iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71795

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Smart Home Appliances

Asmit Sanagare[1], Nachiket Nalawade[2], Pravish Shukla[3], Lokesh Bhandakkar[4], Santosh Kamble[5]
Electronic and Computer Science, Shah and Anchor Kutchhi Engineering College, Mumbai, India

Abstract: ***The rise of the Internet of Things (IoT) has driven the development of smart home systems that enhance comfort, energy efficiency, and security. Traditional systems are either limited in scope or prohibitively expensive. This paper proposes a low-cost, real-time, wireless home automation system based on the ESP32 microcontroller and Google Firebase, incorporating multiple sensors and actuators. The system is designed to allow real-time control and monitoring via a mobile application developed using MIT App Inventor. A user-friendly mobile application developed using MIT App Inventor allows remote monitoring and control. The solution offers an efficient, scalable, and cost-effective platform for modern smart home implementations.***

Keywords: ESP32, Firebase, Relay, Sensors (Temperature & Humidity, LDR , Ultrasonic Sensor) , IOT , MIT App Inventer.

I. INTRODUCTION

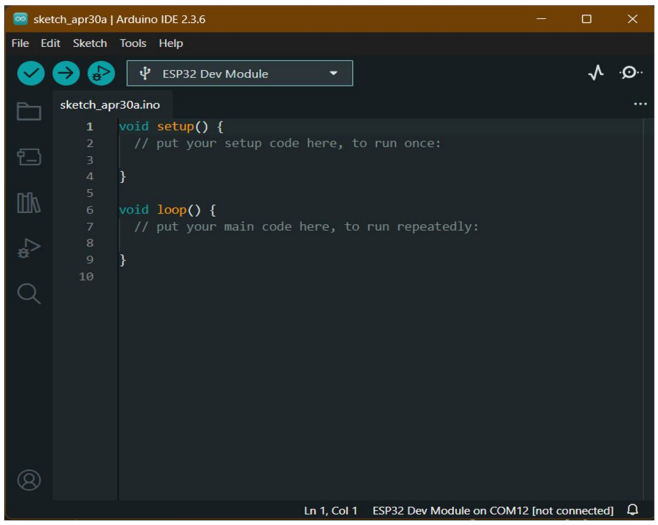
Smart home automation has rapidly emerged as a transformative technology, offering homeowners the ability to monitor, control, and optimize their living spaces from anywhere in the world. It involves the integration and coordination of various household appliances and systems through a central control unit, providing seamless connectivity and interaction between devices. Using a centralized controller, such as a microcontroller or a cloud platform, users can manage everyday utilities ranging from lighting and temperature to security systems and entertainment devices.

Home automation empowers users to configure, schedule, and remotely control multiple home functions via computers, smartphones, or dedicated mobile applications. This convergence not only enhances the comfort and convenience of daily life but also contributes to improved energy efficiency, security, and overall system intelligence. For successful smart home implementation, it is crucial that all connected devices communicate reliably and efficiently, sharing status updates and control signals across the network. The primary objective of smart home systems is to enable real-time monitoring and management of household equipment and environmental parameters, often using wireless communication technologies. Today's automation solutions are highly secure, reliable, and affordable, making them accessible to a broad range of users. The growing availability of user-friendly cloud services and low-cost microcontrollers has fueled widespread adoption, allowing homeowners to benefit from smarter, more responsive living environments. This project focuses on enhancing traditional, non-smart electrical appliances by enabling wireless control without the need for expensive smart devices. It aims to bridge the gap between legacy systems and modern IoT-based controls using affordable hardware and open-source platforms. Specifically, this work leverages the ESP32 microcontroller, known for its powerful onboard Wi-Fi capabilities, to interface with appliances through three relay modules. Real-time data handling and communication are facilitated using Google's Firebase Realtime Database, enabling fast and reliable data storage and retrieval over the cloud. Additionally, this project integrates essential environmental monitoring through sensors such as the DHT11 for temperature and humidity measurement, an LDR for ambient light detection, and an ultrasonic sensor for presence or motion detection. A user-centric mobile application, developed using MIT App Inventor, acts as the command center, allowing users to interact intuitively with their home appliances and sensor systems. Unlike more complex systems that require separate websites or backend hosting platforms, the proposed solution centralizes all functionalities within the mobile app and Firebase cloud, offering a more straightforward and accessible user experience.

