Development of an Open-Source Voice-Controlled Smart Home System

Olja Krčadinac[1], Željko Stanković[2], Dragana Dudić[3], Lazar Stošić[4]
[1]"Union – Nikola Tesla" University, Faculty of Informatics and Computer science, Belgrade, Serbia, okrcadinac@unionnikolatesla.edu.rs, 0000-0002-6299-371X
[2]Pan-European APEIRON University, Banja Luka, B&H, Vojvode Pere Krece 13, stanz@medianis.net, 0000-0002-9893-9088
[3]"Union – Nikola Tesla" University, Faculty of Informatics and Computer science, Belgrade, Serbia, ddudic@unionnikolatesla.edu.rs, 0000-0001-8513-6529
[4]"Union – Nikola Tesla" University, Faculty of Informatics and Computer science, Belgrade, Serbia, lstosic@unt.edu.rs, 0000-0003-0039-7370

Original scientific paper
https://doi.org/10.7251/JIT2402111K UDC: 621.391:004.414.23

Abstract: This paper explores the design and implementation of a voice-controlled smart home system utilizing the Raspberry Pi platform and Node.js framework. The system aims to provide users with an intuitive method for managing household devices through voice commands while enhancing functionality with integrated sensors. Leveraging the Hidden Markov Model Toolkit (HTK) for speech recognition, the system accurately interprets user commands, facilitating control over lighting, temperature, and various IoT devices. In addition to voice activation, the system incorporates multiple sensors: a temperature sensor for monitoring ambient conditions, a motion sensor (PIR) for detecting occupancy, and a light sensor for assessing natural light levels. These sensors work in harmony with the voice control features, enabling automated responses such as adjusting the heating or cooling based on temperature readings, activating lights upon detecting movement, and regulating artificial lighting according to available daylight. Comprehensive testing demonstrated the system's high accuracy in command recognition and its responsiveness to user inputs, as well as its robust integration with additional smart devices. User feedback was instrumental in refining the system, leading to improvements in command clarity and operational efficiency. This research highlights the potential of combining voice control and sensor technology to create a more responsive and user-friendly smart home environment.

Keywords: IoT, Smart home systems, voice recognitions, sensors, open-source

Introduction

In recent years, smart home systems have gained significant popularity, driven by advancements in Internet of Things (IoT) technology [1] and artificial intelligence (AI) [2]. Voice-controlled systems, in particular, have revolutionized the way users interact with devices, providing hands-free, intuitive control over various household functions [3]. Proprietary solutions like Amazon Echo and Apple HomeKit have dominated the market, offering integrated services that facilitate voice control over smart home environments [4] [5]. However, these solutions are limited by their commercial nature, which restricts customization, scalability, and user control over data privacy.

To address these limitations, we propose a fully open-source, affordable voice-controlled smart home system that provides flexibility, transparency, and ease of customization. Our system is designed to operate on widely available, cost-effective hardware, enabling users to create personalized smart home environments without the constraints of proprietary platforms. By combining open-source tools such as the HTK Tool for speech and speaker recognition with Node.js as the software framework, this system not only provides core functionalities but also offers modular extensibility for further community-driven development. The HTK (Hidden Markov Model Toolkit) was selected for its robust capabilities in speech and speaker recognition, offering a proven, adaptable framework well-suited to accurately interpreting voice commands in diverse home environments [6]. Node.js was chosen for its efficient, event-driven architecture, making it ideal for managing real-time data from multiple sensors and processing voice commands seamlessly, which are essential for responsive smart home control [7].

Our choice of the Raspberry Pi as a hardware platform ensures that the system remains affordable [8], accessible [9], and easy to expand, while integrated text-to-speech (TTS) [10] capabilities facilitate natural user interaction. With voice control predicted to play a crucial role in the future of IoT, this project contributes to the field by providing a foundation for open-source smart home systems that encourage innovation and user involvement [11].

This paper presents the design, implementation, and potential applications of our voice-controlled smart home system, aiming to establish a scalable framework that can be utilized and extended by a broad user base.

