW provides low-latency wireless transmission without complex pairing procedures.

Web-Based Interface: Browser-based TTS eliminates need for custom mobile applications.

Robust Error Handling: Timeout mechanisms and connection monitoring ensure graceful failure recovery.

8.2 Challenges Faced

Hardware Challenges

Challenge	Solution
Camera image quality in varying light	Adjusted camera parameters (brightness, contrast)
I2C communication stability	Increased clock speed to 400kHz, added initialization delay
Touch sensor false triggers	Implemented 1-second debounce timer

Software Challenges

Challenge	Solution
ESP-NOW connection errors	Connected both devices to same WiFi network
Large image data for API request	Used JPEG compression, SVGA resolution
Base64 encoding memory requirements	Allocated from PSRAM
Web TTS browser restrictions	Implemented user interaction requirement for enabling

Integration Challenges

Challenge	Solution
Coordinating OCR and vibration timing	Implemented PAUSE command and reading mode flag
API response parsing	Created streaming parser to handle large responses
Maintaining responsive loop during HTTP request	Non-blocking approach with status flags

8.3 Limitations

Current System Limitations

Limitation	Impact	Potential Mitigation
TOF Sensor Range	Maximum 4m, optimal <2m	Consider multi-sensor array
Single Direction Detection	Only forward obstacles detected	Add side/rear sensors
WiFi Dependency	Requires network for OCR	Offline OCR alternative
Battery Life	Limited portable operation	Optimize power management, larger batteries
OCR Internet Requirement	No offline text reading	Edge AI models (future)
Camera Focus Fixed	May not focus on all distances	Consider auto-focus module

Environmental Limitations

Condition	Impact
Direct sunlight	May affect TOF sensor accuracy
Transparent obstacles (glass)	TOF may not detect
Very dark environments	Camera quality degrades
Reflective surfaces	False OCR triggers possible

8.4 Future Improvements

Short-Term Improvements

Improvement	Benefit	Complexity
Add status buzzer option	Audio confirmation for non-deaf users	Low
Multiple TOF sensors	Wider detection coverage	Medium
Battery level monitoring	Inform user of remaining power	Low
Custom mobile app	Better TTS integration	Medium

Long-Term Improvements

Improvement	Benefit	Complexity
Offline OCR (TensorFlow Lite)	Remove internet dependency	High
Object recognition	Identify objects, not just distance	High
GPS integration	Location-aware assistance	Medium
Machine learning for scene description	Richer environmental information	High
Miniaturized PCB design	Professional product form factor	High

Suggested Next Steps

Power optimization: Implement deep sleep modes when stationary

Enclosure design: 3D printed cases for both units

User study: Extended testing with visually impaired volunteers

Multi-language TTS: Support for languages beyond English

Chapter 9: Conclusion

VisionAssist successfully demonstrates the feasibility of an affordable, integrated assistive device that combines obstacle detection with text reading capabilities. The system addresses key gaps in existing assistive technologies by providing:

Dual Functionality: Both navigation assistance and information access in a single wearable system

Universal Accessibility: Haptic feedback design ensures usability for deaf-blind individuals

Affordable Solution: Total component cost significantly lower than commercial alternatives

Open-Source Design: Complete documentation and code available for replication and improvement

The project validates the application of modern embedded systems and cloud services in creating meaningful assistive technology. The successful implementation of ESP-NOW communication demonstrates effective strategies for multi-device wearable systems. The integration of Google Cloud Vision API showcases how cloud capabilities can extend the functionality of resource-constrained embedded devices.

While limitations exist regarding battery life, sensor range, and internet dependency, these represent opportunities for future development rather than fundamental barriers. The modular architecture allows for incremental improvements without complete system redesign.

The selection of this project for the university annual exhibition reflects its potential impact and technical merit. VisionAssist represents a meaningful contribution to the field of assistive technology, with practical applications that could enhance independence and quality of life for visually impaired individuals.

