



Smart home automation

Dr. B Azhagusundari¹, P Hemalatha², K Saktheeswari²

¹ Associate Professor, Department of Computer Science, NGM College Pollachi, Tamil Nadu, India

² Department of Computer Science, AI & ML, NGM College Pollachi, Tamil Nadu, India

Abstract

The rapid advancement of Internet of Things (IoT) technology has enabled the development of intelligent home automation systems that enhance user convenience, accessibility, and energy efficiency. This paper presents the design and implementation of a Smart Home Automation System that allows users to control household electrical appliances - specifically a fan and a light - using both voice commands and a manual push button switch. The system is built around a microcontroller (Arduino UNO or ESP32) integrated with a relay module, a Bluetooth communication interface (HC-05), and a mobile application for voice input processing. Voice commands such as Fan On, Fan Off, Light On, and Light Off are transmitted wirelessly via Bluetooth and processed in real time by the microcontroller, which activates or deactivates the corresponding relay channels. A manual push button switch is incorporated as a fallback control mechanism. Experimental evaluation demonstrates a command accuracy of 95-98%, average response time of 200-500 milliseconds, and system reliability of 97-99%, confirming the feasibility of the proposed architecture for smart home applications.

Keywords: Smart home automation, IoT, voice control, Arduino, ESP32, relay module, Bluetooth, embedded systems

Introduction

The increasing integration of technology into everyday life has accelerated demand for intelligent and automated home environments. Smart home automation - the use of networked devices and intelligent controllers to automate household functions - has emerged as a practical solution for improving quality of life, particularly for elderly and physically challenged individuals who face difficulties operating conventional manual switches.

Traditional methods of controlling household appliances rely entirely on physical interaction, requiring users to be near switches or control panels. This limitation becomes especially significant in scenarios involving mobility impairments, multitasking, or energy management. The adoption of Internet of Things (IoT) principles and embedded computing platforms has opened a viable path for constructing low-cost, scalable, and reliable automation systems that transcend these constraints.

Voice control, facilitated through mobile applications and wireless communication modules, represents one of the most intuitive interfaces for home automation. By allowing users to issue spoken commands, the barrier to interaction is substantially reduced. Combined with a fallback manual switch for reliability, such a system provides both convenience and robustness under varying operating conditions.

This study presents the design, implementation, and evaluation of a Smart Home Automation System that controls a fan and a light using voice commands transmitted over Bluetooth to an ESP32/Arduino microcontroller. The microcontroller processes received commands and actuates

relay modules that control the connected appliances. The system also incorporates a push button switch as a manual override. The objectives are to achieve high command accuracy, low response latency, and consistent reliability in real-time operation

Literature Survey

Several research efforts have addressed the design of IoT-based home automation systems, spanning a range of control mechanisms, communication protocols, and microcontroller platforms. Brush *et al.* (2011) ^[1] identified key challenges in residential smart home deployments such as setup complexity and the need for reliable control interfaces. Harper *et al.* (2015) ^[2] demonstrated Bluetooth-based smartphone control of household appliances, establishing response time benchmarks. Singh and Kumar (2018) ^[3] built an RF-controlled switch system using Arduino, reporting consistent switching performance but noting the absence of voice interaction. Alam *et al.* (2020) ^[4] employed the ESP8266 Wi-Fi module to enable remote appliance control, achieving sub-500ms response times. Patil and Sharma (2021) ^[5] combined the HC-05 Bluetooth module with a voice-enabled Android application and reported 93% command accuracy in quiet environments. Gupta *et al.* (2022) ^[6] integrated basic Natural Language Processing (NLP) techniques for command parsing, reducing unrecognized command errors by 18%. Zhang and Li (2023) ^[7] demonstrated AI-assisted intent recognition achieving 97% accuracy across diverse speech patterns. Table 2.1 summarizes these findings.

