IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 28, Issue 5, Ser. III (Sep. – Oct. 2026), PP 01-16
www.iosrjournals.org

Design and Performance Evaluation of a Low-Cost Offline Multilingual IoT-Enabled Smart Cane for Speech-Based Assistive Navigation

[a]Nworu, Ursula Amaka, [a]Igwe, Joseph Sunday and [b]Chizoba Ezeaku-Ezeme
[a]Computer Science Department, Ebonyi State University, Abakaliki
[b]Computer Science Department, Caritas University, Enugu
Corresponding Author: ursula.nworu@ebsu.edu.ng

Abstract
Independent mobility remains a major challenge for persons with visual impairment, particularly in low-resource and linguistically diverse environments where sophisticated assistive technologies and reliable internet connectivity may be limited. This study presents the design and performance evaluation of a low-cost, offline-capable multilingual IoT-enabled smart cane for assistive navigation. The prototype integrates an ESP32 DevKit V1 microcontroller, three HC-SR04 ultrasonic sensors, a DFPlayer Mini audio module, a microSD-based multilingual audio repository, rechargeable batteries, and Blynk wireless communication. The sensors monitor the left, centre, and right regions of the walking path, while a deterministic rule-based algorithm converts measured distances into directional navigation instructions. Pre-recorded audio messages in English, Igbo, and Ezza dialect provide speech feedback without continuous internet connectivity. Blynk serves as a supplementary IoT platform for language selection, system monitoring, sensor visualization, and caregiver notifications. Unit, integration, and system-level tests were conducted under controlled laboratory conditions. Obstacles were successfully detected at 30, 50, and 80 cm, while distances of 100 and 150 cm were classified as clear, confirming the 80-cm danger threshold. All tested obstacle configurations generated the expected navigation responses, and all 27 multilingual audio files were successfully played. Language switching among English, Igbo, and Ezza was also successfully demonstrated. Ten response-time trials ranged from 115 to 141 ms, with a mean of 126 ms, below the 500-ms requirement. Offline testing confirmed continued operation of core obstacle detection and audio feedback without Wi-Fi, whereas Blynk-dependent functions required connectivity. The findings demonstrate the feasibility of a low-cost, multilingual, offline-capable smart cane for assistive navigation.
Keywords: *Smart cane; Assistive navigation; Multilingual speech; Natural Language Processing; Internet of Things (IoT).*

Date of Submission: 15-09-2026 Date of Acceptance: 29-09-2026

I. Introduction

Independent mobility is a fundamental component of human autonomy, social participation, education, employment, and quality of life. For persons with visual impairment, however, safe movement through indoor and outdoor environments remains a persistent challenge because environmental information that is normally acquired visually must be obtained through alternative sensory and technological channels. The scale of this challenge is substantial. The World Health Organization (WHO) estimates that at least 2.2 billion people globally live with near or distance vision impairment, with at least 1 billion cases preventable or yet to be addressed. Importantly, approximately 90% of people with unaddressed vision impairment or blindness live in low- and middle-income countries, where access to specialized healthcare, assistive technologies, digital infrastructure, and affordable mobility devices is often constrained (WHO, 2023, 2026).

The conventional white cane remains the most widely recognized mobility aid for visually impaired persons because of its simplicity, low cost, portability, and ability to provide immediate tactile information about nearby ground-level obstacles. Nevertheless, its sensing capability is fundamentally limited by the physical reach of the cane and by the user's ability to interpret tactile interactions with the surrounding environment. It provides little or no information about obstacles beyond its immediate reach, overhead hazards, the identity or nature of obstacles, or the direction in which a safer path may lie. Consequently, although the white cane remains indispensable, contemporary assistive-technology research has increasingly focused on augmenting rather than replacing its fundamental mobility function through electronic sensing, intelligent processing, wireless communication, and multimodal feedback (Anagnostopoulos et al., 2022; Bauer et al., 2025).

This evolution has resulted in the emergence of smart canes and electronic travel aids that integrate technologies such as ultrasonic and infrared sensors, cameras, LiDAR, inertial sensors, GPS, embedded processors, wireless communication, vibration motors, and speech interfaces. Recent reviews demonstrate that the field has progressed from relatively simple obstacle-detection devices toward sensor-rich systems capable of environmental perception, object recognition, localization, context-aware guidance, and, increasingly, artificial-intelligence-assisted navigation (Adegboye et al., 2023; Wazirali et al., 2025). Recent systems have demonstrated the feasibility of combining ultrasonic sensing with GPS, computer vision, machine learning, and multimodal feedback to improve mobility and situational awareness. For example, contemporary low-cost smart-cane research has explored simultaneous obstacle detection and pathway-surface recognition, while other systems have incorporated IoT communication, emergency functions, and remote monitoring (Benjamin et al., 2017; Okolo et al., 2025; Wazirali et al., 2025).

Despite these advances, technological sophistication does not necessarily translate into accessibility or suitability for low-resource users. High-end systems frequently depend on computationally intensive cameras, LiDAR, cloud services, smartphones, or continuous network connectivity. Such requirements can increase device cost, energy consumption, system complexity, maintenance requirements, and dependence on external digital infrastructure. Recent reviews of smart-cane technologies identify sensor limitations, computational requirements, power consumption, ergonomics, false alarms, cognitive workload, and the need for robust human-centred design as continuing challenges (Clark-Carter et al., 2023; Okpor, 2023; Wazirali et al., 2025). Thus, an important research question is not simply whether a smart cane can provide more information than a conventional cane, but whether it can provide the right information, at the right time, in an understandable form, and at a cost and level of technological complexity appropriate for its intended users.

The communication modality through which navigation information is delivered is particularly important. Haptic feedback can effectively communicate the presence and approximate direction of an obstacle, but users must learn and interpret vibration patterns. Speech, in contrast, can provide explicit and semantically meaningful information, such as “obstacle ahead,” “move left,” or “move right,” without requiring the user to translate an abstract vibration pattern into an action. The integration of speech technologies into assistive navigation therefore provides an important pathway toward more intuitive human–device interaction. Recent assistive-navigation research has increasingly combined voice output with environmental sensing and tactile feedback to provide users with richer information about their surroundings (Chen et al., 2023; Abidi et al., 2024., Okolo et al., 2025).

However, speech-based accessibility introduces another important dimension: language. A navigation system that communicates only in a dominant international language may remain technologically advanced while being practically inaccessible to users who are more proficient or comfortable in another language. This challenge is particularly relevant in linguistically diverse societies such as Nigeria, where English coexists with numerous indigenous languages and dialects. Consequently, linguistic accessibility should be considered an integral component of assistive-technology design rather than an optional feature added after the hardware has been developed.

The problem is even more pronounced for African low-resource languages. Recent natural language processing research shows that African languages remain substantially underrepresented in speech and language technologies because of limited datasets, linguistic diversity, acoustic variability, computational constraints, and insufficient commercial investment. Imam et al. (2025) identify data scarcity, linguistic complexity, computational limitations, acoustic variability, bias, and privacy as major challenges affecting automatic speech recognition for African low-resource languages. Similarly, recent work on multilingual African NLP demonstrates considerable progress in developing language technologies across hundreds of African languages, but also highlights the continuing disparity between widely supported global languages and underrepresented African languages (Ezeani et al., 2020; Adebara et al., 2024).