Methods and materials

Our open-source voice-controlled smart home system was developed using affordable hardware and readily available software tools, ensuring ease of configuration, expandability, and cost-effectiveness. Below is a table (table 1) of the core components and methods applied in the system's development:

Table 1. *Core Components and Methods for the Development of an Open-Source Voice-Controlled Smart Home System*

	Component	Description
Hardware components	*Raspberry Pi*	Model 3, chosen for affordability, versatility, and support for IoT applications; serves as the primary processing unit, handling audio processing and integration with sensors/actuators.
	Microphone and Speakers	USB microphone and speakers connected to the Raspberry Pi for capturing voice commands and providing audio feedback to the user.
	Additional Sensors & Actuators	Integrated with various sensors (temperature, motion, light) and actuators (smart lights, plugs) for expanded smart home control.
Software Components	*HTK (Hidden Markov Model Toolkit)*	Used for speech and speaker recognition; robust algorithm for interpreting voice commands accurately. Commands are pre-defined and trained in HTK to optimize accuracy.
	Node.js Framework	Serves as the primary software platform, processing commands, handling sensor data, and managing device states; includes custom modules to interpret HTK output and process commands.
	Text-to-Speech (TTS) Module	Integrated within Node.js to enable verbal responses for user interaction, status updates, and answering questions.
System Architecture	*Input Processing*	Microphone captures user commands, which are processed by HTK to identify command content and speaker identity (when multiple users are set up).
	Command Interpretation	Node.js matches interpreted commands with pre-defined functions, triggering appropriate responses like turning on lights or adjusting temperature.
	User Feedback	TTS module synthesizes responses for playback through speakers, confirming command execution or providing additional info.
Evaluation and Testing	*Testing Approach*	Tested for command accuracy, response time, and ease of integration with additional devices. Repeated commands in varied conditions (e.g., background noise, different voices) to assess HTK recognition accuracy and robustness.
	User Feedback	Collected from user trials to refine the command set and enhance response clarity.

For this research, we implemented a voice-controlled smart home system in five selected households, utilizing Raspberry Pi as the central processing unit. The process involved developing custom modules in Node.js, configuring affordable hardware, and integrating low-cost sensors and actuators. Below are the key steps:

1. Selection of Households

Five households were chosen based on diverse family structures and varying levels of familiarity with technology. This ensured a representative sample for evaluating user experiences.

2. Deployment of Raspberry Pi

The Raspberry Pi Model 3 was installed in each household due to its affordability, versatility, and compatibility with IoT applications. It served as the central unit for processing voice commands, integrating sensors, and controlling actuators.

Setup Procedure: The Raspberry Pi was pre-configured with the latest Raspbian OS to ensure compatibility with the software. Each device was connected to the household Wi-Fi network to enable seamless communication with sensors and actuators.

3. Software Components Installation

To enable system functionality, the following software components were installed:

Node.js Framework was installed using Node Version Manager (nvm) to manage software versions and host the custom application for processing commands and managing device states. HTK (Hidden Markov Model Toolkit) was configured for accurate speech recognition and optimized for predefined command sets.

4. Sensor and Actuator Integration

Sensors and actuators were installed to extend the system's functionality:

- Temperature Sensors: For monitoring ambient conditions and enabling automatic climate control.
- Motion Sensors: Positioned near entry points to trigger actions like turning on lights or sending alerts.
- Light Sensors: To detect natural light levels and adjust artificial lighting accordingly.
- Voice Control: USB microphones and speakers were installed for voice command input and audio feedback.

5. Testing and Calibration

After installation, the system was tested and calibrated for: Sensor accuracy and reliability under different environmental conditions;

Command recognition accuracy using HTK in scenarios with varying background noise;

Integration and response times between the Raspberry Pi, sensors, and actuators.

6. User Training and Feedback Collection

Households received training on system usage, followed by a feedback collection phase to identify usability issues and inform system refinements.

Proposed System Architecture

The proposed architecture for the voice-controlled smart home system is designed to offer a modular, flexible, and scalable solution for smart home automation (Figure 1). The system integrates various components that work together to provide seamless user interaction and control over home devices. The main elements of the system include the Raspberry Pi as the central unit, voice processing through the HTK (Hidden Markov Model Toolkit), and real-time control of smart devices using Node.js.

