STM JOURNALS

Journal of
VLSI Design Tools & Technology
https://journals.stmjournals.com/jovdtt

ISSN: 2249-474X (Online)
ISSN: 2321-6492 (Print)
Volume 15, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoVDTT

Research JoVDTT

Voice Control Music System Using Raspberry Pi

Prakash Kumar[1], Deep Satish Jedhe[2,*], Vijay H. Hovel[2], Omkar S. Sindhe[2]

Abstract

This paper presents a voice-controlled music system using Raspberry Pi, designed for hands-free music playback through voice commands. The system is implemented using Raspberry Pi 4B, a microphone, a speaker, and a 32GB SD card. The software is developed in Python using Thonny IDE, leveraging libraries such as SpeechRecognition, PyAudio, and gTTS for speech processing and audio playback. The system recognizes user commands like play, pause, next, and stop, converting speech input into text and executing corresponding actions. The Raspberry Pi serves as the main controller, interpreting voice commands via a microphone and playing music through connected speakers. It supports both locally stored files and streaming services, creating a smooth and interactive audio experience. This setup provides users with convenient, hands-free control over their music, enhancing accessibility and user engagement in smart audio systems. Performance evaluation includes voice recognition accuracy, response time, and real-time execution efficiency. Future enhancements include multilanguage support, noise reduction, and smart assistant integration. This work contributes to advancing voice-based Internet of Things (IoT) applications in entertainment and accessibility solutions.

Keywords: Voice recognition, Raspberry Pi 4B, speech-to-text processing, hands-free music control, IoT-based audio system

INTRODUCTION

Hands-free control solutions have grown in popularity with speech recognition, and the Internet of Things (IoT) has advanced rapidly. Particularly in situations such as driving, smart home management, or industrial automation, these technologies enable people to communicate with equipment using voice commands, resulting in increased convenience and safety. By eliminating the need for manual input, voice-controlled devices enable users to multitask effectively and remain focused on their core duties. Voice recognition is essential for establishing smooth human-machine interactions as IoT becomes more pervasive in daily life. This combination enables smart devices to perform tasks autonomously or respond instantly to spoken instructions [1–3]. The growing demand for accessible, user-friendly technology is driving further innovation in this field. Consequently, hands-free systems are now being adopted across various industries, including healthcare, automotive, and consumer electronics, making everyday tasks easier and more intuitive. The synergy between IoT and voice recognition continues to reshape modern user experience. This paper presents a voice-controlled music system using Raspberry Pi, designed to enhance user experience by enabling music playback through voice commands. Traditional music systems require manual interactions, which may not always be convenient. This project eliminates the need for physical control, allowing users to operate the system using simple voice instructions such as play, pause, next, and stop [4–6].

The system is built using Raspberry Pi 4B as the central processing unit, interfaced with a

***Author for Correspondence**
Deep Satish Jedhe
E-mail: deepjedhe25@gmail.com

[1]Assistant Professor, Department of Electronics Telecommunication Engineering, Shri Chhatrapati Shivajiraje College of Engineering, Maharashtra, India
[2]Student, Department of Electronics Telecommunication Engineering Shri Chhatrapati Shivajiraje College of Engineering, Maharashtra, India

Received Date: April 17, 2025
Accepted Date: May 08, 2025
Published Date: August 21, 2025

Citation: Prakash Kumar, Deep Satish Jedhe, Vijay H. Hovel, Omkar S. Sindhe. Voice Control Music System Using Raspberry Pi. Journal of VLSI Design Tools & Technology. 2025; 15(2): 34–39p.

microphone for voice input and a speaker for audio output. The software was developed in Python using the Thonny IDE, integrating libraries, such as SpeechRecognition, PyAudio, and gTTS, to process voice commands and control music playback. The system supports both local music files and streaming services and offers flexible and interactive experiences.

This study explored the hardware setup, software implementation, speech recognition accuracy, and system performance. Future improvements include multilingual support, noise filtering, and integration with smart home assistants. The proposed system serves as a foundation for developing AI-powered voice-controlled entertainment solutions, contributing to the growing field of IoT-based automation.