Recent advances in low-resource speech synthesis provide grounds for optimism. Katumba (2025), for example, demonstrated that crowd sourced speech recordings can help reduce the dependence of text-to-speech development on expensive single-speaker datasets. Other recent research has demonstrated that synthetic speech data can substantially expand the resources available for African-language automatic speech recognition, illustrating the accelerating development of computational tools for linguistically underserved communities (Rahman et al., 2024; DeRenzi et al., 2025). Nevertheless, the existence of emerging language models does not automatically solve the deployment problem. Assistive mobility devices operate under strict constraints of processing power, memory, battery capacity, latency, reliability, and physical size. For safety-critical navigation, dependence on remote servers or continuous internet connectivity may introduce delays or complete loss of functionality when connectivity becomes unavailable.

This consideration motivates the increasing importance of offline and edge-oriented assistive intelligence**.** Processing critical environmental information locally can reduce communication latency, decrease dependence on cloud services, preserve functionality during network interruptions, and potentially improve privacy. For low-resource environments, offline operation is particularly valuable because mobile internet access may be intermittent, expensive, or unavailable. Accordingly, a practical smart cane for deployment in such settings should maintain its core navigation functions independently of internet connectivity while using wireless communication only for supplementary services such as configuration, monitoring, emergency communication, or caregiver support (Fiorini et al., 2022; Abidi et al., 2024).

The Internet of Things (IoT) provides an additional opportunity to achieve this balance. IoT-enabled assistive systems can connect embedded sensing devices with smartphones or remote monitoring platforms, allowing environmental events, system status, user preferences, and selected safety information to be communicated beyond the physical device. Recent smart-cane research has demonstrated the feasibility of integrating connected services with environmental sensing and safety functions, although connectivity, privacy, power consumption, and system complexity remain important design considerations (Khan et al., 2022; Abidi et al., 2024., Kathiria et al., 2024., Wazirali et al., 2025; Tulluri et al., 2025). A carefully designed architecture can therefore maintain local, real-time obstacle detection and speech feedback while reserving IoT connectivity for non-critical functions such as language configuration and caregiver monitoring.

Against this background, there remains a significant opportunity to develop a smart cane that prioritizes affordability, offline reliability, multilingual accessibility, directional obstacle detection, and optional IoT connectivity within a single practical platform. Rather than attempting to replicate expensive camera- or LiDAR-intensive systems, a low-cost architecture based on multiple ultrasonic sensors can provide a pragmatic approach to near-field spatial awareness. Three-directional sensing can extend obstacle awareness beyond a single forward-looking detection zone by separately monitoring the left, centre, and right regions of the user's path. When combined with appropriately structured voice feedback, such sensing can transform distance information into actionable navigation cues.

The present study therefore proposes the design and performance evaluation of a low-cost offline multilingual IoT-enabled smart cane using Natural Language Processing for assistive navigation**.** The system is built around an ESP32 microcontroller, three HC-SR04 ultrasonic sensors, a DFPlayer Mini audio module, an SD card-based audio repository, rechargeable battery power, and Blynk-enabled wireless communication. The three ultrasonic sensors provide directional obstacle information, while locally stored multilingual audio messages provide navigation feedback without requiring continuous internet access. English, Igbo, and Ezza dialect are incorporated to address linguistic inclusivity within the Nigerian context. The IoT component provides a complementary communication channel for language configuration and caregiver-oriented monitoring rather than making internet connectivity a prerequisite for basic obstacle detection and user feedback.

The study consequently addresses four interconnected requirements that remain insufficiently integrated in many existing assistive-navigation systems: (i) real-time directional obstacle detection, (ii) multilingual and culturally relevant speech feedback, (iii) reliable offline operation, and (iv) low-cost IoT-enabled remote support**.** This combination is important because affordability without usability limits adoption; multilingual functionality without reliable sensing limits safety; sophisticated sensing without offline capability limits deployment in poorly connected environments; and connectivity without local autonomy creates a vulnerability when the network becomes unavailable. The proposed architecture seeks to balance these competing requirements through a lightweight embedded design that preserves essential navigation functions locally while using IoT connectivity selectively.

Accordingly, the research contributes not merely another ultrasonic smart-cane prototype, but a context-oriented assistive-navigation architecture designed around the realities of linguistically diverse and resource-constrained environments. Its performance is evaluated in terms of obstacle-detection accuracy, response time, multilingual audio performance, language-switching efficiency, operational reliability, and offline functionality. By examining these parameters collectively, the study provides empirical evidence on whether a low-cost embedded platform can deliver dependable multilingual navigation assistance while maintaining the affordability, portability, and independence expected of a practical mobility aid. The work thereby contributes to the broader development of inclusive assistive technologies in which sensing, embedded intelligence, speech interaction, multilingual accessibility, and IoT connectivity are integrated according to the actual infrastructural and linguistic conditions of their intended users.

System design and implementation

System Design

The proposed system was designed as a low-cost, embedded, offline-capable smart cane integrating directional obstacle sensing, multilingual speech feedback, and IoT-enabled remote monitoring. The design followed a modular architecture in which sensing, decision-making, audio playback, language management, communication, power management, and system initialization were developed as independent but interconnected functional modules. This approach facilitates component-level testing, simplifies maintenance, and permits future expansion of the system.

The overall architecture consists of four principal functional layers: perception, processing, output, and communication**.** The perception layer acquires environmental information using three HC-SR04 ultrasonic sensors positioned to monitor the left, centre, and right regions of the user's walking path. The processing layer, implemented on an ESP32 DevKit V1 microcontroller, converts the sensor measurements into distance values and applies predefined decision rules to determine the presence, direction, and configuration of obstacles. The output layer uses a DFPlayer Mini and locally stored audio files to provide spoken navigation instructions. The communication layer uses the ESP32's Wi-Fi capability and the Blynk IoT platform to support language selection, real-time system-status visualization, sensor-data monitoring, and caregiver notifications.

The principal hardware components comprise an ESP32 DevKit V1, three HC-SR04 ultrasonic sensors, a DFPlayer Mini MP3 module, microSD card, 3-W/8-Ω speaker, rechargeable 18650 battery pack, DC-DC buck converter, TP4056 charging module, and protective electronics enclosure. The ESP32 provides the central processing and communication functions, while the DFPlayer Mini enables local playback of pre-recorded multilingual navigation messages. The microSD card stores the audio library, allowing the core navigation function to remain operational without internet access. The system supports English, Igbo, and Ezza dialect, with the architecture allowing additional languages to be incorporated by extending the audio repository and corresponding language mapping.

Hardware Configuration and Physical Integration

The three ultrasonic sensors constitute the primary environmental sensing subsystem. They are mounted at different heights and angular orientations to provide directional coverage of the user's immediate walking environment. The left sensor is positioned approximately 50 cm above the bottom of the cane and oriented approximately 45° to the left, the centre sensor is positioned at approximately 60 cm and directed forward, while the right sensor is positioned at approximately 75 cm and oriented approximately 45° to the right. This arrangement provides spatially distributed sensing rather than relying on a single forward-facing detection zone.

The ESP32, DFPlayer Mini, power-conversion circuitry, and battery pack are housed within an enclosure mounted near the middle section of the cane. The speaker is positioned near the handle and directed toward the user to improve audibility. The electrical system uses two rechargeable 18650 cells as the primary power source, with a DC-DC buck converter providing regulated voltage to the electronic components. A TP4056 module provides battery charging and protection. The detailed electrical connections and component placement are presented in Tables 4.3 and 4.4 and Figures 4.1 and 4.2.

The ultrasonic sensors interface with the ESP32 through dedicated trigger and echo GPIO connections. The DFPlayer Mini communicates with the ESP32 through UART, while a 1-kΩ resistor is incorporated on the DFPlayer RX line for appropriate signal interfacing. The complete pin configuration is provided in Table 4.4.

