wjert, 2025, Vol. 11 Issue 7, 21-30. *Original Article* ISSN 2454-695X

World Journal of Engineering Research and Technology
WJERT
www.wjert.org SJIF Impact Factor: 7.029

VOICE COMMAND ASSISTANT HOME AUTOMATION SYSTEM FOR SMART CITIES USING SINRIC PRO

Rohit Goel*, Suraj, Pradeep Kumar and Mukta Goel

The Technological Institute of Textile and Sciences, Bhiwani, Haryana.

Article Received on 06/05/2025 **Article Revised on 26/05/2025** **Article Accepted on 16/06/2025**

***Corresponding Author**
Rohit Goel
The Technological Institute of Textile and Sciences, Bhiwani, Haryana.

ABSTRACT

The Internet of Things (IoT) is transforming the way humans interact with everyday appliances. This paper presents the development and implementation of an IoT-based voice commands assistant system utilizing ESP32 and Sinric Pro. The system is designed to control home appliances—such as lights, fans, and doors—through simple voice commands using a smart phone. The methodology includes hardware integration with ESP32, relay modules, servo motors and software platforms such as Voiceflow and Arduino IDE. A protection circuit is designed to protect the microcontroller from high current or voltages and testing of system confirmed successful voice recognition and device control with minimal latency. The study demonstrates the potential of combining low-cost microcontrollers with cloud services to create efficient, smart home automation systems. Future enhancements may include expanded voice command functionality and multi-device synchronization for improved user experience.

KEYWORDS: IoT (Internet of things), ESP32, Sinric Pro, Arduino, Voice Command System, Smart Home Automation, Servomotor.

INTRODUCTION

The Internet of Things (IoT) pertains to the widespread pairing of items to the internet, enabling them to communicate data to the cloud and maybe receive instructions for executing actions. The rapid growth of IoT has enabled the development of smart systems that enhance user convenience, safety and automation in daily life.[1] Smart Cities are characterized as the use of multiple forms of technology to improve the quality of life for the people who live there. This includes applying these technologies to various domains such as governance, public transit, housing, learning and businesses.[2]

One significant application of IoT is smart home automation, where voice-controlled systems simplify how users interact with household devices. Smart Homes use sensing units located inside the house for providing content about the home. These sensors may include user activity monitors such as environmental sensors, motion trackers and energy usage sensors.[3]

This research focuses on designing an IoT based Voice Commands Assistant System that can operate home appliances using voice input and a low-cost microcontroller—ESP32. This system aims to provide hands-free control of devices such as lights, fans and doors, particularly benefiting individuals with mobility impairments. The voice commands are processed through cloud services and translated into control signals for physical appliances. This paper outlines the system's development, tools used and testing outcomes. The main objective is to create an efficient, responsive, and low-cost solution for basic home automation.

Literature review gives overview of designs and software used by researchers for home automation. The next section explains the methodology, and a brief overview of hardware and software components, basic workflow and the design of protection circuit used. Further a detailed comparison of designs already used and the proposed design is given. Following sections discuss the major achievements of this work, concludes the paper and give future direction for the researchers.

Literature review

A lot of study is undertaken on smart home automation system which involves use of voice controlled systems. Smart home system based on IoT is designed in.[4] A framework for designing the system with an overview of applications is given. A smart home system using face recognition for opening and closing of doors in home is proposed in.[5] It uses Arduino UNO and Wi-Fi ESP8266 module to achieve the cost effective smart system. In,[6] all the devices of home are automated using Arduino and cloud based platform ThingSpeak. This is used as an interface between the mobile and the home devices. For LAN communication MQTT protocol is designed. A number of sensors such as flame sensor, gas sensor are used.

Servomotors are used along with relays for opening and closing of doors and switch on and off the lights and fans respectively. A prototype for a room and kitchen is designed.

Using ESP8266 and Tuya Convert OS for cloud services, a smart home automation system is designed.[7] A qToggle device comprising of sensors and other hardware is implemented. The ESP8266 Wi-Fi module interacts with Raspberry pi 4 controller for automatic ON-Off the devices. The system is used for AC temperature control, lights, energy monitoring, power monitoring and in irrigation using proper sensors. In,[8] authors discuss the applications and challenges of IoT in smart cities. It provides a framework for different applications which can be automated. A detailed framework for cloud based solution to smart cities is illustrated in.[9] It gives comparison of various cloud API‘s that can be used for virtual communication at smart homes while sitting at a far off place.