LITERATURE SURVEY

The integration of voice recognition technology into smart systems has gained widespread adoption in recent years. Numerous studies have explored the role of speech processing, Raspberry Pi applications, and automated music playback in IoT-based environments. This section reviews the key research contributions of voice-controlled music systems.

Voice recognition technology is a crucial component of hands-free automation. Early studies introduced the Hidden Markov Model (HMM) for speech recognition, which became the foundation of modern voice processing systems. Recent advancements in deep learning based speech recognition, such as Google Speech API and PocketSphinx, have improved the accuracy and response time, making real-time speech-to-text conversion more efficient [1].

Studies have explored the use of neural networks for speech recognition, leading to the development of voice assistants, such as Google Assistant and Amazon Alexa. However, lightweight solutions such as PocketSphinx and VOSK are preferred for offline and low-power applications, making them suitable for Raspberry Pi based projects [2].

In IoT and automation, Raspberry Pi has emerged as a popular platform for smart home automation and embedded systems due to its affordability, compact size, and extensive support for programming. A previous study demonstrated the use of Raspberry Pi as a central hub for IoT applications, including voice-controlled operations for household devices [3].

Similarly, a study developed a voice-controlled home automation system using Raspberry Pi and Google Assistant, proving its capability for real-time speech processing. This study validates the feasibility of using Raspberry Pi for voice-activated control, aligning with the objectives of this project [4].

Music playback systems on embedded platforms require efficient audio processing algorithms. Studies have analyzed the implementation of Music Player Daemon (MPD) and VLC Media Player on low-power devices, highlighting their suitability for Raspberry Pi based systems [5]. A study explored voice-controlled media playback using Raspberry Pi with Python-based speech recognition, enabling hands-free control over audio functions, such as play, pause, and skip [6].

These studies formed the basis for implementing voice-controlled music playback in this project. By combining speech recognition, Python programming, and IoT-based control, this system offers an interactive and user-friendly approach for music automation.

SYSTEM MODEL

The voice-controlled music system utilizes a Raspberry Pi 4B as the core processing unit to execute voice-based commands for music playback. The system is structured into three main modules.

- *Voice input module:* Captures the user’s voice using a microphone and converts it into text using speech recognition and PyAudio.

- *Processing module:* Raspberry Pi processes the recognized command and maps it to predefined actions, such as play, pause, next, or stop.
- *Audio output module*: Plays music through a speaker using VLC, Pygame, or MPD, supporting both local and streaming playback.

The system was implemented in Python using Thonny IDE and integrated libraries such as SpeechRecognition, gTTS, and Spotipy for enhanced functionality.

The workflow follows these steps:

1. User speaks a command (e.g., "play music").
2. The microphone captures the voice and sends it to the Raspberry Pi.
3. SpeechRecognition processes the input and converts it into text.
4. The system matches the command and executes the corresponding function.
5. Music playback is controlled accordingly via connected speakers.
6. If the command is unrecognized, the system provides an error message or asks for repetition.
7. The system can retrieve and play specific songs, albums, or playlists based on voice input.
8. A noise filtering mechanism ensures accurate command recognition in noisy environments.
9. The user can adjust the volume using voice commands such as "increase volume" or "decrease volume."
10. The system logs command history to improve accuracy and responsiveness over time.

This model ensures real-time execution, hands-free control, and seamless music playback while offering an intuitive and interactive experience. Additionally, the system is designed to be lightweight and power-efficient, making it ideal for continuous operation [7]. Voice command flexibility enables users to interact naturally with a system without predefined keywords. Future updates will focus on improving response time and adding support for additional streaming services. The implementation of machine learning can further enhance speech recognition accuracy and ensure adaptability to different accents and languages. Figure 1 shows the block diagram of the model.

METHODOLOGY

A voice-controlled music system was developed using a Raspberry Pi 4B and Python-based voice recognition to execute music playback commands. The methodology consists of hardware setup, software implementation, and system workflow, ensuring efficient and real-time execution.