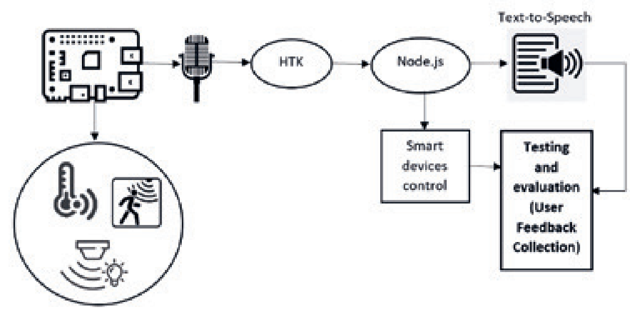


Figure 1. *System Architecture for Voice-Controlled Smart Home*

At the core of the system, the Raspberry Pi serves as the processing hub. It connects to a microphone to capture voice commands from the user. These audio signals are first processed through the HTK module, which performs speech recognition and speaker identification. The HTK system is responsible for converting the user's voice into text commands, allowing the system to interpret and act upon the user's requests. Once the voice command is processed, it is passed to Node.js, which serves as the platform for interpreting these commands and controlling smart devices. Node.js executes the necessary logic to map voice commands to corresponding actions, such as turning on lights, adjusting temperature, or activating other connected devices. The system integrates various smart devices, such as temperature sensors, motion detectors, and smart lights, which can be controlled based on voice inputs or environmental triggers.

To ensure interactive communication with the user, the system also includes a Text-to-Speech (TTS) module. After the system performs the requested actions, the TTS component provides feedback to the user, confirming the actions or providing additional information, such as the status of a device or an error message.

Additionally, the system includes a feedback loop for testing and evaluation. User feedback is gathered through interactions with the system, allowing for continuous improvement. This feedback helps refine the voice recognition and response accuracy, ensuring that the system evolves to meet user needs and preferences effectively.

This architecture provides a robust and user-friendly platform for controlling and automating smart home devices, allowing for easy customization and expansion. The use of open-source components and affordable hardware makes the system accessible while maintaining high scalability and performance. The integration of speech recognition, real-time control, and user feedback creates a system that is not only functional but also adaptable to a wide range of applications in smart home environments.

Results

The implementation and testing of the smart home system across five households yielded valuable insights into the system's functionality, user satisfaction, and overall performance. The results are presented across several key areas: sensor performance, system responsiveness, voice command accuracy, and user feedback.

Table 2. *Summary of Smart Home System Performance and User Feedback*

Category	Result	Observations
Temperature Sensors	Accuracy rate of ~95% in maintaining set temperature levels.	Users reported improved comfort due to automatic temperature adjustments.
Motion Sensors	High detection accuracy with occasional false triggers in households with pets.	Calibration adjustments helped reduce false triggers in households with frequent pet movement.
Light Sensors	Effective in gauging ambient light, enabling automatic lighting adjustments.	Users noticed energy savings, particularly in daylight hours.
System Responsiveness	Average response time of 1-2 seconds from command input to action execution.	Smooth and timely interaction for users; met expected responsiveness benchmark.
Node.js Processing	Efficient handling of multiple sensor data inputs and voice commands with no noticeable latency.	Node.js provided reliable real-time processing, enhancing the user experience.
HTK Voice Processing	Achieved ~92% accuracy in recognizing voice commands, with slightly lower accuracy in noisy environments.	Microphone sensitivity adjustments and HTK fine-tuning helped mitigate noise interference.
Multiple User Recognition	Successful in differentiating between multiple users' voices, enabling personalized responses.	Personalization allowed for tailored interactions based on individual users within each household.
User Satisfaction with Automation	High satisfaction with temperature and lighting automation features.	Users cited convenience and energy savings as major advantages of the system's automation.
Voice Interaction	Users found voice commands intuitive and enjoyed hands-free control.	Suggested expansion of recognized command set to improve flexibility.
Suggestions for Improvement	Users recommended better noise filtering and support for additional devices (e.g., smart locks, cameras).	Future updates could include enhanced noise filtering and expanded device integration for more comprehensive home automation.

The table 2 summarizes the effectiveness and user experience of the implemented smart home system. Overall, the system demonstrated reliable sensor accuracy, prompt responsiveness, and a high level of user satisfaction, particularly in automation features like temperature and lighting control. Minor challenges, such as occasional false triggers and voice recognition sensitivity in noisy environments, were addressed with system adjustments and calibrations. Users expressed an interest in further expanding the system's capabilities to include additional smart devices, indicating both the system's value in enhancing household convenience and opportunities for future improvements.

During the testing phase, two key types of evaluations were carried out: technical system testing and user feedback collection.

The system testing covered the accuracy of voice commands, response time, and integration with additional devices. Commands were tested under various conditions, such as background noise and different user voices, to assess the accuracy and robustness of speech recognition using HTK (Hidden Markov Model Toolkit). The response time of the system and its ability to interface with other smart home devices were also evaluated. Commands were repeated under varying conditions to ensure stable and accurate system performance in real-world usage scenarios.