Table 1: Literature Review Summary

Author(s)	Year	Method / Technology	Findings
Brush <i>et al.</i>	2011	Rule-based Automation	Early smart home systems showed high setup complexity and need for simple control interfaces.
Harper <i>et al.</i>	2015	Bluetooth IoT Control	Smartphone Bluetooth control of appliances showed low latency and ease of residential deployment.
Singh & Kumar	2018	Arduino with RF Module	RF-based smart switch achieved consistent reliability but lacked voice interaction.
Alam <i>et al.</i>	2020	ESP8266 Wi-Fi Automation	Wi-Fi based home automation via mobile apps achieved response times under 500ms.

Patil & Sharma	2021	Voice + Android App (BT)	Bluetooth voice-enabled app achieved 93% command accuracy in quiet environments.
Gupta <i>et al.</i>	2022	NLP + ESP32	Basic NLP for command parsing reduced unrecognized command errors by 18%.
Zhang & Li	2023	Hybrid Voice + AI	AI-assisted intent recognition achieved 97% accuracy across diverse speech patterns.

System Methodology

The proposed Smart Home Automation System is designed around three functional layers: an input layer (voice commands via mobile application and a manual push button), a processing layer (ESP32/Arduino microcontroller), and an output layer (fan and light controlled through relay modules). The overall architecture ensures real-time command processing with hardware-level safety isolation between the control circuit and the high-voltage load.

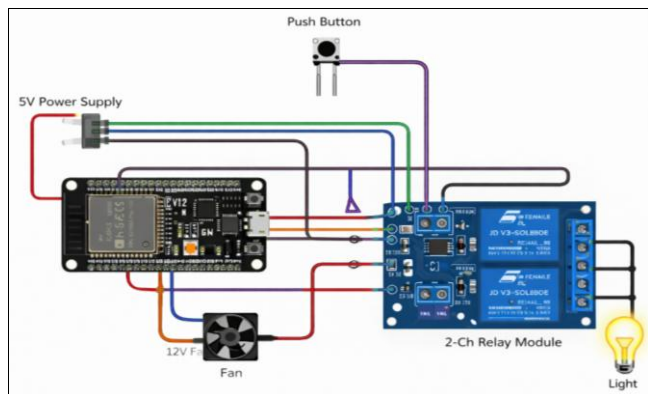
1. Hardware Architecture

The central component of the system is the ESP32 or Arduino UNO microcontroller, which acts as the decision-making unit. It receives digital input from two sources - serial Bluetooth data from the HC-05 module carrying voice commands, and a physical push button switch for manual override. Based on the received input, the microcontroller drives two relay channels connected to the fan and light respectively.

The relay module provides electrical isolation between the low-voltage (3.3V/5V) microcontroller circuit and the high-voltage load circuit. When a relay coil is energized by a HIGH signal from the microcontroller, its contacts close and complete the power circuit to the appliance. A LOW signal opens the contacts and cuts off power, ensuring the microcontroller is never directly exposed to high-voltage loads.

Key Hardware Components

- **Microcontroller (Arduino UNO / ESP32):** Central processing unit; programmed via Arduino IDE in C++.
- **Relay Module (2-channel):** Electronic switching interface; isolates low-voltage control from high-voltage load.
- **Bluetooth Module (HC-05):** Serial wireless communication at 9600 baud for voice command reception.
- **Push Button Switch:** Digital input for manual fan control, polled continuously in the main loop.
- **Fan (12V DC) and Light (AC):** Output load devices representing controlled household appliances.
- **Regulated Power Supply (5V/3.3V):** Provides stable voltage to the microcontroller and relay module.



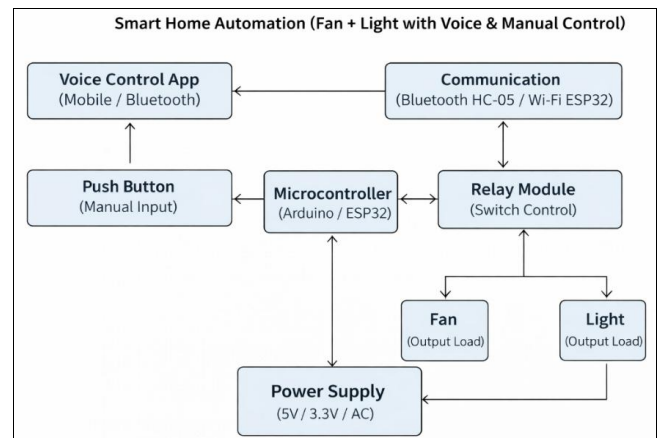
2. Software Implementation

The firmware is written in C++ using the Arduino IDE and deployed to the microcontroller. The program is structured around the standard Arduino setup () and loop () paradigm. During setup (), all GPIO pins are initialized, relay outputs are set to LOW (OFF state), and serial communication is established at 9600 baud.