In,[10] ESP32 is integrated with Blynk app and an IFTTT platform. Blynk dashboard provides for controlling and monitoring the status of various devices. It is used to give commands for opening and closing of bulbs and working of relays. Based on commands from Blynk, IFTTT generates the rules and triggers the actions. A home automation system for four devices is made using ESP8266 microcontroller, a web API and using the Alexa or Google Assistant for voice commands.[11] The prototype can record and analyze the details such as power consumption, network details and ON OFF status.

Now-a-days machine learning is gathering steam and is used in almost every field for better optimization and results. A machine learning approach to classify the type of home appliances is adopted in.[12] Besides controlling the devices remotely using mobile phones, techniques are implemented to know the type of devices to be assisted.

This study follows a qualitative research approach, focusing on developing a functional prototype and measuring its effectiveness.

System design

For designing the home automation system in smart city, a number of components are required. The components used in the hardware implementation include:

- ESP32 Microcontroller – for Wi-Fi-enabled device control.
- Relay Module
- Servo Motor

- Jumper wires
- Power supply
- Smartphone with Voiceflow app
- Sinric Pro Platform

Hardware Components

1. *ESP32 Development Board* – It is the main microcontroller for handling Wi-Fi connectivity and controlling devices.[13] It is an energy efficient and low cost microcontroller having more memory and high processing power and is used for commercial and industrial IoT applications. It also has good security features and reliable connectivity.
2. *Relay Module (2-channel or 4-channel)* – To switch high-voltage devices like fan and light. They receive low power signals from ESP32 microcontroller and manage high power circuits for closing and opening contacts. They can connect and disconnect 220V voltage in homes and is thus used for handling bulbs and fans.
3. *Servo Motor (SG90 or MG90)* – Used for door control (open/close). They are mainly used for precise control of position and movement of a device. These are versatile, lightweight and small in size. It is connected with sensors
4. *LED Bulb or Indicator* – For testing light control.
5. *Fan or Motor (DC or AC)* – For fan control demonstration.
6. *Breadboard and Jumper Wires* – For circuit connections.
7. *Power Supply (5V/9V Adapter or USB)* – To power ESP32 and other components.
8. *Protection circuit* – A circuit is designed using transistor and resistor for protection of ESP32 and other components.

Figure 1 Shows the complete circuit diagram involving all the hardware components.

