# Voice Controlled Application using ESP32

**Ruban Gladwin M.**[1]**, Jeya Preeda R.**[2]**, Dayana S.**[3]**, Pon Aarthi G.**[4]**, Sharmila Fathima V.**[5]**, Vengadeswari I.**[6]

[1]Associate Professor, [2]Assistant Professor, [3-6] UG Student, Department of ECE, Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.

**Email:** [1]rubangladwin@gmail.com, [2]preeda2009john@gmail.com, [3]dayanaananthi1702@gmail.com, [4]aarthi2574@gmail.com, [5]georgesanjay066@gmail.com, [6]ssiyappan1979@gmail.com

**Abstract**

The ESP32 microcontroller is used in this project to develop voice applications for device automation. The ESP32's built-in Bluetooth and Wi-Fi capabilities make it the perfect platform for Internet of Things (IoT) applications, where it can perform voice-activated tasks both online and offline. An external microphone records the voice commands, which the system stores before being interpreted by edge-based speech recognition software like TensorFlow Lite and Vosk, or cloud-based services like Google Assistant or Amazon Alexa. Relays or the ESP32's GPIO pins are used to translate the voice commands into control signals that can be used to operate lights, fans, and other devices. The system can function in any network condition thanks to its support for both offline and cloud computing. The application showcases the versatility of the ESP32 in real-time voice-activated home systems, which improve accessibility and user convenience through voice control.

**Keywords:** ESP32, Wi-Fi, Internet of Things, GPIO.

## 1. Introduction

The development of assistive and smart home automation technologies has been greatly impacted by the explosive growth of embedded systems and the Internet of Things (IoT) in recent years. Out of all the user interfaces currently available, voice-controlled interfaces have become one of the most convenient and useful methods of human-computer interaction. By allowing users to access and operate devices and perform actions via voice commands, these


*DOI: https://doi.org/10.36548/jeea.2025.3.002*
*Received: 21.06.2025, received in revised form: 24.07.2025, accepted: 11.08.2025, published: 22.08.2025*

systems enhance usability, particularly for the elderly, physically disabled individuals, or in situations where hands-free use is preferred.

The ESP32 microcontroller from Espressif Systems is a powerful and affordable platform for these kinds of applications. The ESP32's dual-core processor, built-in Wi-Fi, Bluetooth (Classic and BLE), robust support for digital and analog I/O, and other features enable the design of low-power, networked devices. These features make the ESP32 a solid option for developing voice-activated Internet of Things applications, where real-time processing and wireless connectivity are essential.

This project investigates the design and development of an ESP32-based voice-activated application for controlling electrical appliances, including fans, lights, and other common household devices. The system integrates the ESP32 with a microphone or voice input module to allow it to detect and understand user commands. These commands can be handled by cloud-based voice recognition services like Google Assistant or Amazon Alexa, which require internet connectivity, or by edge-based/offline models like TensorFlow Lite and Vosk, which can operate remotely.

In addition to demonstrating that real-time voice control in embedded systems is feasible, the software places a strong emphasis on modularity, flexibility, and accessibility. By eliminating physical switches and smartphone apps, the system offers a more natural user experience and opens the door for future development of personalized, voice-activated smart environments.

## 2. Related Works

Assistive robotics and voice-activated home automation have advanced significantly in recent years, often utilizing low-cost microcontrollers such as the ESP32. To improve usability and accessibility, Supriya et al. [1] developed a smart home system that enables voice-activated automation of household appliances for individuals with disabilities. This is in line with broader trends toward assistive technology and inclusion.

By creating an MFCC-CNN model that operates directly on an ESP32 in combination with a web application interface, Wisudawan et al. [2] have proven successful on-device voice recognition techniques. Their method demonstrates the feasibility of real-time voice processing on low-resource hardware. Similarly, Li et al. [3] demonstrated the versatility of applications in fields ranging from home automation to robotics by using the ESP32 to voice-control unmanned vehicles.

Security and cloud integration have also been studied. Roy et al. [4] suggested a cloud-based voice control system for household appliances that combines security layers with IoT platforms to guarantee secure operation. In order to provide redundancy and flexibility in smart home configurations, Subki et al. [5] concurrently integrated ESP32 hardware with manual switches and Alexa integration.

Ashu et al. [7] improved these implementations with a home automation system that had built-in voice control and remote access, and was optimized for responsiveness and energy efficiency. Their system provides deployment insights that are applicable to real-world scenarios. By developing a near-real-time anger detection system with ESP32, Fernandez-Morales et al. [8] also showed how voice analysis transcends commands to emotional tracking, advancing voice-based AI into affective computing.