Input and Output Specification

The system receives two principal categories of input: environmental measurements from the three ultrasonic sensors and user configuration commands from the Blynk mobile application. The ultrasonic sensors provide left, centre, and right distance measurements, whereas the Blynk interface provides language-selection input. System outputs are delivered through two complementary channels. The primary user-facing output is multilingual speech generated through locally stored audio files and played by the DFPlayer Mini. The secondary output is the Blynk dashboard, which presents obstacle status, sensor readings, and navigation direction and provides caregiver-oriented notifications.

Table 1.1: Input/Output Specification

Component	Type	Description	Voltage/Logic
Left Ultrasonic Sensor	Input	Detects obstacles on the left (45° angle)	5V, Digital (TRIG/ECHO)
Center Ultrasonic Sensor	Input	Detects obstacles directly ahead	5V, Digital (TRIG/ECHO)
Right Ultrasonic Sensor	Input	Detects obstacles on the right (45° angle)	5V, Digital (TRIG/ECHO)
Blynk App (V1)	Input	Language selection (0=English, 1=Igbo, 2=Ezza)	Virtual Pin (Wi-Fi)
DFPlayer Mini (TX)	Output	Audio feedback via speaker/headphone	5V, UART (9600 baud)
Blynk App (V2)	Output	Obstacle status display	Virtual Pin (Wi-Fi)
Blynk App (V3)	Output	Sensor readings display	Virtual Pin (Wi-Fi)
Blynk App (V4)	Output	Direction guidance display	Virtual Pin (Wi-Fi)
Blynk Notifications	Output	Critical alerts to caregiver	Push Notification

This arrangement separates critical navigation functions from non-critical network functions. Obstacle detection and voice guidance are performed locally, allowing the cane to retain its fundamental navigation capability when internet connectivity is unavailable. Wi-Fi and Blynk connectivity are therefore used primarily for configuration, monitoring, and remote notification.

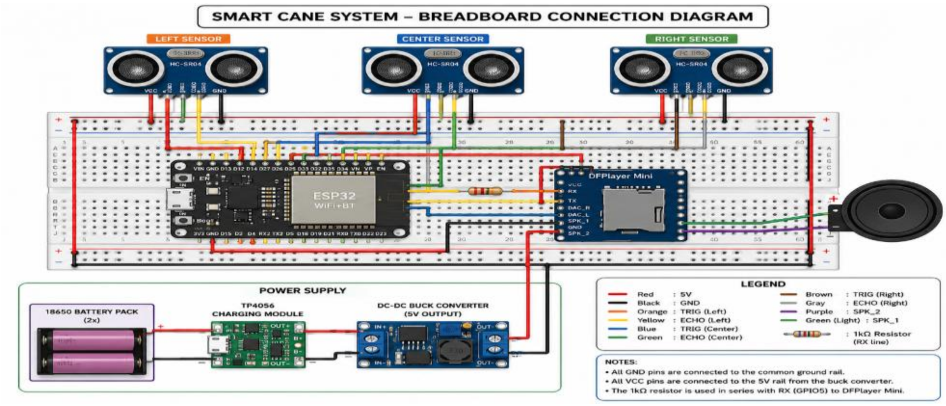

Figure 1.1: Breadboard Connection Diagram

System Processing and Decision Design

The system operates through a continuous processing pipeline consisting of four stages: perception, decision, output, and communication. During perception, the three ultrasonic sensors acquire distance measurements from the surrounding environment. The ESP32 converts the measured echo time into distance values in centimetres and compares each measurement with a predefined safety threshold.

During the decision stage, the controller classifies each sensing direction as either blocked or unblocked. The decision logic distinguishes among clear-path conditions, single-direction obstacles, and multiple-direction obstacle configurations. Where a safe alternative direction is available, the system generates an appropriate directional instruction, such as moving left, moving right, or proceeding forward. Where the centre path is blocked and one side remains available, the system prioritizes the available direction. This rule-based approach provides deterministic and computationally lightweight decision-making suitable for the resource-constrained embedded platform.

Algorithm 1: Obstacle Detection and Guidance

```
Algorithm: detectAndGuide()
Input: left.distance, center.distance, right.distance
Output: Voice message played via DFPlayer
IF (currentTime - lastPlayTime < MIN_MESSAGE_INTERVAL) THEN
    RETURN
END IF

leftBlocked ← left.distance < DANGER_DISTANCE
centerBlocked ← center.distance < DANGER_DISTANCE
rightBlocked ← right.distance < DANGER_DISTANCE

blockedCount ← (leftBlocked?1:0) + (centerBlocked?1:0) + (rightBlocked?1:0)

IF (blockedCount == 0) THEN
    messageType ← MSG_CLEAR
ELSE IF (blockedCount == 1) THEN
    IF (leftBlocked) THEN messageType ← MSG_LEFT_OBSTACLE
    ELSE IF (centerBlocked) THEN messageType ← MSG_FRONT_OBSTACLE
    ELSE IF (rightBlocked) THEN messageType ← MSG_RIGHT_OBSTACLE
    END IF
ELSE // Multiple directions blocked
    IF (centerBlocked) THEN
        IF (!leftBlocked) THEN messageType ← MSG_MOVE_LEFT
        ELSE IF (!rightBlocked) THEN messageType ← MSG_MOVE_RIGHT
        ELSE messageType ← MSG_FRONT_OBSTACLE
        END IF
    ELSE // Center is clear
        IF (leftBlocked && rightBlocked) THEN messageType ← MSG_CLEAR
        ELSE IF (leftBlocked) THEN messageType ← MSG_MOVE_FORWARD_RIGHT
        ELSE IF (rightBlocked) THEN messageType ← MSG_MOVE_FORWARD_LEFT
```

```
        END IF
    END IF
END IF

IF (messageType != lastPlayedFile) THEN
    fileNumber ← messageType + LANGUAGE_OFFSETS[currentLanguage]
    mp3.playMp3FolderTrack(fileNumber)
    lastPlayedFile ← messageType
    lastPlayTime ← currentTime
    Blynk.virtualWrite(VPIN_DIRECTION, getDirectionString(messageType))
    IF (messageType != MSG_CLEAR) THEN
        Blynk.virtualWrite(VPIN_STATUS, "⚠ " + direction)
        Blynk.logEvent("Obstacle: " + direction)
    END IF
END IF
```

To prevent excessive repetition of identical voice messages, the decision module incorporates a minimum message interval and records the last navigation message played. A new audio instruction is generated only when the navigation state changes or the prescribed message interval has elapsed. This reduces unnecessary auditory interruptions while maintaining timely warnings.