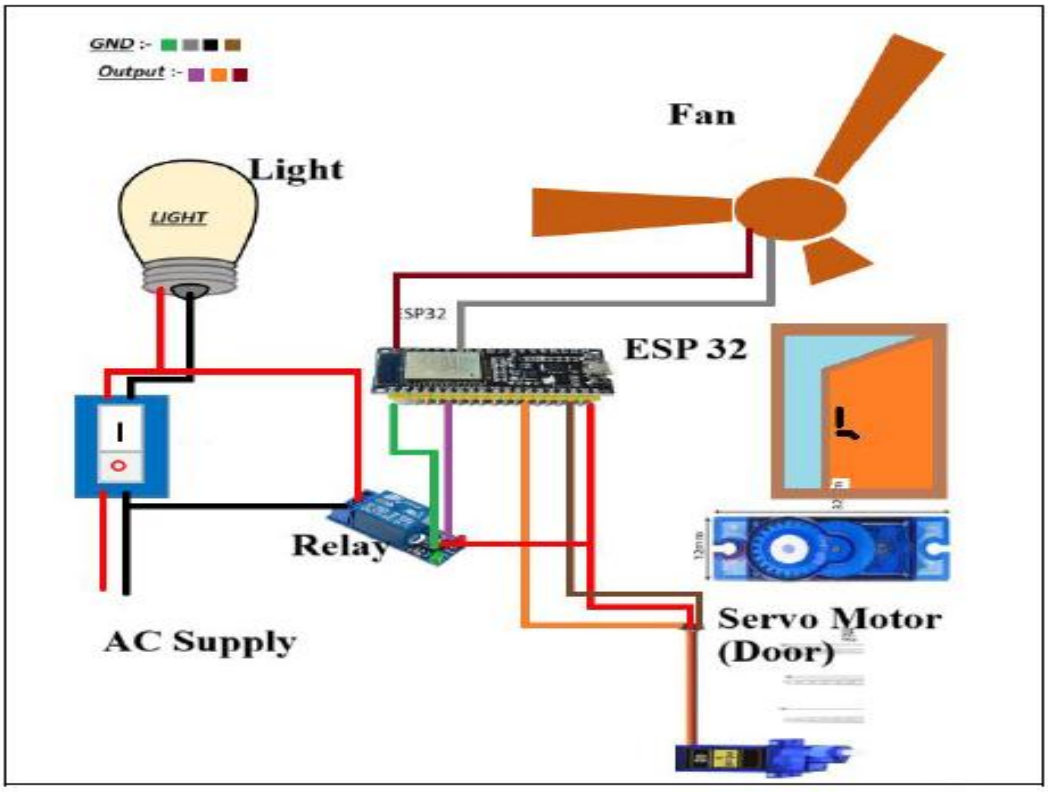

Fig. 1: Circuit diagram.

Software requirements

1. Arduino IDE: The Integrated Development Environment is used to write, and upload code to Arduino boards. It is an open-source software application to provide a user-friendly interface for microcontrollers. It is helpful for both beginners as well as experienced developers.
2. Sinric Pro Platform: It is a cloud platform and is used to control IoT devices. They make use of web sockets for faster two-way communication.[14] Voice commands through Google Home or Alexa can be easily controlled using it. It also supports control through mobile apps. A device connects to Sinric Pro cloud via WebSocket using API key and device ID.
3. Voice Assistant App: It is a software application which makes use of AI to give voice commands.
4. Sinric Pro Library: For integrating ESP32 with Sinric Pro in Arduino IDE.
5. Wi-Fi Network: Sinric Pro server is connected to ESP32 using it.

Cloud Platform (Sinric Pro)

Sinric Pro acts as the cloud server between the user and the ESP32 device. The various functions performed by Sinric Pro are:

Sinric pro dashboard: It is a web or mobile interface that allows manual control and monitoring of devices. It is also used for configuring routines.

Device registration: Each device (light, fan, and door) has a unique ID. A maximum of three devices can be registered for free.

Command routing: It receives commands from voice assistants and routes them to ESP32.

Real-Time updates: Sinric Pro can provide real-time feedback on device state.

Alexa or google assistant: It is used to give voice commands like "Turn on the fan" or "Open the door". Devices can be toggled ON/OFF, and logs can be viewed.

Feedback & Monitoring: Users receive real-time status updates of devices through the app.

Control logic (Workflow)

The basic workflow is having two directions. On one side, signals or voice command can be given from Alexa or Google assistant to Sinric Pro, who identifies the device. Then through cloud servers, the data reaches the ESP32 microcontroller through Wi-Fi. It triggers the related GPIO (General Purpose Input Output) pin. Now, the Arduino IDE program or code helps the related devices for which the signal is given to ON or OFF or open and closing the door through relay or servomotor respectively.

On the other side, data can be collected through sensors and received by ESP32 board. It then sends the data to Sinric Pro via cloud for monitoring and analysis. This way, home automation in smart cities can be easily done even from a remote site like a mobile. This can also help in looking after elderly persons at home by directly giving commands from a remote place.

Protection circuit

Another important work which differentiates from other researchers is the implementation of a protection circuit. The relays used for lights and fans mainly consist of inductances. The proposed circuit protects the microcontroller from high current or voltage spikes from such inductive loads. Also, GPIO pins can provide currents of 10-20 mA range. So, to manage large currents, a transistor is used which can deal with currents up to 600mA. The transistor also isolates microcontroller power from load power. This helps in safe and reliable switching.

The components used in protection circuits are

1. Transistor 2N2222: It acts as a switch. It turns ON when a small base current is applied, allowing a larger current to flow from collector to emitter.
2. 1K resistor: Limits current flowing into the base of the transistor. It prevents damage to the GPIO pin.
3. GPIO Pin: Sends a HIGH (3.3V or 5V) signal to turn ON the transistor via the base resistor.
4. Load: The device you want to control (like a fan, relay, or LED). It is powered by an external power source, isolated from the microcontroller. The basic protection circuit is shown in Figure 2:

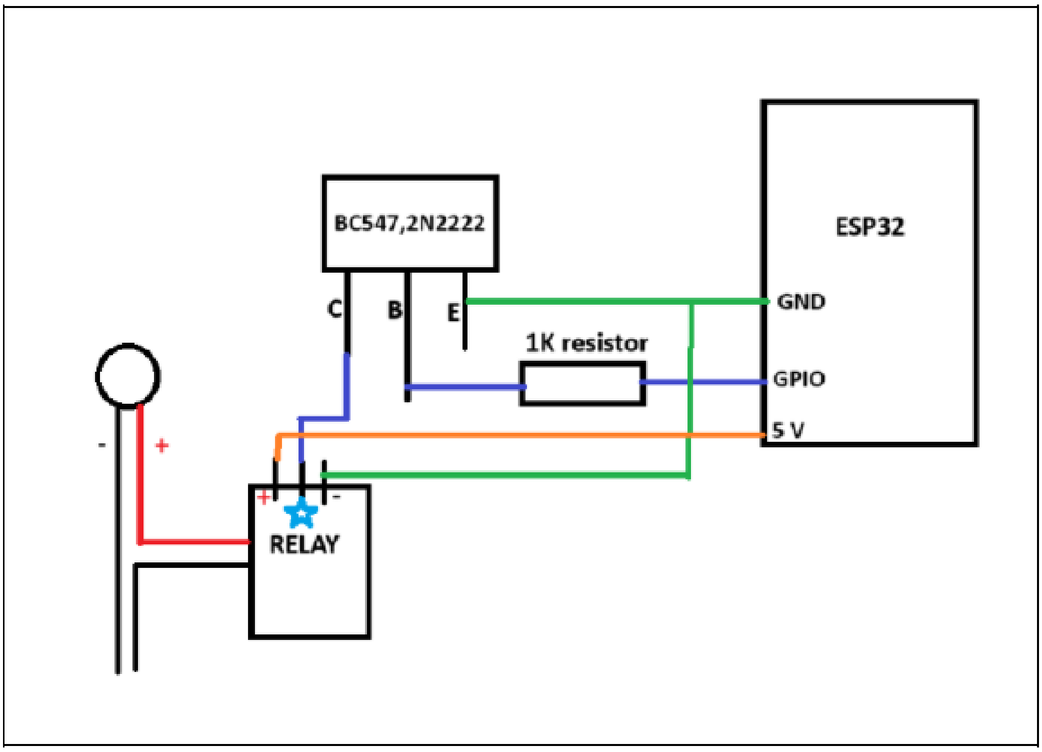

Fig. 2: Protection circuit.

RESULTS

The prototype system was successfully implemented and tested in a lab environment. Three devices (light, fan and door) were used for home automation. Firstly, device registration for all these three devices was done in Sinric Pro as shown in Figure 3. Voice commands such as "Turn on the light", "Turn off the fan", and "Open the door" were given using Google assistant and was correctly recognized. The voice command was then forwarded to ESP32 microcontroller through cloud platform and the corresponding devices responded accurately.

The ESP32 showed excellent connectivity with Sinric Pro, and all commands were executed within 1–2 seconds. Sinric pro activity log is given in Figure 4 which gives the correct information about the device turned ON or OFF. The servo motor rotated precisely to simulate door operation, and the relays triggered the fan and light as expected. No significant delays or errors were observed during multiple trials.

Fig. 3: Sinric pro device display.

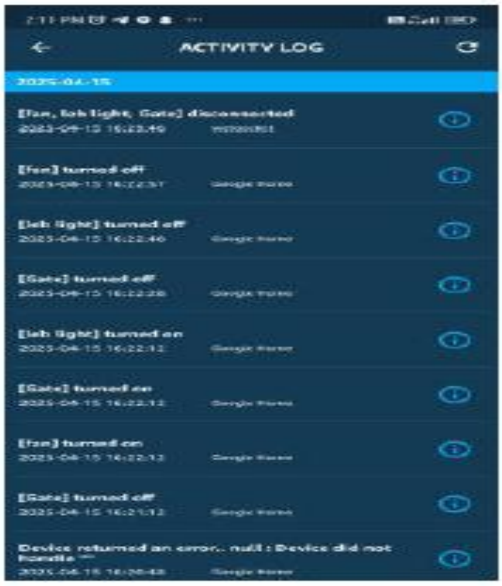

Fig. 4: Sinric pro activity log.

DISCUSSION

The results confirm that the IoT-based voice command assistant system functions effectively for home automation. The ESP32's reliable Wi-Fi capabilities and integration with Sinric Pro allowed real-time control of devices. Compared to traditional manual switches, voice control provides a seamless and hands-free experience. The system is especially beneficial for users with disabilities or the elderly, offering ease of access and enhanced safety.

