www.ijarp.com ISSN 2456-9992 Page: 01-15

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 07

PI TALK: ESP32 BASED AI VOICE ASSISTANT USING CLOUD-BASED ARTIFICIAL INTELLIGENCE AND VOICE INTERACTION

***Prerana Kumari Singh, Krishna Kant Chouhan, Rohit Vishwakarma, Simran Verma Sumanta Karmakar**

Asansol Engineering College.

Article Received: 05 June 2026, Article Revised: 25 June 2026, Published on: 15 July 2026

***Corresponding Author: Prerana Kumari Singh**
Asansol Engineering College.
Doi: https://doi-doi.org/101555/ijarp.3381

ABSTRACT

Artificial Intelligence (AI) voice assistants have become an essential component of modern smart environments by enabling natural and hands-free human-machine interaction. Commercial solutions such as Amazon Alexa, Google Assistant, and Apple Siri provide advanced conversational capabilities but often operate within closed ecosystems that limit customization and increase deployment costs. This paper presents Pi Talk, an ESP32 based AI voice assistant that integrates cloud-based Artificial Intelligence services with embedded hardware to create a compact, portable, and cost-effective intelligent assistant platform. The proposed system incorporates an INMP441 I2S microphone, MAX98357A I2S audio amplifier, SSD1306 OLED display, rechargeable battery module, charging circuitry, and speaker. Voice commands are captured through the microphone and transmitted via Wi-Fi to cloud-based Speech-to-Text (STT) and Artificial Intelligence services. The generated response is processed using Text-to-Speech (TTS) technology and played through the speaker while the OLED display provides real-time visual feedback regarding system status. Experimental evaluation demonstrates reliable speech recognition, low response latency, efficient power utilization, and seamless voice interaction. The proposed architecture provides a practical solution for smart home automation, educational platforms, assistive technologies, and Internet of Things (IoT) based applications requiring intelligent voice interaction.

INTRODUCTION

Artificial Intelligence has significantly transformed the way humans interact with digital systems. Among the most influential developments in this field is the emergence of voice assistants that allow users to communicate with machines through natural language. By eliminating the need for conventional input devices such as keyboards and touchscreens, voice assistants offer a more intuitive and accessible method of interaction.

Commercial voice assistant platforms such as Siri, Alexa, and Google Assistant have demonstrated the practical potential of conversational Artificial Intelligence in everyday life. These systems are capable of answering questions, controlling smart home devices, providing reminders, and performing numerous automated tasks. Despite their widespread adoption, such solutions are typically proprietary and operate within closed ecosystems. As a result, developers face challenges related to limited customization, restricted hardware integration, vendor dependency, and concerns regarding privacy and data ownership.

Recent advancements in embedded systems and Internet of Things technologies have created opportunities for developing custom voice assistant platforms using low-cost microcontrollers. The ESP32 microcontroller has emerged as a powerful solution for such applications due to its dual-core architecture, integrated Wi-Fi connectivity, low power consumption, and native support for digital audio communication through the Inter-IC Sound (I2S) protocol. These capabilities enable the ESP32 to efficiently interface with microphones, speakers, and cloud-based services while maintaining a compact and energy-efficient design.

The objective of this work is to develop a portable and intelligent voice assistant named Pi Talk, which combines embedded hardware with cloud-based Artificial Intelligence services. The proposed system captures user voice commands, transmits them to cloud servers for processing, and delivers intelligent spoken responses in real time. In addition to audio interaction, the system incorporates an OLED display that provides visual feedback regarding operational states such as listening, processing, and response generation.

The major contributions of this work are summarized as follows:

1. Design and implementation of a compact AI-powered voice assistant using the ESP32-S3 microcontroller.
2. Integration of cloud-based Speech-to-Text, Artificial Intelligence, and Text-to-Speech services for conversational interaction.
3. Development of a portable battery-powered architecture suitable for standalone deployment.

4. Implementation of a multimodal feedback mechanism through audio and visual interfaces.
5. Demonstration of a low-cost and scalable solution for smart home and IoT applications.

The remainder of this paper is organized as follows. Section II reviews related work in the field of embedded voice assistants. Section III describes the system architecture. Section IV presents the methodology. Section V explains the operational principle of the proposed system. Section VI discusses experimental results. Section VII highlights advantages and limitations, while Sections VIII and IX present future work and conclusions respectively.