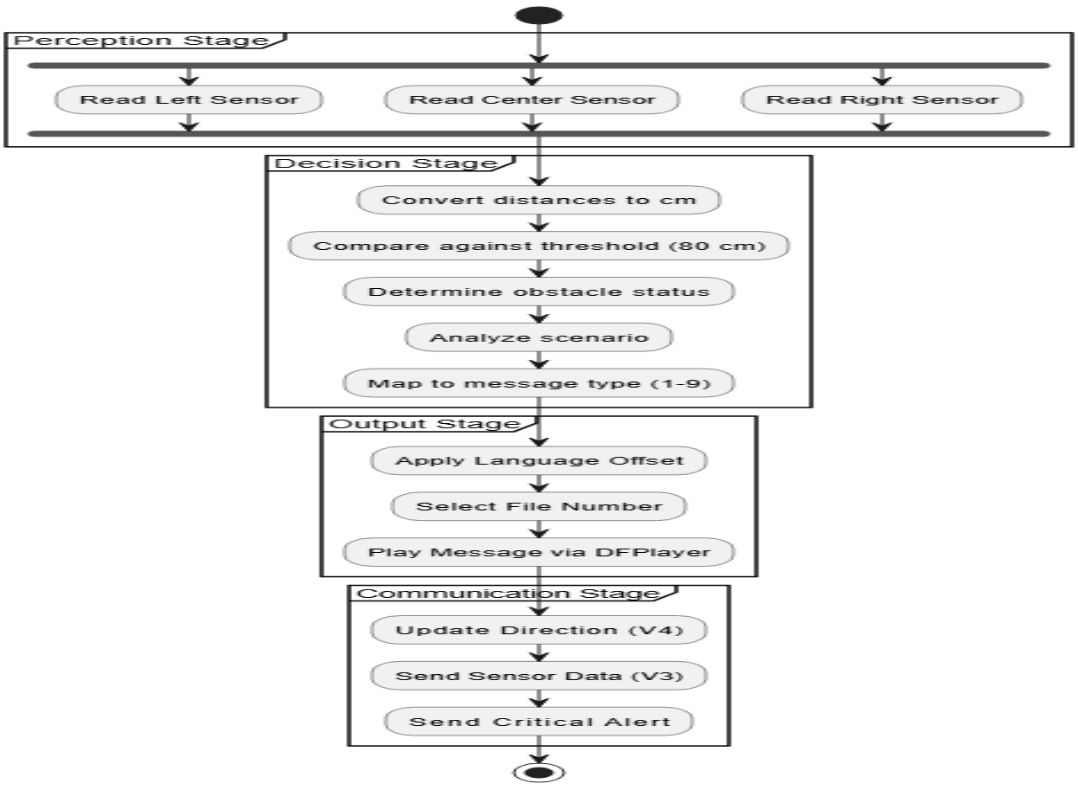

Figure 1.2: System Process Flow Diagram

The complete decision procedure is represented in Algorithm 1, while the system process flow is illustrated in Figure 1.2. The operational flowchart in Figure 1.3 shows the sequence from initialization through continuous sensing, obstacle classification, message selection, audio playback, IoT update, and repetition of the monitoring cycle.

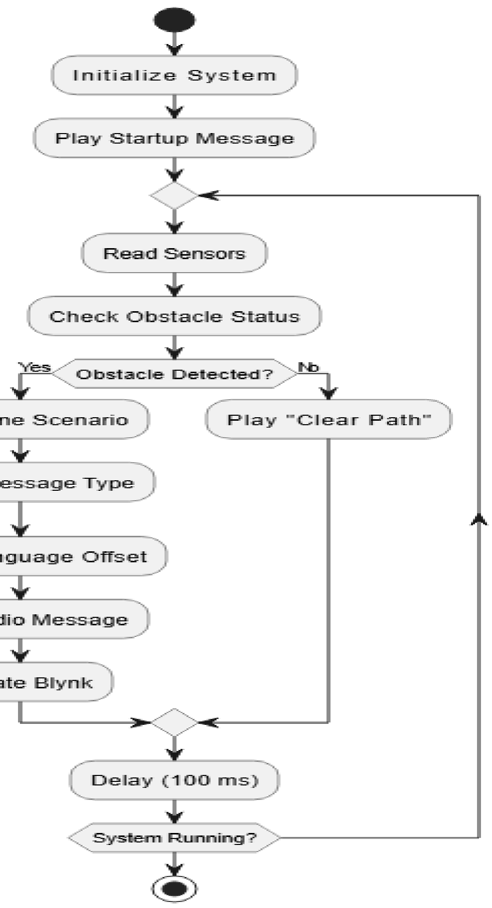

Figure 1.3: System Flowchart

Multilingual Audio and Language Management

Multilingual interaction is implemented through a local audio-library architecture rather than real-time cloud-based speech synthesis. Navigation messages are pre-recorded in English, Igbo, and Ezza dialect and stored as MP3 files on a microSD card. The language-management module uses an offset-based file-mapping strategy to associate each navigation message with its corresponding language. English messages occupy one numerical range, Igbo messages another, and Ezza messages a third. Consequently, the same navigation state can be mapped to different language-specific audio files without modifying the underlying obstacle-detection algorithm.

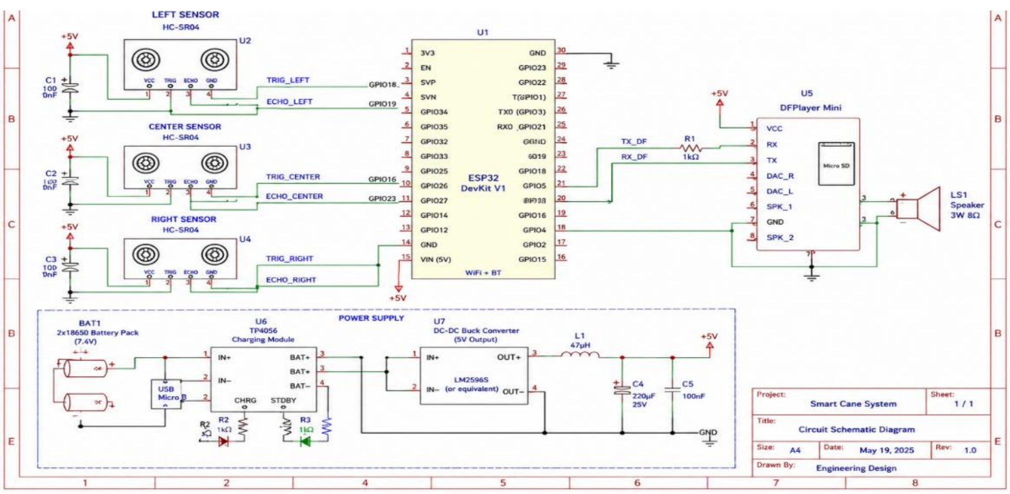

Figure 1.4: Circuit Schematic Diagram

Language selection is performed through a segmented switch in the Blynk mobile application. Once a language is selected, the corresponding language offset becomes active and subsequent navigation messages are retrieved from that language's audio range. The approach minimizes software complexity and permits additional languages to be introduced by adding appropriately organized audio files and updating the language-offset configuration.

Although the architecture supports multilingual speech interaction, the implemented prototype uses pre-recorded speech output rather than dynamic text-to-speech generation or automatic speech recognition. This distinction is important because it enables completely offline operation and consistent pronunciation while limiting the system to a predefined vocabulary of navigation messages.

IoT Communication and Remote Monitoring

The IoT subsystem is implemented using the ESP32's 2.4-GHz Wi-Fi capability and the Blynk platform. Communication between the embedded system and the mobile application is organized through virtual pins. The application provides language selection through V1, obstacle-status information through V2, sensor measurements through V3, and directional guidance through V4. Critical obstacle events can also be logged and transmitted as notifications to a designated caregiver.

Table 1.2: Implementation Highlights

Component	Implementation Approach
ESP32 Programming	Arduino IDE with C++ (sketch-based development)
Sensor Interfacing	Digital I/O with pulseIn() for distance measurement
Audio Playback	Makuna's DFMiniMp3 library for serial command control
IoT Connectivity	Blynk 2.0 with virtual pins and event logging
Language Management	Offset-based file mapping (English = 0, Igbo = 10, Ezza = 20)
Power Management	Voltage regulation using a DC-DC buck converter

The IoT layer is designed as a complementary rather than essential component of navigation. Core sensing, decision-making, and audio feedback are executed locally on the ESP32 and DFPlayer Mini. Consequently, interruption of the Wi-Fi connection does not prevent the system from detecting obstacles or issuing stored voice instructions. When connectivity is available, the Blynk platform provides additional functionality, including language configuration, sensor visualization, system-status monitoring, and caregiver awareness.