User feedback was collected through trial runs of the system, allowing for refinement of the command set and improvement of response clarity. Users tested the system in everyday settings, using voice commands to control devices and interact with sensors. Based on their feedback, adjustments were made to the system's settings, and further optimizations were applied to improve accuracy and ease of use.

Testing showed that the system correctly recognizes voice commands in various conditions and efficiently controls smart devices. The user interface was rated as intuitive, and users expressed satisfaction with the system's response speed and recognition accuracy.

Discussion

The implementation of a smart home system across five households has offered significant insights into the capabilities and limitations of low-cost home automation solutions. Overall, the system proved reliable in managing basic tasks through components like the Raspberry Pi and sensors for temperature, motion, and light. The high accuracy rates of these sensors highlight the effectiveness of low-cost technology in providing comfort and energy efficiency in everyday household operations. However, minor challenges, such as false triggers in homes with pets, indicate that further adjustments and calibrations would improve sensor reliability, especially in households with unique environmental factors.

Voice recognition, facilitated by HTK and managed through Node.js, was a well-received feature, achieving high accuracy in quieter settings. Yet, background noise presented occasional difficulties in command recognition, particularly in homes near busy streets. Users appreciated the ease of hands-free control, but the need for further noise filtering became apparent, suggesting that enhanced audio processing could strengthen the system's usability across diverse household environments. Additionally, participants expressed a desire for a broader set of recognized commands, indicating that an expanded command repertoire would increase interaction flexibility and overall user satisfaction.

Feedback from users also revealed a high level of satisfaction with automated features, such as lighting and temperature adjustments. These functions not only added convenience but also contributed to energy savings, fulfilling one of the project's primary objectives. However, several users suggested integrating additional smart devices, such as security cameras and smart locks, to extend the system's functionality beyond home automation to include security and monitoring. This interest in added features points to a growing user expectation for more comprehensive smart home ecosystems.

This study's findings resonate with the global trend toward accessible, modular smart home technology. Using affordable components like the Raspberry Pi and basic sensors, the system demonstrated that effective home automation can be achieved on a budget, aligning with industry efforts to make smart home solutions more widely available without high costs.

Feedback on voice control limitations also reflects a broader industry focus on refining accuracy in diverse home environments. Major companies are enhancing noise filtering and multi-user recognition, addressing similar challenges faced in this study [12] [13].

Conclusion

This study demonstrates the feasibility and impact of implementing a low-cost smart home system in enhancing household comfort, energy efficiency, and convenience. The system, based on the Raspberry Pi and supported by various sensors, performed reliably in automating essential functions, such as temperature and lighting adjustments. Additionally, the integration of voice command functionality provided an intuitive, hands-free experience that was well-received by users, though some minor challenges with background noise highlighted areas for refinement in audio processing.

User feedback indicated high satisfaction with the automation features, especially regarding energy savings and ease of use. Suggestions for expanding the system's functionality to include additional devices, such as smart locks and security cameras, reflect the growing interest in comprehensive smart home ecosystems that go beyond basic automation.

In summary, this smart home system offers a promising solution for accessible and efficient home automation. Future enhancements, including improved noise filtering, broader command recognition, and compatibility with a wider range of smart devices, could further elevate its usability and appeal. With these refinements, the system has the potential to bring smart home technology into everyday use for a diverse range of households, promoting a more integrated, energy-conscious living environment.

References

[1] Aiman, F., Saquib, Z., & Nema, S. (2016). Hidden Markov Model system training using HTK. In 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) (pp. 806-809). IEEE.

[2] Almusaed, A., Yitmen, I., & Almssad, A. (2023). Enhancing smart home design with AI models: A case study of living spaces implementation review. Energies, 16(6).

[3] Bermúdez-Ortega, J., Besada-Portas, E., López-Orozco, J., Bonache-Seco, J., & De la Cruz, J. (2015). Remote web-based control laboratory for mobile devices based on EJsS, Raspberry Pi and Node. js. IFAC-PapersOnLine, 48(29), 158-163.

[4] Choi, W., Kim, J., Lee, S., & Park, E. (2021). Smart home and internet of things: A bibliometric study. Journal of Cleaner Production, 301.

[5] Iliev, Y., & Ilieva, G. (2022). A framework for smart home system with voice control using NLP methods. Electronics, 12(1), 116.