RELATED WORK -The development of intelligent voice assistants has attracted significant attention from researchers and industry practitioners. Early implementations primarily relied on powerful Single Board Computers (SBCs) such as Raspberry Pi systems. These platforms provided sufficient computational capability for speech recognition and natural language processing tasks while supporting operating systems capable of running complex software frameworks. Numerous studies successfully integrated Raspberry Pi devices with cloud-based services such as Google Assistant SDK and Amazon Alexa Voice Service. Although these systems demonstrated robust functionality, they were associated with higher power consumption, increased hardware costs, and larger physical footprints.

To overcome these limitations, researchers explored microcontroller-based solutions using the ESP32 family of devices. Initial ESP32 based implementations focused primarily on home automation and voice-controlled switching applications. These systems benefited from low cost, integrated wireless connectivity, and energy-efficient operation. However, due to limited processing resources, most early solutions relied on predefined voice commands or external mobile applications for command interpretation. Consequently, they lacked the conversational capabilities associated with modern Artificial Intelligence systems.

Recent developments have introduced hybrid edge-cloud architectures that leverage both local processing and cloud computing resources. In these systems, the ESP32 performs lightweight operations such as audio acquisition, buffering, and wake-word detection, while computationally intensive tasks including speech recognition, natural language understanding, and speech synthesis are executed on remote cloud servers. This approach combines the energy efficiency of embedded hardware with the virtually unlimited computational resources available through cloud infrastructure.

Several studies have demonstrated the effectiveness of cloud-assisted voice assistants for IoT environments. Advanced speech recognition services and Large Language Models (LLMs) have significantly improved the accuracy and contextual understanding of conversational systems. However, many existing implementations remain limited by fragmented hardware designs, lack of portability, and absence of visual feedback mechanisms. Most prototypes operate as development-board demonstrations without integrated battery management systems or dedicated user interfaces.

Another significant challenge in existing voice assistant systems is user uncertainty during processing delays. Since cloud communication introduces unavoidable latency, users often experience confusion regarding the current operational state of the system. The absence of visual indicators reduces overall usability and negatively impacts user experience.

The proposed Pi Talk system addresses these limitations through the integration of a portable power management subsystem and an OLED-based visual interface. By combining cloud-powered conversational intelligence with real-time visual feedback, the system provides a more interactive and user-friendly Human-Machine Interface. The architecture bridges the gap between low-cost embedded platforms and advanced Artificial Intelligence technologies, resulting in a practical solution for modern IoT applications.

SYSTEM DESIGN AND ARCHITECTURE

The Pi Talk voice assistant is designed using a hybrid edge-cloud architecture that combines embedded hardware processing with cloud-based Artificial Intelligence services. The architecture is modular in nature and consists of hardware acquisition units, communication modules, cloud processing services, and output interfaces. This design enables efficient voice interaction while maintaining low hardware cost and power consumption.

The overall system is divided into two primary layers:

- Edge Layer – Responsible for voice acquisition, local processing, display management, and wireless communication.
- Cloud Layer – Responsible for Speech-to-Text (STT), Artificial Intelligence processing, and Text-to-Speech (TTS) generation.

The ESP32 serves as the central controller and coordinates communication between all peripheral devices and cloud services.

A. Hardware Components

The proposed system utilizes the following hardware components:

(1)ESP32 Microcontroller-The ESP32 acts as the main processing unit of the system. It features a dual-core processor, integrated Wi-Fi connectivity, low power consumption, and native support for the Inter-IC Sound (I2S) protocol. These features make it suitable for real-time audio streaming and cloud communication.

2) INMP441 Digital Microphone-The INMP441 is an omnidirectional MEMS microphone that provides high-quality digital audio output using the I2S communication protocol. It captures user voice commands and transmits them directly to the ESP32.

3) MAX98357A Audio Amplifier-The MAX98357A is a Class-D digital audio amplifier with an integrated Digital-to-Analog Converter (DAC). It receives digital audio data from the ESP32 and converts it into amplified analog signals suitable for speaker output.

4) SSD1306 OLED Display

A 0.96-inch OLED display is used to provide visual feedback to users. It displays system states such as:

- System Initialization
- Wi-Fi Connection Status
- Listening Mode
- Processing State
- Response Generation
- Error Notifications

5) Speaker-The speaker converts amplified electrical signals into audible sound and delivers synthesized speech responses generated by the Artificial Intelligence engine.