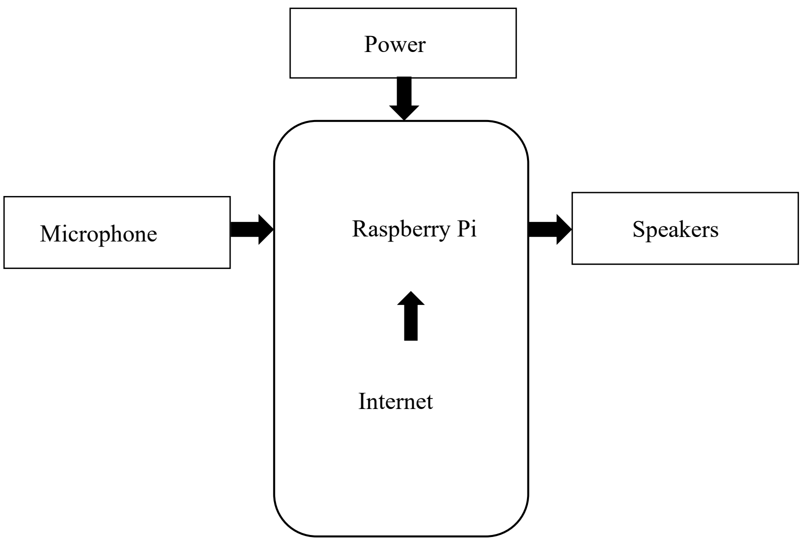

Figure 1. Block diagram description.

System Overview

The system followed a structured approach to process voice commands and control music playback. The implementation was divided into three major phases.

1. *Hardware setup*—Connecting the Raspberry Pi with a microphone, speaker, and storage.
2. *Software development*—Writing Python scripts to handle speech recognition and music playback.
3. *System workflow*—Capturing the user's voice, processing commands, and playing music.

Hardware Setup

The following components are used to build the system (Figure 2).

- *Raspberry Pi 4B*—Central processing unit for executing commands.
- *Microphone*—Captures user voice input.
- *Speaker*—Outputs audio based on commands.
- *32GB SD Card*—Stores the operating system and required libraries.
- *Power Supply*—Ensures stable power to Raspberry Pi and peripherals.

Software Implementation

The software is developed using Python in the Thonny IDE, integrating into the following libraries:

- *SpeechRecognition and PyAudio*—For speech-to-text processing.
- *gTTS (Google Text-to-Speech)*—Provides voice feedback.
- *VLC and Pygame*—For music playback.