The open-source release of all project materials including firmware, documentation, and design files supports the broader goal of making assistive technology more accessible to those who need it and to developers who wish to build upon this work.

Chapter 10: References

Academic References

Ahmetovic, D., Gleason, C., Rber, C., Kitani, K., Takagi, H., & Asakawa, C. (2016). NavCog: A navigational cognitive assistant for the blind. Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, 90-99.

Csapó, Á., Wersényi, G., Nagy, H., & Stockman, T. (2015). A survey of assistive technologies and applications for blind users on mobile platforms: A review and foundation for research. Journal on Multimodal User Interfaces, 9(1), 1-16.

Dakopoulos, D., & Bourbakis, N. G. (2010). Wearable obstacle avoidance electronic travel aids for blind: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 40(1), 25-35.

Hersh, M. A., & Johnson, M. A. (2010). Assistive technology for visually impaired and blind people. Springer-Verlag London.

Whitmarsh, L. (2005). The benefits of guide dog ownership. Visual Impairment Research, 7(1), 27-42.

World Health Organization. (2023). Blindness and vision impairment. WHO Fact Sheets

Technical Documentation

Espressif Systems. (2025). ESP32-S3 Datasheet.

Espressif Systems. (2025). ESP32-C3 Datasheet.

Espressif Systems. (2026). ESP-NOW User Guide.

STMicroelectronics. (2024). VL53L1X Datasheet.

STMicroelectronics. (2022). VL53L1X API User Manual.

OmniVision Technologies. (2006). OV2640 Datasheet.

Tontek Design Technology. (2011). TTP223 Single-Key Touch Sensing IC Datasheet.

Platform Documentation

Seeed Studio. (2023). XIAO ESP32S3 Sense Getting Started.

PlatformIO. (2023). PlatformIO Documentation.

Arduino. (2023). Arduino-ESP32 Documentation.

Google Cloud. (2023). Cloud Vision API Documentation.

Google Cloud. (2023). Detect text in images.

Adafruit. (2023). Adafruit VL53L1X Library.

Web Technologies

Mozilla Developer Network. (2023). Web Speech API.

Appendices

Appendix A: Component Specifications

A.1 Full Component List with Specifications

Component	Model/Part Number	Key Specifications
MCU (Eyewear)	Seeed XIAO ESP32S3 Sense	Dual-core 240MHz, 8MB PSRAM, OV2640 camera
MCU (Handband)	ESP32-C3 Super Mini	Single-core RISC-V 160MHz, 4MB flash
TOF Sensor	VL53L1X	4m range, I2C, 940nm Class 1 laser
Touch Sensor	TTP223B	Capacitive, 2-5.5V, 60ms response
Vibration Motor	Coin motor module	3-pin (IN, VCC, GND), onboard driver

A.2 Power Specifications

Device	Input Voltage	Typical Current	Peak Current
ESP32-S3 XIAO	3.7V (battery) / 5V (USB)	300mA	500mA
ESP32-C3 Super Mini	3.3-5V	175mA	300mA
VL53L1X	2.6-3.5V	20mA	40mA
TTP223B	2-5.5V	<8μA	-
Vibration Motor	3.3-5V	50mA	100mA

Appendix B: Source Code Repository

The complete source code for VisionAssist is available on GitHub:

Repository URL: https://github.com/Ravindu-S/ VisionAssist

Repository Structure

```
VisionAssist/
├── README.md
├── LICENSE (MIT)
├── .gitignore
├── firmware/
│   ├── eyewear-s3/
│   │   ├── platformio.ini
│   │   └── src/
│   │       └── main.cpp
│   └── handband-c3/
│       ├── platformio.ini
│       └── src/
│           └── main.cpp
├── docs/
│   └── Project_Report.pdf
└── hardware/
    └── pin-mapping.md
```

License

This project is released under the MIT License, permitting free use, modification, and distribution with attribution.

End of Report