To reduce unnecessary communication overhead and power consumption, sensor and status information are transmitted according to predefined update intervals rather than continuously streaming all raw measurements to the cloud.

Software Architecture and Implementation

The embedded software was developed in C++ using the Arduino IDE with the ESP32 board-support package. Libraries supporting Blynk communication, DFPlayer Mini control, serial communication, and sensor interfacing were incorporated into the implementation. The software architecture comprises seven functional modules: Sensor Interface, Decision Logic, Audio Playback, Language Management, IoT Communication, Power Management, and System Initialization.

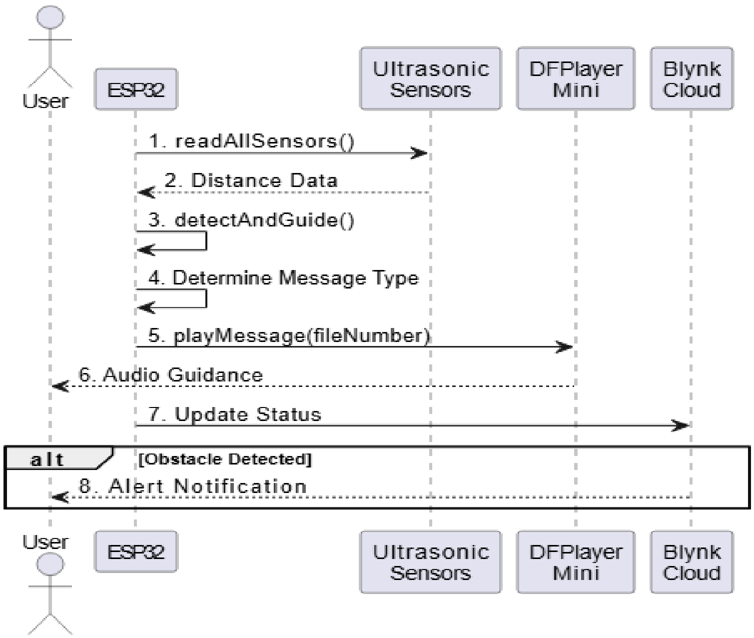

Figure 1.5: Sequence Diagram – Obstacle Detection and Guidance

The Sensor Interface Module acquires distance measurements from the three ultrasonic sensors using trigger pulses and echo-time measurement. The measured time is converted into distance and passed to the decision layer. The Decision Logic Module constitutes the principal control component. It evaluates the three directional measurements, determines obstacle status, identifies the navigation scenario, and selects the appropriate instruction. A message-suppression mechanism minimizes repetitive audio output. The Audio Playback Module communicates with the DFPlayer Mini through UART and selects the appropriate MP3 file based on the navigation state and currently selected language. The DFPlayer retrieves the file from the microSD card and produces the corresponding voice instruction through the speaker. The Language Management Module controls the mapping between navigation messages and language-specific audio files. The implemented prototype supports English, Igbo, and Ezza dialect and is structured to accommodate additional languages.

Table 1.3: Hardware Requirements

Component	Quantity	Specification	Function
ESP32 DevKit V1	1	38-pin, Dual-core, 4 MB Flash	Main microcontroller
HC-SR04 Ultrasonic Sensor	3	2–400 cm range, 5 V operation	Obstacle detection (Left, Center, Right)
DFPlayer Mini MP3 Module	1	3.2–5 V, UART control, 3 W amplifier	Audio playback
MicroSD Card	1	4–16 GB, FAT32 format	Stores audio files
Speaker	1	3 W, 8 Ω impedance	Audio output
3.5 mm Audio Jack (Optional)	1	Stereo TRS socket	Headphone connection
18650 Batteries	2	3.7 V, 2500 mAh or higher	Power source
DC-DC Buck Converter	1	5 V output, 2 A maximum	Voltage regulation
TP4056 Charging Module	1	With battery protection	Battery charging and management
Jumper Wires	1 Set	Silicone insulated	Electrical connections
Enclosure Box	1	Waterproof (IP65)	Protects electronic components
Cane Body	1	Aluminum/Carbon fiber	Walking cane structure
Resistor	1	1 kΩ	Level shifting for DFPlayer RX

The IoT Communication Module manages Wi-Fi connectivity and communication with Blynk. It updates sensor readings, obstacle status, and navigation direction and generates event notifications for critical conditions. The Power Management Module provides the electrical interface between the rechargeable battery system and the electronic components.

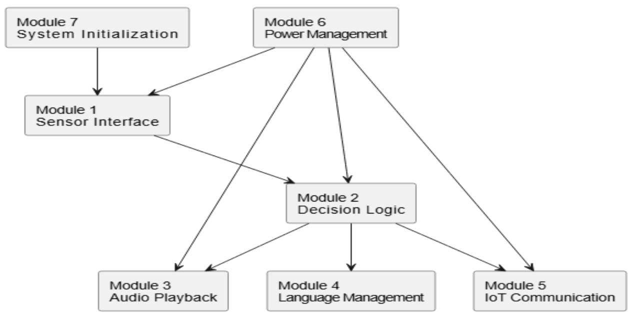

Figure 1.6: System Module Architecture

The current prototype employs regulated power delivery through the DC-DC buck converter, while the TP4056 module provides charging and battery protection. The System Initialization Module configures GPIO pins, initializes serial communication and the DFPlayer Mini, establishes Wi-Fi connectivity, initializes the Blynk service, and sets the operating parameters before the system enters its continuous monitoring loop. The modular organization improves maintainability and facilitates independent testing of individual components before complete system integration. The overall module architecture is shown in Figure 1.6.

Table 1.4: Software Requirements

Software	Version	Purpose
Arduino IDE	2.3.x or later	Development environment, code compilation, and upload
ESP32 Board Package	3.0.x or later	ESP32 board support for Arduino
Blynk Library	1.0.0 or later	IoT connectivity and virtual pins
DFMiniMp3 Library	Latest	DFPlayer Mini serial control
SoftwareSerial Library	Built-in	Serial communication with DFPlayer
Blynk IoT App	Latest	Mobile interface for monitoring and language selection
Python	3.8+	Audio file generation
gTTS Library	Latest	Text-to-speech conversion

Table 1.5: Justification for the Choice of Programming Language and Platform

Factor	Justification
Hardware Compatibility	Arduino IDE provides complete ESP32 support, including GPIO, Wi-Fi, UART, and peripheral libraries.
Library Ecosystem	Existing libraries for DFPlayer Mini, Blynk, and SoftwareSerial reduce development effort and improve reliability.
Community Support	Extensive documentation, tutorials, and community forums facilitate development and troubleshooting.
Performance	C++ produces efficient machine code suitable for the ESP32's dual-core processor and real-time operation.
Debugging Capability	The Serial Monitor enables real-time monitoring of sensor readings and system status.
Portability	Sketch-based development allows easy migration between compatible ESP32 development boards.
Cost-Effectiveness	Arduino IDE is free and open-source, eliminating software licensing costs.

System Algorithm

At startup, the ESP32 initializes the connected sensors, audio module, communication interfaces, and operating parameters. When initialization is completed, the system enters a continuous monitoring cycle. The three sensors acquire distance measurements, which are compared with the predefined danger threshold. If all monitored directions are clear, a clear-path message is issued. When an obstacle is detected, the decision algorithm determines its direction and evaluates whether an alternative path is available.