Hercog et al. [9] reviewed ESP32-based IoT devices, discussing general hardware aspects and offering implementation strategies like integrated speech recognition and peripheral management. Litayem [10] also demonstrated scalable and affordable smart home control using ESP32-S3, emphasizing affordable automation platforms that are best suited for a variety of users and environments.

## 3. Existing Work

Voice-based systems have drawn major interest in the fields of smart homes, assistive technology, and IoT-driven automation. Numerous research works and business offerings have explored the integration of voice control with embedded devices for controlling various electrical and electronic appliances. Early voice-controlled application deployments were largely driven by cloud-based platforms such as Google Assistant, Amazon Alexa, and Microsoft Cortana. Such systems provided audio input on remote servers, enabling high accuracy but requiring constant internet connectivity. For instance, home automation systems like Google Nest and Amazon Echo utilize cloud-based natural language processing (NLP) to recognize commands and control smart devices via Wi-Fi. While efficient, these solutions raise privacy concerns and are not practical for locations with poor or no internet connectivity.

To address these limitations, developers and researchers have studied offline speech recognition systems on embedded platforms. Vosk, PocketSphinx, and TensorFlow Lite offer architectures for voice processing on edge devices such as Raspberry Pi or microcontrollers with sufficient processing capacity. Offline voice recognition improves system responsiveness, offers data privacy, and allows application in isolated or bandwidth-constrained environments. Hardware-wise, the ESP32 microcontroller has gained popularity as it supports Wi-Fi and Bluetooth onboard, features a dual-core processor, and has low power consumption. Voice-control projects based on ESP32 typically use external modules such as microphone arrays, analog sound sensors, or I2S microphones to capture voice input. The inputs are either transmitted to a cloud platform via Wi-Fi for processing or processed locally with the assistance of lightweight models. Additionally, current research has included smartphone apps or third-party APIs to allow communication between the ESP32 and cloud services, allowing further freedom of control. Voice-controlled assistants based on Bluetooth or smartphone speech-to-text systems are utilized by others, which are then executed within the ESP32. Work on command-specific keyword-spotting models (e.g., "turn light on," "fan off") has also been investigated, enabling limited but helpful offline control with few resources.

Although such advancements have been achieved, most currently available systems are either heavily cloud-based or cannot process in real time locally, causing latency or reduced functionality. Most commercial systems are also proprietary and expensive, making them unsuitable for low-cost or DIY purposes. Therefore, there is a growing need for hybrid voice-controlled systems that accommodate the flexibility of ESP32 hardware and offer both cloud-based and offline voice recognition features with the promise of reliability, affordability, and data privacy. The contribution made in this paper attempts to address these gaps by developing a dual-mode ESP32-based voice-controlled system with the capability to be used in both connected and standalone modes.

## 4. Methodology

Every system component receives the necessary electrical energy from the power supply unit, which also maintains steady voltage and current levels for reliable operation. The process begins with a voice command from the user, such as "Switch on the light" or "Turn on the fan." The voice recognition module receives and processes the verbal input, then analyzes the audio signal to determine and recognize the desired command. The ESP32 AI board, which can also receive audio input straight from its microphone, receives a signal from the module upon identification. By comprehending complex speech patterns, the ESP32 uses Tiny Machine Learning (TinyML) algorithms to increase recognition accuracy. The relay module acts as the devices' electronic switch. Depending on the instruction, the relay allows the ESP32 to turn devices on or off. Finally, by translating voice commands into physical movements, the output devices such as the fan and LED respond to the processed commands, closing the automation loop.