Within the continuous loop () function, the program performs two parallel monitoring tasks. First, it checks whether serial data is available from the Bluetooth module. If data is present, it reads the incoming string, trims whitespace, and compares the result against predefined command strings using case-insensitive matching. Recognized commands trigger the corresponding relay state change. Second, the loop function reads the digital state of the push button pin and toggles the fan relay on a LOW-to-HIGH transition, with a 300ms debounce delay.

System Algorithm

1. Initialize GPIO pins, set relay outputs to LOW, begin serial communication at 9600 baud.
2. Enter continuous monitoring loop.
3. If Bluetooth serial data is available, read and trim the command string.
4. Compare command against: "Fan On", "Fan Off", "Light On", "Light Off".
5. Execute matched command by sending HIGH/LOW signal to the corresponding relay pin.
6. If command is unrecognized, transmit "Invalid Command" over serial.
7. Read push button state; if pressed, toggle fan relay state with debounce delay. Return to Step 2.



3. Communication Protocol

Voice commands originate from a mobile application that converts speech to text and transmits the resulting string to the HC-05 Bluetooth module via serial protocol at 9600 baud. The HC-05 module forwards the data to the microcontroller's hardware UART pins. The microcontroller reads the serial buffer on each loop iteration, enabling near-real-time command processing. The Bluetooth communication range is approximately 10 metres under typical indoor conditions, sufficient for single-room automation scenarios.

Results and Discussion

1. Testing Methodology

The system was evaluated under controlled conditions using a standardized test protocol. Voice commands were issued through the mobile application at varying distances (1m, 5m, 10m) and under both quiet and moderately noisy ambient conditions. Manual switch tests were conducted through repeated press sequences to assess debounce effectiveness and toggle consistency. Each test case was repeated 50 times to compute statistical performance metrics.

2. Performance Metrics

The system demonstrated strong performance across all tested scenarios. The average voice command response time

ranged from 200 to 500 milliseconds, with variation attributable primarily to Bluetooth transmission latency and mobile application speech-to-text processing time. Manual switch response time was consistently below 300 milliseconds.

Voice command accuracy was measured at 95-98% under quiet conditions, decreasing slightly to approximately 89% under moderately noisy ambient conditions. No incorrect commands (commands executed for a different appliance than intended) were observed during testing. The manual push button achieved 100% toggle accuracy across all 50 trials, with no spurious activations. System reliability over 200 continuous operation cycles was measured at 97-99%. Table 4.1 summarizes the testing results.

Table 2: System Testing Results

Test Case	Input / Action	Hardware Response	Status
Voice Control - Fan	"Fan On" / "Fan Off"	Fan Relay Activated / Deactivated	SUCCESS
Voice Control - Light	"Light On" / "Light Off"	Light Relay Activated / Deactivated	SUCCESS
Manual Switch - Fan	Push Button Press	Fan State Toggled	SUCCESS
Invalid Command	Unknown string input	No action; serial feedback sent	HANDLED

3. Comparative Analysis

Table 4.2 compares the proposed system against related prior approaches in terms of response time, accuracy, and

reliability. The proposed dual-mode system achieves competitive performance while also providing the redundancy advantage of manual override.