Similar projects in recent literature have shown comparable outcomes, but many rely on expensive hardware or complex setups. Earlier ESP8266 was mostly used while ESP32 is used in this research which has a dual processor and supports multitasking. It supports simultaneous Wi-Fi and Bluetooth connectivity, has higher number of GPIO pins with advanced cryptographic in-built hardware for enhanced security. It also has low power consumption ensuring long battery life. So, in terms of features, performance and power consumption ESP32 far outweighs ESP8266. This project uses affordable components and open-source platforms, making it suitable for practical implementation in low-cost smart homes. In comparison to cloud server platforms used in most literature (IFTTT and others)

Sinric Pro uses encrypted communication and secure tokens for sending commands. It has hassle free device set up procedure and there is no need of any echo device which is mainly used by early research works. Sinric Pro has GUI based dashboard and supports multiple cross platforms which helps in easy connect with multiple systems. Another outstanding feature of this research is the protection circuit to prevent the ESP32 from high voltage or current fluctuations.

One limitation encountered was occasional latency due to internet speed fluctuations. Future enhancements can include offline voice processing, integration with multiple users, and compatibility with platforms like Google Assistant or Alexa.

CONCLUSION

This research demonstrates a successful implementation of a voice-controlled home automation system using ESP32 and Sinric Pro. The prototype reliably responded to voice commands, controlling lights, fans, and door mechanisms efficiently. The system highlights the potential of combining IoT and cloud technologies to develop scalable, affordable smart home solutions. Continued development can improve reliability and introduce more advanced functionalities, leading to widespread adoption in smart environments.

ACKNOWLEDGEMENT

I would like to express sincere gratitude to the department for valuable suggestions, providing resources and conducive environment to carry out the research. Their support, encouragement, and technical insights were crucial to the successful completion of this research.

REFERENCES

1. Sharma, A., Kaur, R. "IoT-based voice-controlled smart home system using Google Assistant and NodeMCU", International Journal of Advanced Research in Computer Science, 2020; 11, 3: 45-49.
2. Abbas Shah Syed, et al. "IoT in Smart Cities: A Survey of Technologies, Practices and Challenges", 2021; 4, 2: 429-475.
3. Suryawanshi, V., Patil, A. "Voice Controlled Home Automation Using IoT and Cloud Technology", International Journal of Scientific & Engineering Research, 2021; 12, 5: 112–116.
4. Vivek Kadiyan, et al. "Smart Home Automation System Based on IoT", International Journal of Scientific Research in Engineering and Management (IJSREM), 2023; 7, 5: 1-8.

5. A. P. Nirmala, et al. "IoT Based Secure Smart Home Automation System", 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 2022.
6. Mubashir Ali, et al., "An IoT based Approach for Efficient Home Automation with ThingSpeak", International Journal of Advanced Computer Science and Applications (IJACSA), 2020; 11, 6: 118-124.
7. Cristina Stolojescu-Crisan, Calin Crisan, Bogdan-Petru Butunoi, "An IoT-Based Smart Home Automation System", International Symposium on Electronics and Telecommunications (ISETC), Sensors Journal, 2020; 21, 11: 1-23.
8. Iqra rafiq, et al., "IoT applications and challenges in smart cities and services", The Journal of Engineering, 2023; 4: 1-25.
9. Tanweer Alam, "Cloud-Based IoT Applications and Their Roles in Smart Cities", Cloud-Based IoT Applications for Smart Cities, Smart Cities Journal, 2021; 4, 3: 1196-1219.
10. Kunal Kumar Sahu, "IoT Based Voice Controlled Home Automation System using Google Assistant", i-manager's Journal on IoT and Smart Automation, 2024; 2, 1: 27-37.
11. Vaishnavi Kulkarni, et al. "Smart Home Automation Using Google Assistant and Alexa", International Education and Research Journal (IERJ), 2024; 10, 5: 203-206.
12. Quadri Waseem, et al. "Exploring Machine Learning in IoT Smart Home Automation", IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS), Penang, Malaysia, 2023.
13. Espressif Systems, "ESP32 Series Datasheet", https://www.espressif.com/en/products/socs/esp32.
14. Sinric Pro., "Smart Home Platform for Developers", https://sinric.pro.