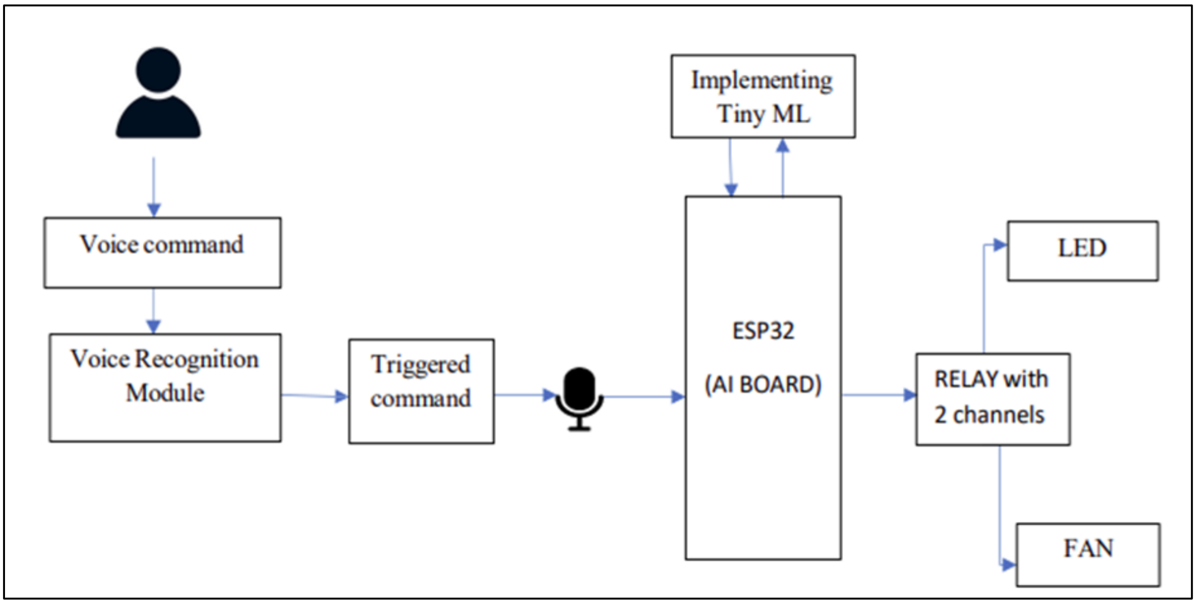


**Figure 1.** System Architecture of the Proposed System

## 5. Hardware Details

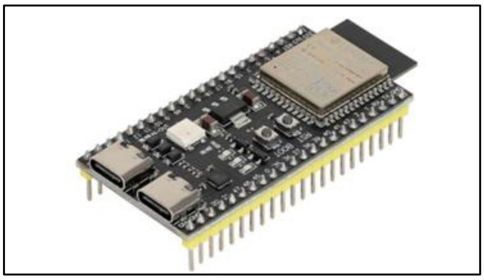

**Figure 2.** ESP32-S3

The ESP32-S3(refer to fig 2) collects data from various sensors connected to its I/O pins. These sensors might monitor environmental conditions such as temperature, humidity, gas levels, or light intensity. The data is then processed by the ESP32-S3's processor (either the core 0 or core 1).

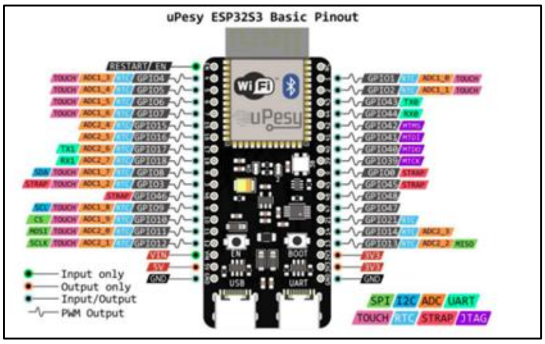


**Figure 3.** ESP32-S3 Pin Diagram

With built-in Wi-Fi and Bluetooth, the ESP32-S3 (refer to fig 3) is a potent microcontroller that's perfect for automation and Internet of Things applications. Available GPIO pins, power pins, and communication interfaces like ADC, DAC, UART, SPI, and I2C are all shown in the pin diagram. By connecting sensors, actuators, and other peripheral devices to the ESP32-S3, developers can create a flexible hardware integration for machine learning and smart home automation applications.

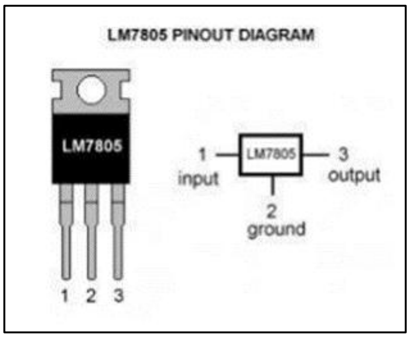


**Figure 4.** IC7805

(Refer to fig 4) The IC7805 voltage regulator provides a steady +5V DC output in the suggested system, which is necessary for the ESP32-S3 and related parts to operate dependably. It guarantees that delicate modules like the relay and ESP32 are shielded from voltage fluctuations by controlling input variations, preserving steady system performance.