6) Battery Module and Charging Circuit-A rechargeable lithium-ion battery powers the entire system. A TP4056 charging module provides battery charging and protection against overcharging, over-discharging, and short-circuit conditions.

B. System Architecture Overview

The operational workflow begins when the user speaks a voice command. The microphone captures the audio signal and transmits it to the ESP32 through the I2S interface. The microcontroller processes and streams the audio to cloud-based speech recognition services using Wi-Fi connectivity.

The cloud server converts the audio into text and forwards the text to an Artificial Intelligence model for analysis. The AI engine generates an appropriate response which is subsequently converted into speech using a Text-to-Speech service. The synthesized audio is returned to the ESP32 and played through the speaker.

Simultaneously, the OLED display updates the current operational status, creating a multimodal interaction experience.

C. Data Flow Pipeline

The complete data flow of the system can be summarized as follows:

User Voice

↓

INMP441 Microphone

↓

ESP32 Controller

↓

Cloud Speech-to-Text Engine

↓

Artificial Intelligence Model

↓

Text-to-Speech Engine

↓

ESP32 Controller

↓

MAX98357A Amplifier

↓

Speaker Output

This architecture ensures reliable communication between local hardware and cloud services while maintaining efficient resource utilization.

METHODOLOGY

The operational methodology of the Pi Talk voice assistant follows a sequential state-machine model. Each state represents a specific stage of system operation, ensuring efficient resource management and responsive user interaction.

A. System Initialization

When power is supplied to the device, the ESP32 executes its initialization sequence. During this stage:

- Hardware peripherals are configured.
- OLED display is initialized.
- Wi-Fi connectivity is established.
- Cloud APIs are authenticated.
- Memory buffers are allocated.

After successful initialization, the system enters standby mode.

B. Wake Word Detection

To minimize unnecessary processing and power consumption, the system continuously monitors incoming audio for predefined wake words.

Examples include:

- “Pi Talk”
- “Hello Assistant”
- “Jarvis”

Detection of a valid wake word activates the voice acquisition stage.

C. Voice Acquisition

The INMP441 microphone captures the user's speech and converts acoustic signals into digital audio samples.

The ESP32 receives this data using the I2S interface and stores it temporarily within memory buffers for transmission.

D. Speech-to-Text Conversion

The captured audio stream is transmitted to a cloud-based Speech-to-Text service.

The STT engine performs:

- Acoustic analysis
- Noise filtering
- Speech segmentation
- Text transcription

The output is a machine-readable text representation of the user's command.

E. Artificial Intelligence Processing

The transcribed text is forwarded to a cloud-based Artificial Intelligence model.

The AI system performs:

- Natural Language Understanding (NLU)
- Context Analysis
- Intent Recognition
- Response Generation

For example:

User Query:

“What is the weather today?”

AI Response:

“Today's weather is 32°C with partly cloudy skies.”

F. OLED Status Management

Throughout system operation, the OLED display continuously updates the current processing state.

Typical display messages include:

- Listening...
- Processing...
- Thinking...
- Response Ready

This visual feedback significantly improves user experience by reducing uncertainty during cloud-processing delays.

G. Text-to-Speech Generation

The response generated by the AI model is transmitted to a cloud-based TTS service.

The TTS engine converts textual information into natural-sounding speech using advanced speech synthesis techniques.

H. Audio Playback

The synthesized audio stream is downloaded by the ESP32 and transmitted to the MAX98357A amplifier.

The amplifier converts the digital signal into an analog waveform and drives the speaker, producing an audible response.

I. Return to Standby Mode

After playback completion, all temporary buffers are cleared and the system automatically returns to wake-word detection mode.

This cyclic operation enables continuous and uninterrupted voice interaction.

WORKING PRINCIPLE

The operational principle of the proposed Pi Talk voice assistant is based on a synchronized edge-cloud communication framework that enables natural and intelligent voice interaction. The system converts human speech into machine-readable data, processes it through Artificial Intelligence services, and returns the generated response as synthesized speech.

A. Acoustic Signal Acquisition

The interaction process begins when the user speaks a command. The INMP441 digital microphone captures the acoustic signal and converts it into a digital audio stream using the I2S protocol. Unlike conventional analog microphones, the digital output minimizes noise and improves signal quality.

The ESP32 receives the audio samples through Direct Memory Access (DMA), ensuring efficient audio buffering without excessive processor utilization.

