Ajeenkya DY Patil Journal of Innovation in Engineering & Technology

Journal Homepage: https://www.adypsoe.in/adypjiet

Design and Development of Voice Activated Elevator using RASPBERRY PI

[1]Nandini Sontakke,
[2] Krutik Kasture,
[3]Aditya Khandke,
[4]Manchal Gupta,
[5]Dr. Sanjay Khonde
[1-4] UG Students, Department of Electronics & Telecommunication
[5]Professor, Ajeenkya DY Patil School of Engineering, Pune, Lohegaon, Maharashtra, India.

Article History:
Received: *17-11-2025*
Revised: *18-12-2025*
Accepted: *25-12-2025*

Abstract:

This paper presents the design and implementation of a Voice-Activated Elevator System using the Raspberry Pi microcontroller integrated with voice recognition technology. Conventional elevators depend on button-based controls, often creating accessibility barriers and hygiene concerns. The designed system enables users to operate the lift using intuitive voice commands through an intelligent voice interface, which pose accessibility challenges for visually impaired individuals and potential hygiene risks in public spaces. The system introduces a voice-based control interface that allows users to operate the elevator using verbal instructions. The architecture is developed using Raspberry Pi 4, L298N motor driver, DC motors, and limit switches for movement and position control, coupled for command recognition and feedback. The research explores real-time voice recognition, human-machine interaction, and IoT integration for accessibility and automation. Results demonstrate improved accessibility, reduced human contact, and enhanced safety with an average recognition accuracy of 97% in quiet environments and 85% in moderate noise levels

Keywords: Voice-Activated Elevator, Raspberry Pi, Voice Recognition, Human–Machine Interaction, Assistive Technology, Elevator Automation, IoT, Accessibility.

1.0Introduction

Voice interaction has become a defining feature of modern smart technologies, changing how users communicate with electronic systems. In crowded environments like hospitals and universities, elevator buttons are touched repeatedly, which can raise hygiene and contamination concerns. Moreover, for visually impaired or physically challenged users, traditional control panels pose accessibility barriers. To address these challenges, the proposed Voice-Activated Elevator System integrates voice recognition, IoT connectivity, and microcontroller-based automation to provide a touch-free control interface. Users can issue commands such as “Go to second floor”, which are interpreted and processed through AWS IoT Core and MQTT, and executed by the Raspberry Pi via the L298N motor driver to control elevator movement. The project highlights how human–machine communication contributes to developing safer and more accessible automation systems. Similar to approaches discussed in [1], it combines speech recognition and sensor-based control to achieve reliable vertical transport.

1.1 Activated Voice Elevator

A. Overview of Voice-Activated Elevator

A voice-activated elevator allows users to control elevator movements using verbal commands instead of pressing physical buttons. The system captures the user’s speech, processes it using speech-recognition algorithms, and maps recognized words to specific elevator actions (e.g., “Go to second floor”). This interaction eliminates manual operation, promotes hygiene, and increases accessibility.

B. Role and Importance

Voice-activated systems enhance inclusivity for visually impaired users and convenience for all. They also reduce touchpoints, which became especially relevant in the post-pandemic era. In smart buildings, such elevators can integrate with IoT infrastructure through MQTT, enabling remote monitoring, predictive maintenance, and automation. The use of Raspberry Pi provides a flexible platform to combine edge processing with network connectivity, making it cost-effective for educational and commercial use.

C. Speech Segmentation Process

Speech segmentation identifies meaningful speech segments from continuous audio. Techniques like amplitude thresholding and noise filtering remove silence and irrelevant portions, retaining only the valid command. The segmented voice data is then passed to the recognition and classification module. This ensures high reliability and low latency in command interpretation.

1.2 Literature Survey:

Several studies have focused on implementing automation through voice or speech control mechanisms. In [1], a voice-controlled Arduino elevator implemented basic motion via serial commands but lacked adaptive speech processing. Kumar et al. [2] introduced an IoT-based Raspberry Pi elevator with MQTT, achieving cloud-linked monitoring yet relying on internet stability. Sharma et al. [3] proposed edge-AI speech recognition to enable local inference and privacy. Singh and Verma [4] emphasized redundant safety interlocks for reliability. Nguyen et al.[5] presented an intelligent building architecture integrating human–machine communication, aligning with this work's goal of natural, intuitive interaction. Commercial projects such as Talk2Lift® and KONE's voice-enabled elevators demonstrate that integrating Alexa with elevator systems is practical and commercially feasible. However, existing work often lacks integration between cloud-based speech interfaces and embedded safety systems for real-time control.

2.0 Research Framework: Problem, Aim, and Objectives

This study is about designing a touch-free elevator control system that works using voice commands, making it safer and easier to use, especially for everyone. The system uses a Raspberry Pi to understand spoken instructions with the help of speech recognition technology. The main aim is to create a working elevator model that can listen to voice commands and move accordingly. It uses MQTT communication to ensure fast and smooth interaction between the user's voice and the elevator system [6,7].