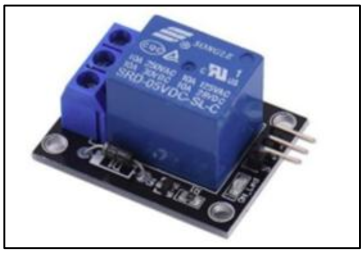

**Figure 5.** Relay

The relay module serves as the primary switching interface between high-power household appliances and the ESP32-S3 microcontroller. This work ensures safe isolation between the high-voltage load and the low-power ESP32 signals by using a single-channel relay to control devices like fans and LEDs with voice commands.

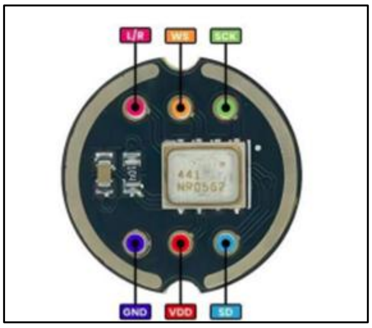


**Figure 6.** INMP441 MIC

High-performance I2S microphone INMP441(refer to fig 6) is intended for smart home automation and voice-activated systems. It ensures high-fidelity audio with little distortion and noise by transmitting digital audio data to microcontrollers via an I2S interface. It provides crisp, accurate sound for voice recognition tasks with a 48 kHz sample rate and 16-bit resolution. Because of its low power consumption, it is perfect for home automation systems and battery-operated devices.

**Figure 7.** LED

(Refer to fig 7) In this system, the LED serves as a visual indicator and an appliance to test the operation of voice-activated commands. The LED instantly reacts when the ESP32 receives a command such as "Turn on the light," confirming that the command was successfully executed.

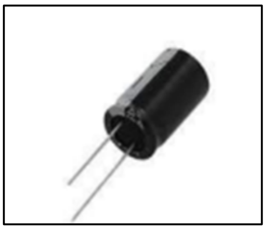

**Figure 8.** Capacitor

(Refer to fig. 8) The circuit design incorporates capacitors to stabilize voltage levels and reduce power supply fluctuations. By ensuring that the relay and ESP32 board receive clean, steady power, they lower the possibility of malfunctions caused by brief spikes in power.

**Figure 9.** Resistor

(Refer to fig 9) The system uses resistors to bias and limit current. They guarantee steady operation throughout the circuit while shielding delicate parts from excessive current, including LEDs, sensors, and the ESP32 microcontroller.

## 6. Simulation setup

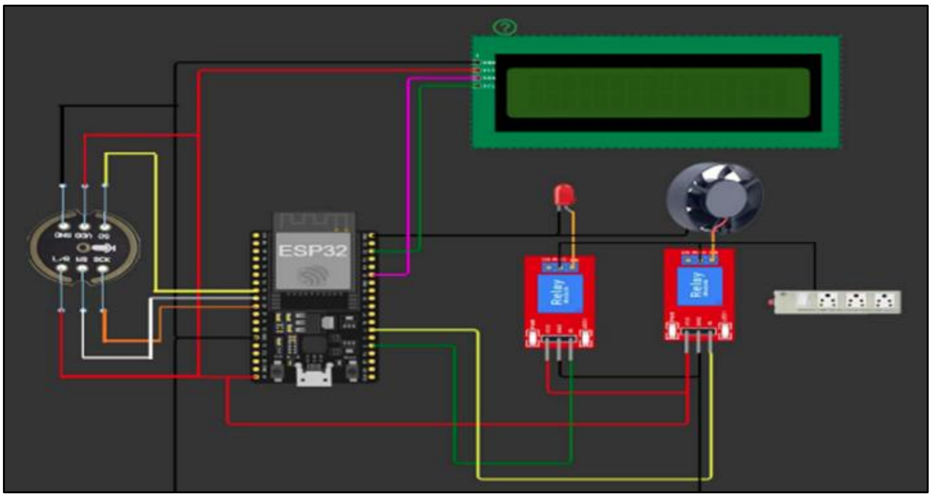


**Figure 10.** Circuit Diagram of the Proposed System

Figure 10 shows the simulation diagram of the proposed system.