For multiple obstacles, the algorithm prioritizes the centre path and evaluates the left and right alternatives. The resulting navigation state is then mapped to the selected language using the corresponding language offset. The appropriate audio file is played through the DFPlayer Mini, while the Blynk interface is updated with the current direction and obstacle status. Critical obstacle events are also logged for remote notification. The system then waits for the programmed short processing interval before repeating the sensing and decision cycle.

The use of deterministic threshold-based rules minimizes computational requirements and enables rapid response on the ESP32. The resulting architecture is therefore appropriate for a low-cost assistive device in which immediate local response is more important than computationally intensive scene interpretation.

User Interface Implementation

The user interface was implemented through the Blynk mobile application and designed to provide simple access to language selection and system information. The dashboard contains a segmented language-selection control and monitoring widgets for obstacle status, directional guidance, and sensor readings. The language-selection widget allows the user or authorized caregiver to select English, Igbo, or Ezza dialect. The obstacle-status and directional widgets display the current system state, while the sensor-reading widget provides the measured distances from the three sensing directions.

Table 1.6: Blynk Dashboard Widgets

Widget	Pin	Function	User Action
Segmented Switch	V1	Language Selection	Tap to switch between English, Igbo, and Ezza
Label (Status)	V2	Obstacle Status	Displays the current obstacle status
Value Display	V3	Sensor Readings	Displays left, center, and right sensor distances (cm)
Label (Direction)	V4	Direction Guidance	Displays the current navigation instruction
Notification	–	Critical Alerts	Automatically notifies caregiver when obstacles are detected

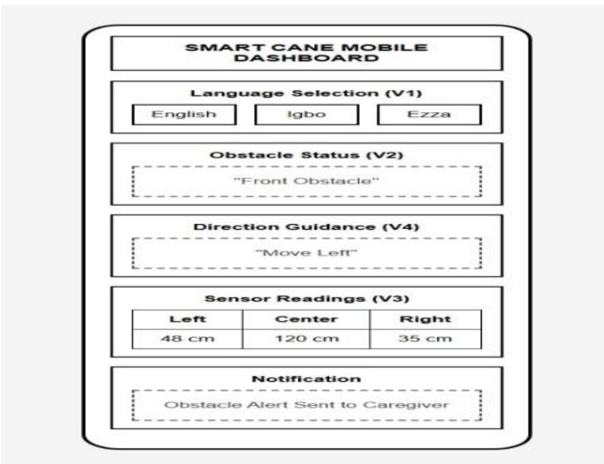

Figure 1.7: Blynk Mobile Application Interface

The interface also supports automatic notifications when an obstacle event is detected. This provides a supplementary caregiver-monitoring function without requiring the caregiver to remain physically close to the user. The Blynk interface and its virtual-pin configuration are presented in Figure 4.8 and Table 4.9.

Final Prototype Implementation

The integrated prototype combines the sensing, processing, audio, power, and communication subsystems within a conventional cane structure. The electronics enclosure contains the ESP32, DFPlayer Mini, voltage-regulation circuitry, and battery pack, while the three ultrasonic sensors are externally mounted at different heights and orientations. The speaker is positioned near the handle for direct audio delivery to the user.

Figure 1.8: Assembled Smart Cane Prototype

The completed prototype therefore provides a compact assistive-navigation platform capable of detecting obstacles in three directions, generating directional voice instructions in three supported languages, operating its core navigation functions without internet connectivity, and communicating selected system information to a mobile IoT platform when connectivity is available. The assembled prototype is shown in Figure 1.8. The implementation establishes a low-cost, locally autonomous, multilingual assistive-navigation architecture in which real-time environmental sensing and safety-critical voice feedback are performed at the device level, while IoT connectivity provides optional remote configuration and monitoring. This architecture directly reflects the study's objective of developing an assistive system suitable for multilingual and resource-constrained environments while maintaining portability, affordability, and operational independence.

System Testing and Evaluation

The developed smart cane was evaluated to determine whether it satisfied its functional and non-functional requirements, with particular emphasis on three-directional obstacle detection, rule-based navigation decisions, multilingual audio feedback, Blynk-based language selection, real-time response, caregiver notification, and offline operation. Testing was performed using the fully assembled prototype under controlled laboratory conditions. Wooden boards, cardboard boxes, and human subjects were used as representative obstacles at predetermined distances. The evaluation comprised unit testing, integration testing, and system-level testing. Unit testing examined the individual sensor, decision-logic, audio-playback, and language-management modules. Integration testing subsequently verified communication among the sensing, processing, audio, and IoT subsystems, while system testing assessed the complete prototype against the predefined requirements. This staged approach enabled faults to be isolated at module level before evaluating the complete system.

Unit and Integration Testing

The sensor interface was tested by placing obstacles at known distances and comparing the measured values with the expected obstacle states. The decision module was evaluated using different combinations of blocked and unobstructed sensor directions, while the audio subsystem was tested by triggering each navigation message in the three supported languages. The language-management module was tested by changing the selected language through the Blynk application and verifying the corresponding audio output. Integration testing confirmed successful interaction between the sensor and decision modules, the decision module and DFPlayer Mini, and the complete sensing–decision–audio–IoT chain. The results showed that the individual modules operated correctly and that their integration did not introduce functional errors.

Obstacle Detection Performance

The three HC-SR04 ultrasonic sensors were evaluated independently at distances of 30, 50, 80, 100, and 150 cm. Each sensor successfully detected obstacles at 30, 50, and 80 cm and correctly classified distances of 100 and 150 cm as clear. This behaviour was consistent across the left, centre, and right sensing directions. The results demonstrate that the implemented 80-cm danger threshold was correctly applied by the controller. The three-sensor arrangement also provided directional information that enabled the system to distinguish between left-, centre-, and right-side obstacles rather than simply indicating the presence of an obstacle.

Table 1.7: Obstacle Detection Test Results

Sensor	30cm	50cm	80cm	100cm	150cm	Detection Status
LEFT	✓ Detected	✓ Detected	✓ Detected	✗ Clear	✗ Clear	Working
CENTER	✓ Detected	✓ Detected	✓ Detected	✗ Clear	✗ Clear	Working
RIGHT	✓ Detected	✓ Detected	✓ Detected	✗ Clear	✗ Clear	Working

Navigation Decision-Logic Performance

The decision algorithm was evaluated using clear-path, single-obstacle, and multiple-obstacle configurations. All tested scenarios produced the expected navigation messages. A clear path generated a "Clear path" instruction, while individual left, centre, and right obstacles generated corresponding directional warnings. Combined obstacles resulted in appropriate movement instructions, including "Move right" for front-and-left blockage and "Move left" for front-and-right blockage. When all three directions were blocked, the system generated a front-obstacle warning. The expected and actual outputs agreed in all tested scenarios, indicating correct implementation of the rule-based decision algorithm. The programmed 2-s message interval also reduced repeated audio announcements and helped prevent audio flooding during continuous obstacle detection.