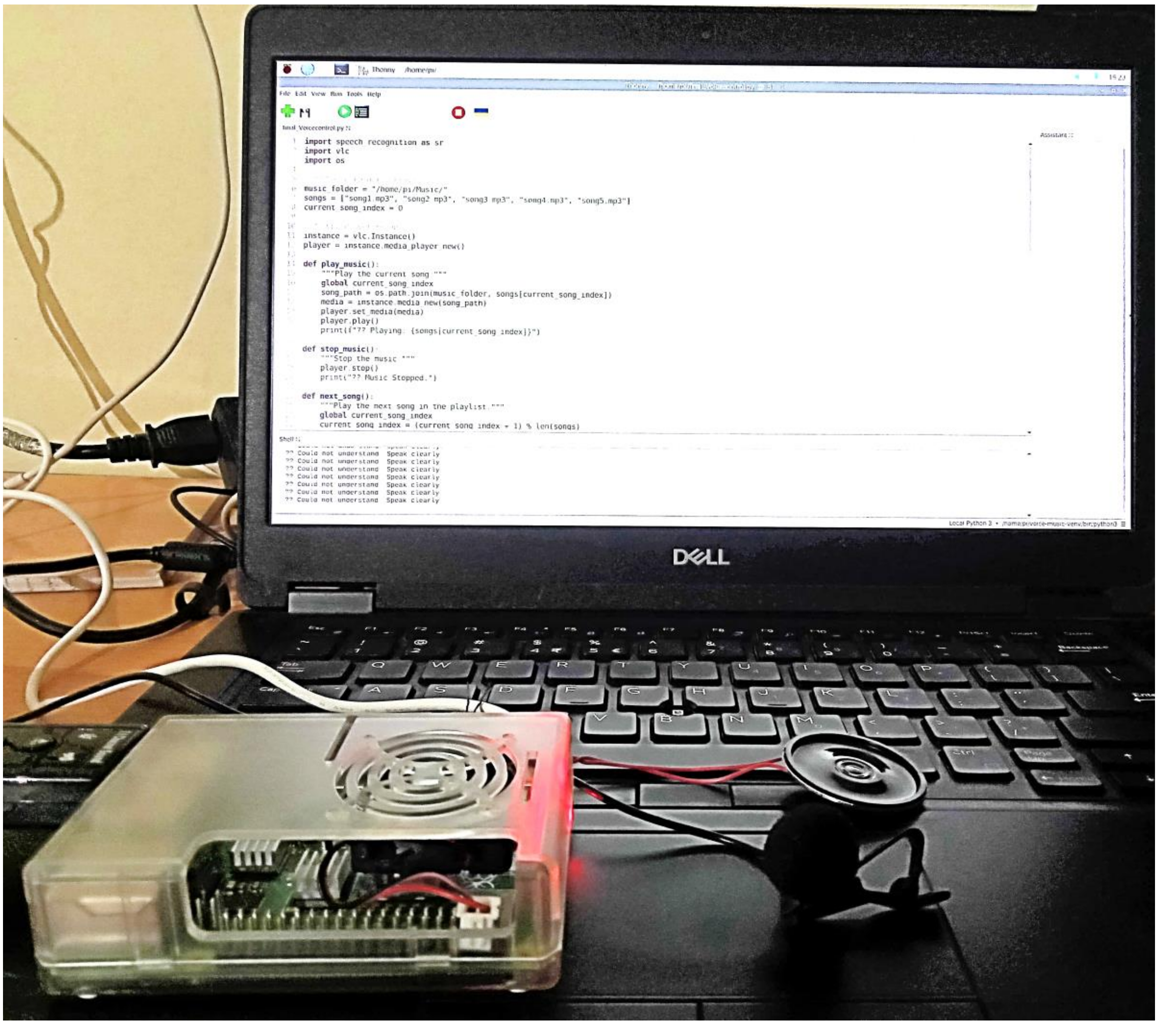

Figure 2. Hardware of a voice-controlled music system using Raspberry Pi.

System Workflow

1. The microphone captures the user's voice.
2. The SpeechRecognition module processes the voice input and converts it into text.
3. The command is analyzed to determine the appropriate action (play, pause, next, or stop).
4. If recognized, the Python script executes the corresponding function using VLC or MPD.
5. Music is played through the connected speaker.
6. If an invalid command is given, an error message is returned.
7. The system logs command history to improve performance over time.

This methodology ensures a structured approach for designing, implementing, and testing a voice-controlled music system, providing seamless user interaction and real-time functionality.

FUTURE SCOPE

The voice-controlled music system has significant potential for future enhancement and broader applications. The following points highlight the scope of this project:

- *Multilanguage support:* The system can be improved by integrating multiple language models to recognize and process voice commands in different languages, making it accessible to a global audience [8].
- *Noise reduction and accuracy improvement:* Advanced noise filtering techniques, such as deep learning based denoising models, can enhance the accuracy of speech recognition, particularly in noisy environments.
- *Integration with smart assistants:* The system can be extended to work with AI-based voice assistants, such as Amazon Alexa, Google Assistant, and Apple Siri, allowing seamless interaction with smart home devices.
- *Gesture-based controls:* Combining voice commands with hand gesture recognition using computer vision can provide an alternative control mechanism, enhancing accessibility for users with speech impairments.
- *Cloud-based music recommendations:* Implementing machine learning algorithms to analyze user preferences and suggest personalized music playlists based on past listening history and mood detection [9].
- *Offline mode enhancement:* Developing a fully offline mode with pre-trained voice models can make the system functional without an Internet connection, thereby ensuring reliability in remote areas.
- *Voiceprint authentication:* Adding a security layer by implementing voiceprint recognition, ensuring that only authorized users can control the system.
- *Hardware optimization:* Exploring more energy-efficient hardware components to reduce power consumption and enhance portability for battery-operated or mobile applications [10].