ISSN: 2582-3051

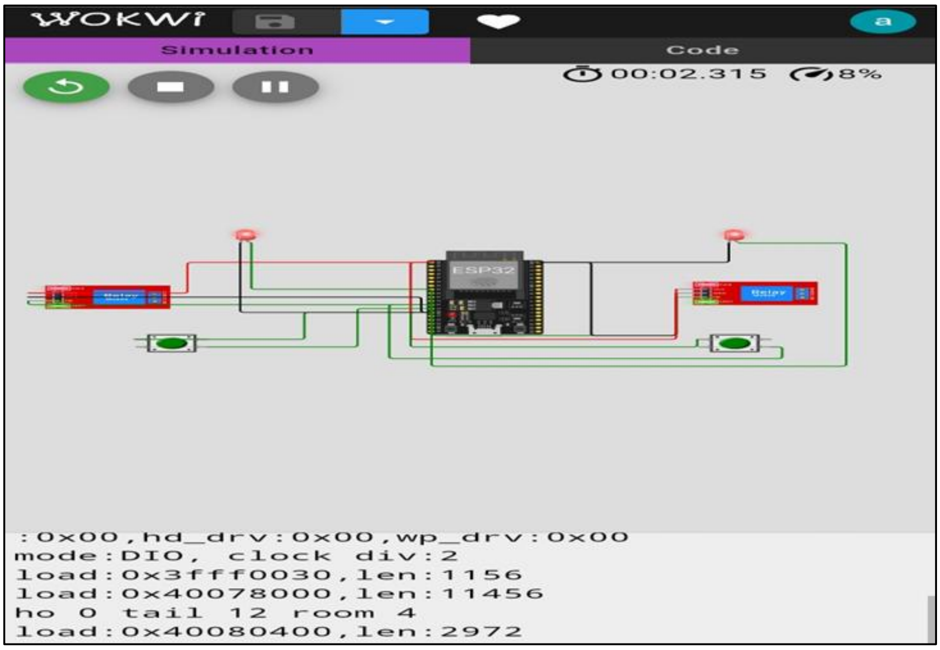


**Figure 11.** Simulation Output

Figure 11 illustrates the simulation output of the proposed system.