Table 1.8: Decision Logic Test Results

Scenario	Left	Center	Right	Expected Message	Actual Message	Verdict
Clear Path	Clear	Clear	Clear	"Clear path"	"Clear path"	✓ Pass
Front Only	Clear	Blocked	Clear	"Obstacle in front!"	"Obstacle in front!"	✓ Pass
Left Only	Blocked	Clear	Clear	"Obstacle on left!"	"Obstacle on left!"	✓ Pass
Right Only	Clear	Clear	Blocked	"Obstacle on right!"	"Obstacle on right!"	✓ Pass
Front + Left	Blocked	Blocked	Clear	"Move right"	"Move right"	✓ Pass
Front + Right	Clear	Blocked	Blocked	"Move left"	"Move left"	✓ Pass
All Blocked	Blocked	Blocked	Blocked	"Obstacle in front!"	"Obstacle in front!"	✓ Pass

Multilingual Audio and Language Switching

The multilingual subsystem was evaluated using nine navigation message types in English, Igbo, and Ezza, giving a total of 27 locally stored audio files. Each file was triggered individually and verified through the DFPlayer Mini and speaker/headphone output. All 27 files were successfully retrieved and played, corresponding to a 100% successful audio-playback rate during the test. The Blynk-based language-selection mechanism was also evaluated by switching sequentially among English, Igbo, and Ezza. Each language change was correctly registered by the ESP32 and resulted in audio output from the newly selected language. The Serial Monitor confirmed the selected language, while the corresponding voice output provided additional confirmation. These results demonstrate that the proposed offset-based language-management approach provides reliable multilingual audio selection without requiring continuous cloud-based speech synthesis during navigation.

Table 1.9: Multilingual Audio Test Results

Message Type	English File	Igbo File	Ezza File	Verdict
Clear path	001.mp3 ✓	011.mp3 ✓	021.mp3 ✓	Pass
Obstacle in front!	002.mp3 ✓	012.mp3 ✓	022.mp3 ✓	Pass
Obstacle on left!	003.mp3 ✓	013.mp3 ✓	023.mp3 ✓	Pass
Obstacle on right!	004.mp3 ✓	014.mp3 ✓	024.mp3 ✓	Pass
Move left or right	005.mp3 ✓	015.mp3 ✓	025.mp3 ✓	Pass
Move forward or right	006.mp3 ✓	016.mp3 ✓	026.mp3 ✓	Pass
Move forward or left	007.mp3 ✓	017.mp3 ✓	027.mp3 ✓	Pass
Move left	008.mp3 ✓	018.mp3 ✓	028.mp3 ✓	Pass
Move right	009.mp3 ✓	019.mp3 ✓	029.mp3 ✓	Pass

Table 1.10: Blynk Language Switching Test Results

Action	Serial Output	Voice Output	Verdict
Switch to English	"Language: English"	"Clear path" (English)	✅ Pass
Switch to Igbo	"Language: Igbo"	"Uzo di n'iru" (Igbo)	✅ Pass
Switch to Ezza	"Language: Ezza"	"Eza di n'iru" (Ezza)	✅ Pass
Switch to English	"Language: English"	"Clear path" (English)	✅ Pass

Real-Time Response Performance

System response time was evaluated over 10 trials by placing an obstacle approximately 50 cm from the centre sensor and measuring the interval between obstacle detection and the corresponding audio output. Recorded response times ranged from 115 to 141 ms, with a mean response time of 126 ms. The measured mean was substantially below the predefined 500-ms response-time requirement, indicating that the ESP32-based processing architecture was capable of providing timely obstacle warnings. The relatively narrow range of response times also suggests consistent processing behaviour under the tested laboratory conditions.

Table 1.11: Real-Time Response Test Results

Trial	Response Time (ms)
1	125
2	118
3	134
4	122
5	141
6	115
7	128
8	132
9	120
10	126
Average	126 ms

Offline Operation and IoT Performance

Offline operation was evaluated by disconnecting the Wi-Fi network after confirming normal system operation. Obstacle detection and locally stored voice feedback continued to operate after network disconnection, demonstrating that the core navigation functions were independent of internet connectivity. In contrast, Blynk-based language switching and cloud notifications were unavailable without Wi-Fi. This finding confirms the hybrid architecture of the system: safety-critical navigation functions are executed locally on the ESP32, whereas non-critical remote functions such as caregiver monitoring and mobile language configuration depend on network connectivity. Such separation improves operational resilience in environments with unreliable internet access.

Table 1.12: Offline Operation Test Results

Function	With Wi-Fi	Without Wi-Fi	Status
Obstacle Detection	✅ Working	✅ Working	Core navigation works offline
Voice Feedback	✅ Working	✅ Working	Core navigation works offline
Language Switching (App)	✅ Working	✖ Not Working	Requires Wi-Fi
Blynk Notifications	✅ Delivered	✖ Not Delivered	Requires Wi-Fi

Summary of Test Results

The overall evaluation showed successful completion of all major functional and non-functional tests. The three ultrasonic sensors detected obstacles within the specified 80-cm threshold; all 27 multilingual audio files played correctly; language switching through Blynk operated successfully; caregiver notifications were delivered when network connectivity was available; and the mean response time of 126 ms remained substantially below the 500-ms requirement. Core obstacle detection and voice-navigation functions also remained operational without internet connectivity.

Table 1.13: Test Results Summary

Requirement	Test	Result	Status
Functional Requirements			
Obstacle Detection (3 directions)	Place obstacles at measured distances	All 3 sensors detect within 80cm	✓ Pass
Multilingual Audio Feedback	Trigger all message types in 3 languages	27/27 files play correctly	✓ Pass
Language Switching via App	Switch languages via Blynk	Instant switching confirmed	✓ Pass
Caregiver Monitoring	Detect obstacle, check Blynk notification	Notifications delivered	✓ Pass
Non-Functional Requirements			
Real-Time Response	Measure detection-to-output time	~126ms (requirement: <500ms)	✓ Pass
Offline Capability	Disconnect Wi-Fi, test core functions	Core functions work offline	✓ Pass

Based on the defined test cases, the prototype achieved a 100% pass rate across the tested core requirements. This result demonstrates functional compliance with the system specifications under the controlled conditions in which testing was conducted.

Evaluation against Research Objectives

The experimental findings indicate that the four principal objectives of the study were achieved. First, the developed cane successfully integrated three HC-SR04 sensors to provide left, centre, and right obstacle detection, with an average response time of 126 ms. Second, the system successfully provided multilingual audio feedback in English, Igbo, and Ezza, with all 27 tested audio files functioning correctly. Third, the Blynk application successfully provided language selection and remote system monitoring. The segmented language-selection control allowed switching among the three supported languages, while caregiver notifications provided remote awareness of detected obstacles when network connectivity was available. Fourth, systematic testing of sensor performance, decision logic, audio playback, language switching, response time, and offline operation confirmed the functional performance of the completed prototype.