B. Speech Processing and Cloud Communication

Once the audio data is captured, the ESP32 establishes communication with cloud-based services through its integrated Wi-Fi module. The recorded audio stream is securely transmitted to a Speech-to-Text (STT) engine.

The STT service performs:

- Speech recognition
- Language processing
- Text generation

The resulting text is forwarded to an Artificial Intelligence model capable of understanding user intent and generating contextual responses.

C. Artificial Intelligence Response Generation

The AI engine processes the transcribed text using Natural Language Processing (NLP) techniques. It analyzes sentence structure, extracts user intent, and generates a meaningful response.

For example:

User Input:

"Tell me about Artificial Intelligence."

AI Output:

"Artificial Intelligence is a branch of computer science that enables machines to simulate human intelligence and perform cognitive tasks."

The generated response is transmitted to a Text-to-Speech service for audio synthesis.

D. Speech Synthesis and Audio Playback

The Text-to-Speech engine converts the generated text into natural-sounding speech. The synthesized audio is downloaded by the ESP32 and streamed through the I2S interface to the MAX98357A amplifier.

The amplifier converts the digital signal into analog form and drives the speaker to produce the final audio response.

E. Visual User Feedback

During each stage of operation, the SSD1306 OLED display provides real-time status updates. The display may show:

- Ready
- Listening
- Processing
- Generating Response
- Speaking

This multimodal feedback mechanism significantly improves user interaction by informing users about the current state of system operation.

RESULTS AND DISCUSSION

The developed Pi Talk prototype was successfully implemented and tested under multiple operating conditions to evaluate system functionality, reliability, and user interaction performance.

The integrated hardware platform demonstrated stable operation throughout continuous testing. Voice commands were successfully captured, transmitted to cloud services, processed by the Artificial Intelligence engine, and returned as synthesized speech responses.

A. Functional Evaluation

The system successfully completed the following operations:

- Wake-word detection
- Voice command acquisition
- Speech-to-Text conversion
- Artificial Intelligence query processing
- Text-to-Speech generation
- Audio response playback
- OLED status display

No significant hardware failures or communication interruptions were observed during testing.

B. Performance Analysis

Table I summarizes the observed system performance.

Table I. System Performance Evaluation

Parameter	**Observation**
Voice Recognition Accuracy	High
Response Generation	Successful
OLED Feedback	Real-Time
Audio Quality	Clear and Audible
Wi-Fi Connectivity	Stable
Portability	Excellent
Power Consumption	Low
System Reliability	High

The ESP32 effectively handled audio acquisition and cloud communication simultaneously without noticeable performance degradation.

C. DISCUSSION

The proposed architecture demonstrates the feasibility of implementing advanced conversational AI capabilities on a low-cost embedded platform. By offloading computationally intensive tasks to cloud services, the system achieves functionality comparable to commercial voice assistants while maintaining reduced hardware complexity.

The OLED display proved particularly beneficial during testing by providing users with immediate visual confirmation of processing states. This reduced user uncertainty during cloud communication delays and improved overall user experience.

ADVANTAGES AND LIMITATIONS - A. Advantages

The proposed Pi Talk system offers several advantages:

1) Low-Cost Implementation

The system utilizes affordable embedded hardware components, significantly reducing overall development costs compared to commercial alternatives.

2) Portable Design

The rechargeable battery and charging module enable fully portable operation without requiring continuous external power.

3) Real-Time AI Interaction

Cloud-based processing allows access to advanced Artificial Intelligence capabilities without requiring powerful local hardware.

4) Energy Efficiency

The ESP32 architecture provides low power consumption while maintaining high performance.

5) Multimodal User Interface

The combination of audio and visual feedback improves interaction quality and user experience.

6) Expandability

The open architecture allows future integration of sensors, cameras, actuators, and IoT devices.

B. Limitations

Despite its advantages, the system exhibits several limitations:

1) Internet Dependency

Core functionalities including STT, AI processing, and TTS require continuous internet connectivity.

2) Cloud Service Dependency

The system relies on external cloud APIs which may introduce operational costs and service restrictions.

3) Battery Constraints

Continuous Wi-Fi communication and audio processing reduce battery runtime.

4) Environmental Noise

Speech recognition accuracy may decrease in noisy environments.

5) Network Latency

Response speed depends on internet bandwidth and cloud server availability.

FUTURE WORK-Several enhancements can be incorporated into future versions of the Pi Talk system.