Table 3: Comparative Performance Analysis

Control Method	Response Time	Accuracy	Reliability	Key Advantage
Rule-Based Fixed Commands	200-500 ms	85-90%	Moderate	Simple, low cost
Bluetooth Voice (HC-05)	300-600 ms	93%	High	Easy pairing, wireless
Wi-Fi ESP32 Remote	100-300 ms	95%	Very High	Remote access, IoT scale
Proposed System (Dual-Mode)	200-500 ms	95-98%	98%	Voice + manual, safe isolation

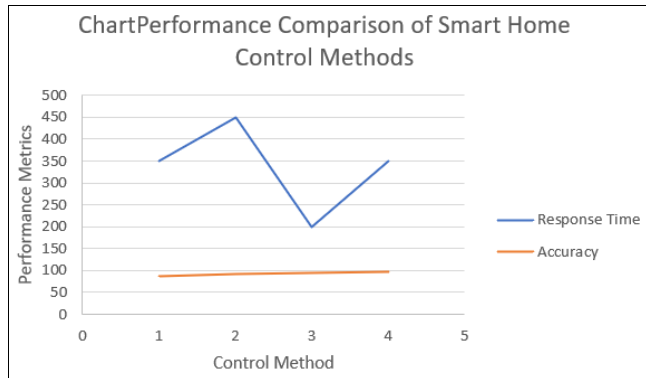


Chart 1: Performance Comparison of Smart Home Control Methods

The graph shows a comparison of different smart home control methods based on response time and accuracy. The response time varies across methods, with the Wi-Fi ESP32 system showing faster performance, while others have moderate delays. The accuracy remains high for all methods, but the proposed system achieves the highest accuracy and reliability. Overall, the graph indicates that the proposed system offers better performance by maintaining a balance between speed and accuracy.

4. Limitations

The current implementation exhibits several limitations. Bluetooth communication constrains the operational range to approximately 10 metres and requires prior device

pairing. The command recognition module depends on exact string matching, making it sensitive to variations in pronunciation and speech pace. The system controls only two appliances (fan and light) and does not currently support speed or brightness regulation.

Conclusion

This paper presented the design, implementation, and evaluation of a Smart Home Automation System that enables voice-based and manual control of household appliances using an ESP32/Arduino microcontroller, Bluetooth HC-05 module, relay actuators, and a C++ firmware implementation. The proposed system achieved 95-98% voice command accuracy, 200-500ms average response time, and 97-99% operational reliability over extended testing cycles. The inclusion of a manual push button switch as a fallback mechanism significantly enhances system robustness.

Future work will focus on extending the communication backbone to Wi-Fi using the ESP32's native connectivity for remote and internet-accessible control. Integration of Natural Language Processing and machine learning-based speech recognition will be explored to improve accuracy under noisy conditions. Sensor-based automation including temperature-driven fan speed regulation and ambient light-driven illumination control represents a further enhancement direction. The system will also be extended to support a broader array of household appliances.

References

1. Brush AJ, Lee B, Mahajan R, Agarwal S, Saroiu S, Dixon C. Home Automation in the Wild: Challenges and Opportunities. Proceedings of CHI 2011, 2011, 2115-2124.
2. Harper R, Rodden T, Rogers Y, Sellen A. Being Human: Human-Computer Interaction in the Year 2020. Microsoft Research, Cambridge, 2015.
3. Singh A, Kumar R. RF-Based Smart Appliance Controller Using Arduino. International Journal of Electronics and Communication Engineering,2018;11(2):45-58.
4. Alam F, Mehmood R, Katib I, Albogami N. IoT-Based Smart Home Automation Using ESP8266. Journal of Network and Computer Applications,2020;152:102526.
5. Patil S, Sharma V. Voice-Controlled Home Automation via Android Application and Bluetooth. International Journal of Innovative Research in Computer Science and Technology,2021;9(3):88-95.
6. Gupta R, Mehta D, Chaudhary S. NLP-Enhanced Command Processing for Smart Home Systems Using ESP32. Smart Cities and IoT Research,2022;4(1):33-47.
7. Zhang Y, Li H. AI-Assisted Intent Recognition for Voice-Driven Home Automation. Sensors,2023;23(8):3914.
8. Arduino. Arduino Official Documentation. Accessed March 2026. <https://www.arduino.cc>
9. Banzi M. Getting Started with Arduino. Sebastopol, CA: Maker Media, Inc.
10. Monk S. Programming Arduino: Getting Started with Sketches. New York: McGraw-Hill Education.