Evaluation of System Performance against Research Objectives

Research objective	Major evidence	Outcome
Develop a smart cane with directional obstacle detection	Three HC-SR04 sensors; detection in left, centre and right directions	Achieved
Implement multilingual audio feedback	27/27 audio files successfully played in English, Igbo and Ezza	Achieved
Provide language selection and caregiver monitoring	Blynk language switching and remote notifications	Achieved
Validate system performance	Unit, integration and system tests; 126-ms mean response; offline operation	Achieved

II. Discussion of Findings

The findings demonstrate that the proposed three-sensor configuration provides effective directional awareness within the tested environment. Unlike a conventional cane that primarily provides tactile information about objects encountered near the ground, the proposed arrangement enables the system to distinguish obstacles on the left, centre, and right and subsequently generate direction-specific warnings. The consistent detection behaviour observed across the tested distances indicates that the selected 80-cm threshold was suitable for the controlled indoor evaluation. The decision-logic results further demonstrate that a relatively lightweight rule-based approach can provide deterministic navigation responses without requiring computationally intensive artificial intelligence. The successful handling of both single and combined obstacle conditions indicates that the ESP32 is adequate for the implemented sensing and decision tasks. The 2-s message interval provided a practical mechanism for limiting repetitive announcements while maintaining continuous environmental monitoring. The multilingual evaluation represents one of the principal contributions of the system. All 27 audio files were successfully played, and switching among English, Igbo, and Ezza was achieved through the Blynk interface. Because the audio resources are stored locally, navigation feedback does not depend on an internet connection or a cloud-based speech-generation service. This is particularly relevant to resource-constrained environments where network availability may be inconsistent. The mean response time of 126 ms further demonstrates the suitability of the ESP32 for real-time operation. Since the measured response was substantially below the 500-ms design requirement, the system was capable of producing warnings rapidly under the tested conditions. However, this result should be interpreted as a prototype laboratory measurement rather than a direct measure of user safety or collision avoidance performance. The offline test produced an important architectural finding. Core obstacle detection, decision-making, and audio feedback remained functional without Wi-Fi, whereas Blynk-based language switching and cloud notifications were unavailable. Thus, the system does not depend on internet connectivity for its fundamental navigation function. Instead, connectivity serves as a supplementary layer for configuration and caregiver monitoring.

System Limitations

Despite the successful test results, several limitations remain. First, the prototype does not provide voice recognition; language selection is performed through the Blynk mobile application. Second, multilingual support is currently limited to English, Igbo, and Ezza, although the file-mapping architecture allows additional languages to be incorporated. Third, the use of pre-recorded audio restricts the system to a predefined set of navigation messages and does not provide dynamic speech generation. Fourth, ultrasonic sensing can be affected by environmental conditions, object geometry, surface characteristics, and sensor positioning. Fifth, the present evaluation was primarily conducted under controlled conditions; therefore, additional testing involving diverse outdoor environments, terrain conditions, moving obstacles, and visually impaired users would be necessary to establish broader real-world usability. Finally, because Blynk-based functions depend on network connectivity, remote caregiver monitoring and application-based language switching cannot be guaranteed when Wi-Fi is unavailable. Nevertheless, this limitation does not prevent the core obstacle-detection and local audio-navigation functions from operating.

Overall Evaluation and Summary

Overall, the experimental evaluation demonstrates that the developed prototype successfully integrates three-directional ultrasonic sensing, rule-based navigation, locally stored multilingual audio, and IoT-enabled monitoring into a low-cost smart cane. The system achieved successful operation across the defined test scenarios, with a 126-ms mean response time, successful playback of 27 multilingual audio files, functional language switching, and continued core navigation operation during network disconnection. The results therefore support the feasibility of the proposed architecture for low-cost, offline-capable and linguistically inclusive assistive navigation. However, the findings should be interpreted within the limitations of the controlled testing environment and the prototype's predefined language and audio-message structure. Future work should focus on user-centred field evaluation, additional Nigerian languages and dialects, voice-based interaction, dynamic speech generation, improved sensing of vertical and moving obstacles, and enhanced power and connectivity management.

References

[1]. Adebara, I., Elmadany, A., & Abdul-Mageed, M. (2024). Cheetah: Natural language generation for 517 African languages. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

[2]. DeRenzi, B., Dixon, A., Farhi, M. A., & Resch, C. (2025). Synthetic voice data for automatic speech recognition in African languages. *Proceedings of the First Workshop on Advancing NLP for Low-Resource Languages*, 152–186.

[3]. Imam, S. H., Sani, B., Gete, D. K., Ahmed, B. Y., Ahmad, I. S., Abdulmumin, I., Yimam, S. M., Bello, M. Y., & Muhammad, S. H. (2025). Automatic speech recognition for African low-resource languages: Challenges and future directions. *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, 89–94.

[4]. Katumba, A. (2025). Building text-to-speech models for low-resourced languages from crowdsourced data. *Applied AI Letters*.

[5]. Okolo, G. I., Althobaiti, T., & Ramzan, N. (2025). Smart assistive navigation system for visually impaired people. *Journal of Disability Research*.

[6]. Tulluri, D. D., Vaidya, S. P., Tyagi, A., & Gottamkula, S. D. N. K. (2025). An IoT-based smart cane for improving mobility and safety of the visually impaired. *2025 IEEE 6th India Council International Subsections Conference (INDISCON)*.

[7]. Wazirali, R., Ashrif, F. F., & Ahmad, R. (2025). AI smart cane technology and assistive navigation for visually impaired users: An overview. *Journal of King Saud University Computer and Information Sciences, 37*, 226.

[8]. World Health Organization (WHO). (2023). *Increasing eye care interventions to address vision impairment*. Geneva: World Health Organization.

[9]. World Health Organization (WHO). (2026). *Blindness and vision impairment*. World Health Organization.

[10]. Abidi, M. H., Noor Siddiquee, A., Alkhalefah, H., & Srivastava, V. (2024). A comprehensive review of navigation systems for visually impaired individuals. *Heliyon, 10*(11), e31825.

[11]. Kathiria, P., Mankad, S. H., Patel, J., Kapadia, M., Kapadia, M., & Lakdawala, N. (2024). Assistive systems for visually impaired people: A survey on current requirements and advancements. *Neurocomputing, 606*, 128284.

[12]. Bauer, Andreas, & Gallo, Matteo. (2025). Low-cost smart cane for visually impaired people with pathway surface transition point detection. Information, 16(8), 707.

[13]. Anagnostopoulos, Panagiotis, Drakopoulos, George, Kalogeropoulos, Christos, & Drosou, Angeliki. (2022). IoT-enabled intelligent stick for visually impaired people for outdoor navigation. Sensors, 22(24), 9725.

[14]. Adegboye, Adebayo, Oladipo, Samson, & Ajiboye, Joseph. (2023). Dialectal variation and challenges in speech recognition for African languages. International Journal of Language Technology, 9(1), 45–58.

[15]. Benjamin, D., Ali, A., Schepis, A., & MacKenzie, I. (2017). Guide dogs and their use in enhancing the mobility of visually impaired persons: A review. Disability and Rehabilitation: Assistive Technology, 12(6), 573–581.

[16]. Okpor, Martin. (2023). Artificial intelligence and linguistic inclusivity in assistive technologies: Opportunities for Africa. AI & Society, 38(1), 15–28.

[17]. Chen, Wei, Zhang, Yu, & Huang, Ming. (2023). Cultural adaptation in speech-based assistive systems: A case study on multilingual TTS. Multimedia Tools and Applications, 82(9), 13521–13545.

[18]. Clark-Carter, D., Heyes, A., & Howarth, C. (2023). Effect of preview distance on walking speed for visually impaired individuals. Journal of Computing, 34:65-79

[19]. Rahman, Md. Shafiqul, & Islam, Tanvir. (2024). Smart assistive navigation system for visually impaired people. Journal of Digital Research, 5(1), Article 0086. ScienceOpen.

[20]. Ezeani, Ignatius, Hepple, Mark, Onyenwe, Ikechukwu, & Uchechukwu, Chinedu. (2020). Igbo-English text-to-speech for language learning applications. Language Resources and Evaluation, 54(3), 753–776.

[21]. Fiorini, Sergio, Peres, Lucas, & Marchi, Rafael. (2022). Computer vision in assistive navigation: Opportunities and challenges. Sensors, 22(12), 4217.

[22]. Khan, Muhammad, Rauf, Aisha, & Ullah, Bilal. (2022). Multilingual accessibility in assistive devices: A framework for inclusive design. Assistive Technology Journal, 34(4), 422–435.