A. Edge Artificial Intelligence

Future implementations may integrate lightweight on-device speech recognition and Natural Language Processing models to reduce cloud dependency.

B. Smart Home Integration

The assistant can be extended to control IoT devices through MQTT, Bluetooth Low Energy (BLE), Zigbee, or Wi-Fi protocols.

C. Computer Vision Support

Integration of camera modules would enable facial recognition, object detection, and visual context awareness.

D. Multi-Language Support

Future versions can support multiple regional and international languages to improve accessibility.

E. Mobile Application Development

A dedicated mobile application can be developed for remote configuration, monitoring, firmware updates, and device management.

F. Enhanced Security

Advanced encryption and authentication mechanisms can be implemented to improve privacy and secure communication.

CONCLUSION

This paper presented the design, implementation, and evaluation of Pi Talk, an ESP32 based Artificial Intelligence voice assistant capable of providing intelligent voice interaction through cloud-based services. The proposed system integrates speech recognition, Artificial Intelligence processing, and speech synthesis within a compact embedded platform while maintaining low power consumption and portability.

The developed architecture successfully combines the advantages of edge computing and cloud computing. The ESP32 performs local audio acquisition and communication tasks, while computationally intensive operations are executed within cloud infrastructure. This hybrid approach enables sophisticated conversational capabilities without requiring expensive processing hardware.

Experimental evaluation confirmed the effectiveness of the proposed design in terms of voice recognition, response generation, system reliability, and user interaction. The addition of an OLED-based visual feedback system further enhances usability by providing real-time status information throughout the interaction process.

The proposed Pi Talk platform demonstrates that advanced Artificial Intelligence services can be deployed on affordable embedded hardware, making intelligent voice assistants more accessible for educational, industrial, healthcare, assistive technology, and smart home applications. Future enhancements involving edge AI, computer vision, multilingual support, and IoT integration will further expand the capabilities of the system and contribute to the advancement of next-generation human-machine interaction systems.

REFERENCES

1. J. Gubbi et al., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645– 1660, 2013.
2. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
3. A. Coucke et al., "Snips Voice Platform: An Embedded Spoken Language Understanding System for Private-by-Design Voice Interfaces," arXiv:1805.10190, 2018.
4. T. Brown et al., "Language Models are Few-Shot Learners," NeurIPS, 2020.

5. A. Vaswani et al., “Attention Is All You Need,” NeurIPS, 2017.
6. A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” NeurIPS, 2012.
7. K. He et al., “Deep Residual Learning for Image Recognition,” CVPR, 2016.
8. A. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017.
9. C. Shorten and T. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, 2019.
10. OpenAI, “GPT Models Technical Overview,” 2024.
11. Espressif Systems, “ESP32 Technical Reference Manual,” 2024.
12. Espressif Systems, “ESP-IDF Programming Guide,” 2024.
13. M. McTear, “The Rise of the Conversational Interface,” Springer, 2020
14. D. Jurafsky and J. Martin, Speech and Language Processing, Pearson, 2023.
15. X. Chen et al., “Overview of Speech Recognition Algorithms and Their Applications,” 2025.
16. S. Gupta et al., “Speech Recognition-Based Wireless Control System Using ESP32,” Electronics, 2025.
17. A. Kumar et al., “Voice Controlled Application Using ESP32,” IRO Journals, 2025.
18. P. Huang et al., “Energy Efficient Voice Controlled Smart Home Systems,” arXiv, 2025.
19. M. Hewitt and H. Cunningham, “Taxonomic Classification of IoT Smart Home Voice Control,” arXiv, 2022.
20. Sumit Kumar et al., “IoT Based Personal Voice Assistant,” arXiv:2305.17788, 2023.
21. “Development of an AI-Powered Voice Assistant,” IJSRSET, 2025.
22. “Online AI Based Voice Assistant,” ResearchGate Publication, 2024.
23. “Voice Assistant Using Automated Speech Recognition,” IJRPR, 2025.
24. “Enhanced AI Voice Assistance Using Machine Learning and NLP,” ResearchGate Publication, 2025.
25. “ESP32 Offline Voice Recognition Using Edge Impulse,” CircuitDigest, 2025.
26. “AI Voice Assistant Device Using IoT,” Journal of Advancement in Electronics Design, 2026.
27. “ESP32-S3 AI Voice Assistant: A CloudAugmented Conversational System,” IJLRP, 2026.