[6] Lakshmi, K. (2016). Design and Implementation of Text to Speech conversion using Raspberry pi. Dimensions, 85, 56.

[7] Liu, Y., Gan, Y., Song, Y., & Liu, J. (2021). hat influences the perceived trust of a voice-enabled smart home system: an empirical study. Sensors.

[8] Luria, M., Hoffman, G., & Zuckerman, O. (2017). Comparing social robot, screen and voice interfaces for smart-home control. In Proceedings of the 2017 CHI conference on human factors in computing systems , (pp. 580-628).

[9] Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of things hardware: performances and constraints. design issues, 3(8), 1-8.

[10] Shafei, H., & Tan, C. (2024). A Closer Look at Access Control in Multi-User Voice Systems. IEEE Access 12, 40933-40946.

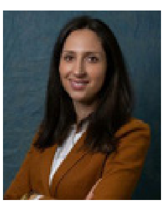
[11] Thapliyal, H., Ratajczak, N., Wendroth, O., & Labrado, C. (2018). Amazon echo enabled iot home security system for smart home environment. In 2018 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), 31-36.

[12] Valero, C., Pérez, J., Solera-Cotanilla, S., Vega-Barbas, M., Suarez-Tangil, G., Alvarez-Campana, M., & López, G. (2023). Analysis of security and data control in smart personal assistants from the user's perspective. Future Generation Computer Systems, 144, 12-23.

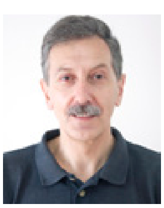
[13] Virtosu, I., & Chen, L. (2022). Bundling and tying in smart living. Smart Cities and Regional Development (SCRD) Journal, 6(2), 97-110.

Received: October 29, 2024
Accepted: November 19, 2024

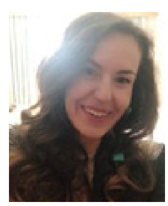
About the authors



Olja Krcadinac (Latinovic, maiden name) is assistant professor at "Union – Nikola Tesla" University - Faculty of Informatics and Computer Science. She earned her Ph.D. in biometric field from University of Belgrade – Faculty of Organizational science, where she conducted groundbreaking research on speaker recognition. In addition to her teaching responsibilities, Olja has authored numerous impactful publications in peer-reviewed journals, contributing valuable insights to the scientific community. Her research focuses on biometric, sensors, IoT and AI, addressing critical issues in AI and making significant contributions to the academic community.



Zeljko Stankovic received his higher education in Cleveland, Ohio, USA, where he graduated in 1981. The topic of the thesis was "Reversible sound in halls". He defended his master's thesis ("Learning control system (LMS) based on ADL SCORM specifications") in 2006 at the University of Novi Sad, Faculty of Science, Department of Informatics. He defended his doctoral dissertation (Laser perception of defined objects and encapsulation of control and logic elements for an autonomous robotic teaching tool) at Singidunum University, Belgrade, in 2010. He has been programming since 1984, creating programs for his first Commodore 64 computer. She works as a full-time professor at Pan-European University "APEIRON". Robotics and bioengineering have been a field of work and interest for many years. He is the holder of the patent right for the teaching tool CD ROBI.



Dragana Dudić obtained BSc and MSc degree in Computer Science at the Faculty of Mathematics, University of Belgrade, and PhD degree in Computer Science at the Faculty of Information Technology, Pan – European University Apeiron Banja Luka. She is an assistant professor at Faculty of Computer Science and Informatics, University Union Nikola Tesla. Dragana is author of more than 20 scientific papers published in peer-reviewed journals, with research focus on bioinformatics, digital literacy and technology integration in education.

Lazar Stošić is a university professor at the Faculty of Informatics and Computer Science, University Union—Nikola Tesla, Belgrade, Serbia and the President of the Association for the Development of Science, Engineering and Education, in Serbia. He is also a leading researcher at the Center for Scientific Competence of DSTU, Department of Scientific and Technical Information and Scientific Publications Don State Technical University, Russia. His expertise includes computer science, IKT, editorial workflow management, conference organization, web technologies, web design, indexing, XML production, SEO, digital marketing, and new media technologies.

For citation

Olja Krčadinac, Željko Stanković, Dragana Dudić, Lazar Stošić, Development of an Open-Source Voice-Controlled Smart Home System, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 14(2024)2:111-116, (UDC: 621.391:004.414.23), (DOI: 10.7251/JIT2402111K, Volume 14, Number 2, Banja Luka, December (89-188), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004