With these advancements, the system can evolve into a more intelligent, secure, and user-friendly solution, expanding its applications in the smart homes, vehicles, and entertainment industries.

CONCLUSION

The development of a voice-controlled music system using Raspberry Pi demonstrates the potential of voice recognition technology to enhance user interaction with multimedia systems. By integrating SpeechRecognition, PyAudio, GTTS, and music playback libraries, the system enables seamless hands-free control, offering users a convenient and accessible way to listen to music. The project successfully implemented real-time speech processing, where commands such as *play, pause, next, and stop* were recognized and executed efficiently. The use of the Raspberry Pi 4B as the central processing unit ensures an affordable and compact solution for voice-controlled applications. Performance evaluation shows that the system functions effectively under normal conditions; however, improvements in noise reduction and multilanguage support can further enhance accuracy and usability. The proposed enhancements, such as integration with smart assistants, gesture controls, and personalized music recommendations, provide opportunities for future advancements. This system is not only applicable for personal entertainment, but also extends to smart homes, automotive applications, and assistive technologies for individuals with disabilities. With continuous developments in artificial intelligence and IoT, this project serves as a foundation for future innovations in voice-controlled multimedia systems. Thus, this work contributes to the growing field of human–computer interaction (HCI) by providing an intuitive, hands-free, and efficient music-control solution using voice commands.

REFERENCES

1. Seth A, Ray N, Paul A, Bag P, Sarkar S, Das KS, et al. Raspberry Pi controlled automated residential system with integrated Alexa Voice Service. 2024 IEEE 1st International Conference on Green Industrial Electronics and Sustainable Technologies (GIEST), Imphal, India. 2024. p. 1–6. doi:10.1109/GIEST62955.2024.10960058.
2. Irugalbandara C, Naseem AS, Perera S, Kiruthikan S, Logeeshan V. A secure and smart home automation system with speech recognition and power measurement capabilities. Sensors. 2023;23:5784. doi:10.3390/s23135784. PMID: 37447635.
3. Groenli TM, Flesch B, Mukkamala R, Vatrapu R, Klavestad S, Bergner H. Internet of things big data analytics: The case of noise level measurements at the Roskilde music festival. 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA. 2018. p. 5153–8. doi:10.1109/BigData.2018.8622406.
4. Revathi A, Sasikaladevi N, Arunprasanth D, Raju N. Raspberry Pi-based robust speech command recognition for normal and hearing-impaired (HI). Multimed Tools Appl. 2024;83:51589–613. doi:10.1007/s11042-023-17543-8.
5. Prasanna G, Ramadass N. Low cost home automation using offline speech recognition. Int J Signal Process Syst. 2014;2:96–101.
6. Marot J, Bourennane S. Raspberry Pi for image processing education. 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece. 2017. p. 2364–2366. doi:10.23919/EUSIPCO.2017.8081633.
7. Kasundra CT, Shirsat AS. Raspberry-Pi based health monitoring system. Int J Adv Res Electr Electron Instrum Eng. 2015;4:7147–54. doi: 10.15662/ijareeie.2015.0408083.
8. Boruah AN, Goswami M, Kumar M, Loyola-González O, editors. Embedded Artificial Intelligence: Real-Life Applications and Case Studies. Boca Raton, Fla., USA: Chapman and Hall/CRC; 2025. doi:10.1201/9781003481089.
9. Mi Suresh S, Rao YS. Modelling of secured voice recognition based automatic control system. Int J Emerg Technol Comput Sci Electron. 2015;13:140–4.
10. Washani N, Sharma S. Speech recognition system: A review. Int J Comput Appl. 2015;115:7–10. doi:10.5120/20249-2617.