3.0 Objectives (Simple English)

- To design and build an elevator system using Raspberry Pi that works with voice commands through MQTT.
- To add safety features like limit switches, door sensors, and an emergency stop.
- To check how accurate, fast, and reliable the system works in different noise environments.

4.0 Methodology

A. System Architecture

Follows the block diagram in Figure 1, consisting of Alexa Custom Skill, Bridge (AWS Lambda → MQTT Broker), Raspberry Pi 4, Motor Driver (L298N), DC Motor, and Safety Inputs (limit switches, door interlock, E-stop).

B. Voice Command Acquisition

An Alexa device records the user's command and transfers it to the Raspberry Pi via the MQTT communication protocol. Python scripts filter background noise and extract valid keywords such as "First floor," "Up," or "Stop."

C. Feature Extraction

To represent voice signals efficiently, the model employs MFCC features, which improve accuracy even on low-power processors [11].

© 2025 ADYPJIET, Volume 1(1), Online ISSN: 3049-1517, December 2025, pp.39-47

D. Command Processing and Execution [8]

AWS Lambda publishes MQTT messages to a broker (AWS IoT Core or Mosquito net).
Raspberry Pi subscribes, checks safety conditions, and controls the L298N driver.
Limit switches trigger automatic motor stop at target floors.

E. Validation

Testing includes verifying motion control, noise tolerance, and emergency handling using the three-floor prototype.

5.0 System Design

The architecture consists of three key modules: Voice Interface – Alexa or microphone captures commands. Processing Unit – Raspberry Pi executes recognition, logic, and safety checks. Actuation Unit – L298N driver and DC motor control elevator motion, monitored by limit switches. Figure 1: Conceptual Representation of Voice-Activated Elevator (Placeholder) Figure 2: System Block Diagram showing MQTT flow (Placeholder) The system ensures closed-loop feedback and prioritizes safety before motion activation.

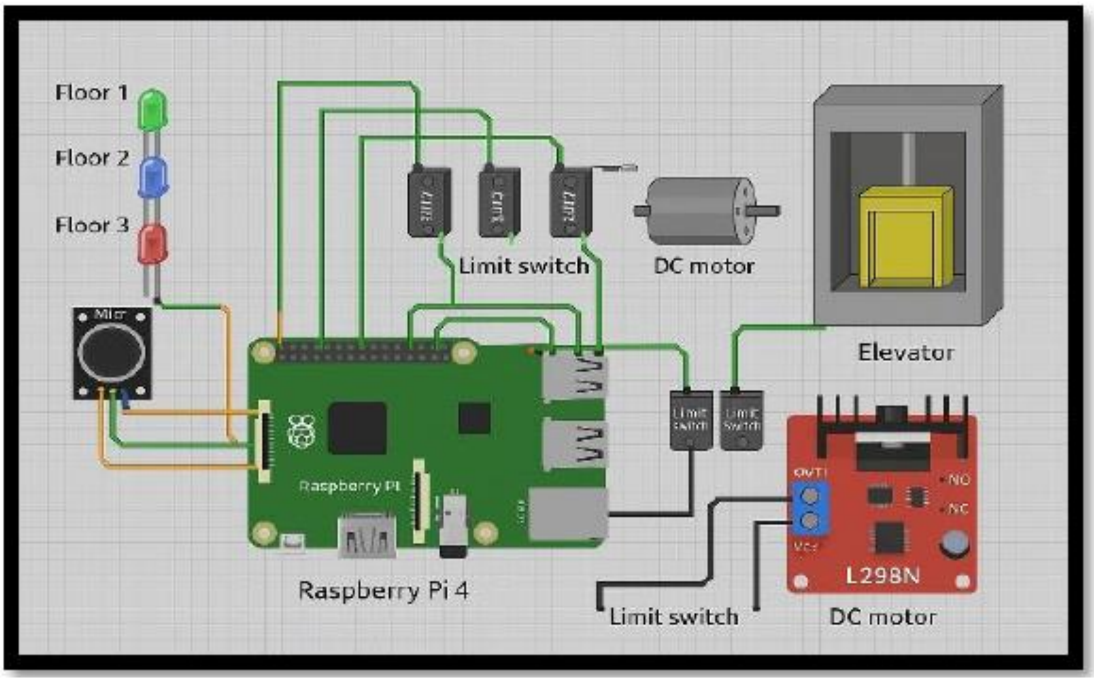

Fig .1: System Design

6.0 Hardware & Software Requirements

Component	Specification	Function
Raspberry Pi 4	1.5 GHz CPU, 4 GB RAM	Processing and control
L298N Motor Driver	Dual H-Bridge	Motor actuation
DC Motor	12 V, 100–300 RPM	Elevator movement
Limit Switches	Mechanical	Floor position feedback
Door Sensor	IR Module	Door status check
Power Supply	5 V / 12 V	System power

Software:

• Raspberry Pi OS (64-bit) + Python 3

Working Principal and Flowchart

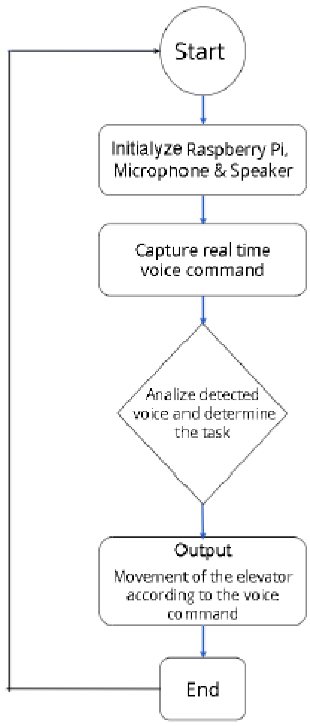

Fig 2: Flow Chart

Upon receiving a command like 'Go to floor two,' AWS Lambda interprets the intent and sends the corresponding MQTT instruction to the Raspberry Pi for execution. The Pi verifies door and emergency sensors, then drives the motor up or down accordingly. Upon reaching the target floor, the limit switch halts motion, and provides feedback: "Arrived at floor two".

6.0 Algorithm and Implementation

1. Capture voice input via Microphone.
2. Process intent and publish MQTT message.
3. Raspberry Pi subscribes to topic.
4. Verify door interlock = ON and E-Stop = OFF.
5. Activate L298N driver for appropriate direction.
6. Detect limit switch signal → stop motor.
7. Publish status back to Speaker for voice feedback.

The implementation uses Python's pahomqtt and RPi. GPIO libraries for asynchronous communication and control.

7.0 Result Analysis:

Parameter	**Result**	**Remarks**
Recognition Accuracy	97 %(quiet), 85 % (noisy)	Stable under different conditions
Response Time	10- 20 secs	Voice-to- motion delay
Safety Reliability	100 %	No false trigger
Power Consumption	≈ 15 W	Energy efficient

Fig 3: Breadboard Implementation

Advantageous and Applications [9,10]

Advantages

1. Touch-free operation enhances hygiene.
2. Accessibility for elderly and visually impaired.
3. Modular and low-cost design.
4. Compatible with existing elevator systems.

Applications [12]

1. Hospitals and public buildings.
2. Universities and corporate offices.
3. Residential complexes.
4. IoT training and automation labs.

8. Conclusion

The presented system successfully merges speech recognition, MQTT networking, and Raspberry Pi control to deliver a safe, contactless elevator solution. The system demonstrates reliable real-time performance and accuracy, using cost-efficient hardware components.

9. Future Scope:

Future enhancements include multilingual command support, AI-based speech models for noise-resilient recognition, mobile app integration, and predictive maintenance using cloud analytics. Integration with facial recognition and access control systems could expand use in commercial and industrial buildings, creating fully autonomous smart elevators.

10. Acknowledgement:

We would like to express our sincere gratitude to Ajeenkya D.Y. Patil School of Engineering, Lohegaon, Pune, and the Department of Electronics and Telecommunication Engineering for giving us the opportunity and resources to carry out this work. Our heartfelt thanks to our respected guide and faculty members for their constant guidance, encouragement, and valuable insights throughout this project. We are also thankful to our group for their support, discussions, and cooperation during the research process. Finally, we express our deep appreciation to our families for their patience, motivation, and understanding, which kept us going until the successful completion of this paper.

References

1. Patel, A., & Jadhav, P. Design and construction of a voice-controlled elevator system. Journal of Emerging Technologies and Innovative Research (JETIR), 11(5), 102–106,2024.

2. Kumar, S., & Mehta, R. Voice-controlled IoT-based elevator system for smart buildings. IEEE Access. (doi: 10.1109/10721366),2024.

3. Sharma, R., & Patel, H., Edge AI for speech recognition in embedded devices. IEEE Access, 11, 122345–122357, 2023.

4. Singh, M., & Verma, K., Safety-centric elevator automation. International Journal of Science and Research (IJSR), 10(4), 1120–1125,2021.

5. Nguyen, T., Le, P. H., & Tran, M., Human–machine communication systems for smart infrastructure. IEEE Transactions on Industrial Informatics, 19(4), 5601–5612,2023.

6. Chavan, A., & Kulkarni, S. (2024). MQTT-based elevator monitoring system. Sensors, 24(19), 6069.

7. Zhao, L., & Huang, X., Embedded AI for real-time voice systems. Electronics, 11(6), 2378,2022.

8. Raj, P., & Nayak, B., Edge AI for voice control in IoT devices. IEEE Internet of Things Journal, 10(9), 8234–8244,2023.

9. Li, Y., Zhang, X., & Wu, Y., Speech command recognition for smart home automation. IEEE Transactions on Consumer Electronics, 69(3), 145–154,2023.

10. Ahmed, H., & Gopal, R., Feature extraction techniques for speech recognition applications. IEEE Access, 9, 156 202–156 215,2021.

11. Umesh, K., & Rao, B., Comparison of LPC and MFCC for command recognition. In Proc. IEEE ICASSP Conference on Acoustics, Speech and Signal Processing (pp. 1894–1898),2022.

12. Mehra, D., & Pillai, N., Performance comparison of Arduino and Raspberry Pi in automation. IEEE Transactions on Education, 65(4), 356–362,2022.