## 7. Experimental Setup and Results

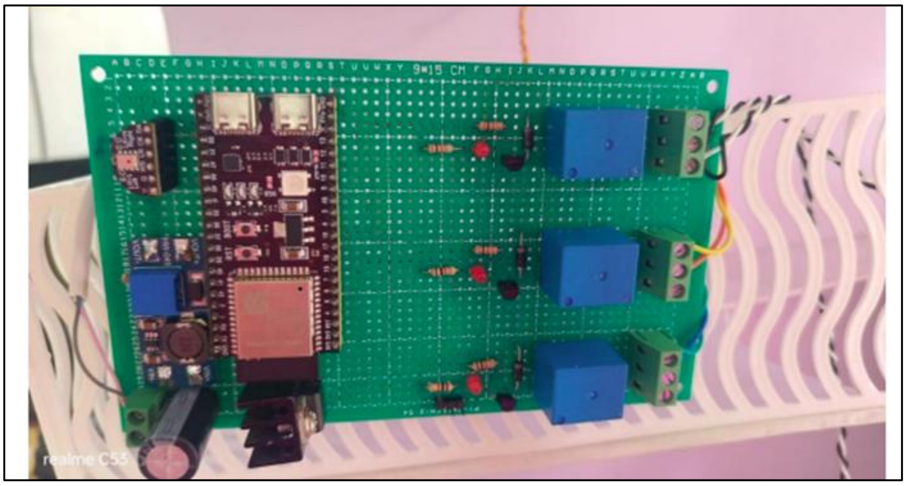

**Figure 12.** Hardware Design of the Proposed System

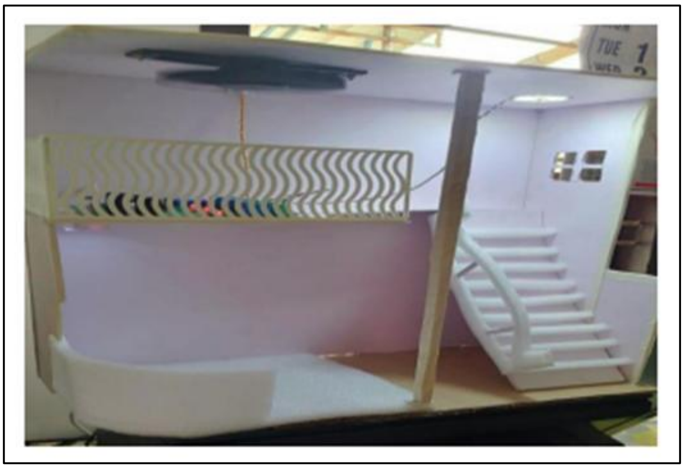

**Figure 13.** Hardware Kit of the Proposed System

Figures 12 and 13 show the proposed work setup. Voice commands driven by artificial intelligence are used to control devices via an ESP32-S3 microcontroller. Among the pre-trained trigger words that the system can identify are "Alexa, turn on the LED," "Alexa, turn on LED2," and "Alexa, turn on the air conditioner." Upon detecting these phrases, the ESP32-S3 turns on the air conditioner or LED lights by activating the appropriate GPIO pins. For offline and responsive interaction, Edge Impulse or comparable platforms are used to train the voice model. For edge voice recognition tasks, the ESP32-S3 is effective due to its built-in AI acceleration. It offers low latency and privacy by interpreting and executing commands without requiring an internet connection. For safe switching, relay module drivers connect the devices.

## 8. Future Scope

In the future, the system can be expanded to include more voice commands to control a greater variety of household appliances, including TVs, air conditioners, and smart locks. Many users can operate the system without any prior training.Additionally, more people will be able to use the system if it is multilingual. By combining motion, light, and temperature sensors, environment-dependent tasks can be automated. System configuration and remote control can be done via a mobile app and Bluetooth/Wi-Fi connectivity. The system would become more intelligent, flexible, and appropriate for state-of-the-art smart home applications with these improvements.

## 9. Conclusion

This project successfully builds a voice-controlled home automation system using the ESP32-S3 microcontroller and TinyML for offline voice recognition. By using simple voice commands to control appliances like fans and light bulbs, it eliminates the need for internet connectivity. By employing a wake word detection mechanism that activates the system when required, energy efficiency is guaranteed. The system uses a machine learning model trained with Edge Impulse to locally process voice commands, resulting in faster response times, improved privacy, and increased reliability. Overall, it provides a practical, reasonably priced, and user-friendly smart home automation solution.

## References

[1] Supriya, N., S. Surya, and K. N. Kiran. "Voice Controlled Smart Home for Disabled." In 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), IEEE, (2024): 1-4.

[2] Wisudawan, Hasbi NP, Suatmi Murnani, and Hendra Setiawan. "On-Device MFCC-CNN Voice Recognition System with ESP-32 and Web-Based Application." In 2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, (2023): 161-166.

[3] Li, Lecheng, Yunlong Guo, and Hengzheng Li. "Design of voice control system for unmanned vehicle based on ESP32." In IET Conference Proceedings CP989, vol. 2024, no. 21, Stevenage, UK: The Institution of Engineering and Technology, (2024): 496-500.

[4] Roy, Suman, Himanshu Sen, Indranil Roy, Rankana Paul, Farhan Akthar, and Ujjwol Barman. "Cloud-Enabled Voice Control for Home Appliances: A Secure and Accessible IoT Approach." In 2024 IEEE 4th International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC), IEEE, (2024): 1-5.

[5] Subki, Muhammad Nazhiim Bin Subki, Lilywati Binti Bakar Bakar, and Zurina Binti Abdul Wahab Abdul Wahab. "Home Automation using ESP-32 Development Board with Alexa Voice Assistant and Manual Switch." Progress in Engineering Application and Technology 4, no. 2 (2023): 319-332.

[6] Gupta, Meenu, Rakesh Kumar, Raju Kumar Chaudhary, and Jayshree Kumari. "IoT based voice controlled autonomous robotic vehicle through google assistant." In 2021 3rd international conference on advances in computing, communication control and networking (ICAC3N), IEEE, (2021): 713-717.

[7] Ashu, Kumar, Adhishree Srivastava, Muskan Muskan, and Himanshu Priyadarshi. "Efficient Home Automation System with Remote Access and Voice Control Facility." IETE Journal of Research (2025): 1-12.

[8] Fernandez-Morales, Ivan, Hugo Mitre-Hernandez, Carlos Lara-Alvarez, Hector De-La-Torre-Gutierrez, and Uziel Jaramillo-Avila. "Compact near Real-Time Anger Detection System through Voice Analysis Implemented on ESP32 Microcontroller." IEEE Transactions on Instrumentation and Measurement (2025).

[9] Hercog, Darko, Tone Lerher, Mitja Truntič, and Oto Težak. "Design and implementation of ESP32-based IoT devices." Sensors 23, no. 15 (2023): 6739.

[10] Litayem, Nabil. "Scalable smart home management with ESP32-S3: A low-cost solution for accessible home automation." In 2024 International Conference on Computer and Applications (ICCA), IEEE, (2024): 1-7