II. SOFTWARE APPLICATION

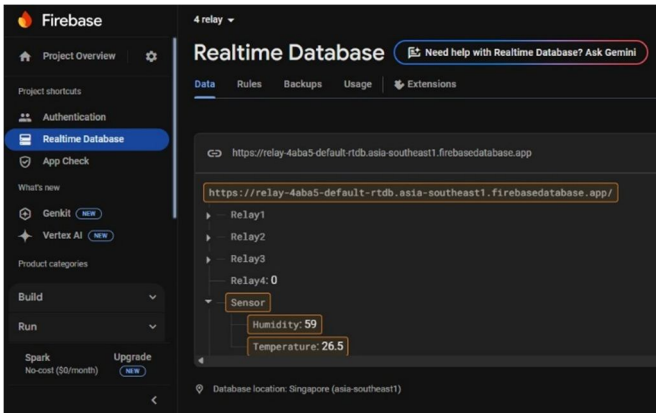
A. Audino IDE

The software implementation for this smart home automation system utilizes the Arduino Integrated Development Environment (IDE) for programming the ESP32 microcontroller. Arduino IDE is an open-source platform that provides a simple interface for writing, compiling, and uploading code. It supports multiple operating systems including Windows, Linux, and macOS, and is compatible with both C and C++ programming languages. The ESP32 Devkit V1 board is connected to the development machine via a micro-USB cable, which also supplies power to the module, as indicated by an onboard LED. Once the code is compiled, it is uploaded directly to the ESP32, which executes the programmed instructions to interact with connected sensors and control relays accordingly.



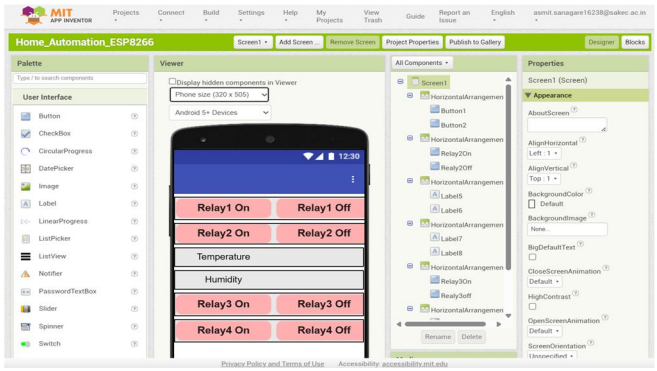
B. Firebase

For real-time data management and cloud connectivity, this system employs Google Firebase, a Backend-as-a-Service (BaaS) platform. Firebase provides a suite of cloud-based tools that facilitate secure, scalable, and real-time application development. Specifically, the Firebase Realtime Database is used in this project to store and synchronize sensor readings and control signals between the ESP32 and the user interface. The database enables bi-directional communication, allowing data to be written and retrieved from the cloud in real time.



C. MIT App Inventer

To provide an accessible and user-friendly mobile interface, MIT App Inventor is used to develop a custom Android application. The app is designed to interact with Firebase, enabling users to monitor sensor data and control the three relay modules remotely. Through this interface, users can switch appliances on or off and receive real-time feedback from the sensors, thereby enhancing the system's usability and responsiveness.



III. HARDWARE SPECIFICATIONS

The ESP32, developed by Espressif Systems, is a highly versatile and cost-effective System-on-Chip (SoC) microcontroller, making it an ideal choice for IoT-based applications. It is offered in both single-core and dual-core variants, with integrated Wi-Fi and Bluetooth capabilities as standard features. The schematic structure of the ESP32 is illustrated in Fig. 1. Key specifications of the ESP32 module include:

- 34 programmable General Purpose Input/Output (GPIO) pins.
- A 32-bit LX6 microprocessor, available in single or dual-core configurations, operating at clock speeds up to 240 MHz.
- Support for IEEE 802.11 b/g/n Wi-Fi standards with speeds up to 150 Mbps.

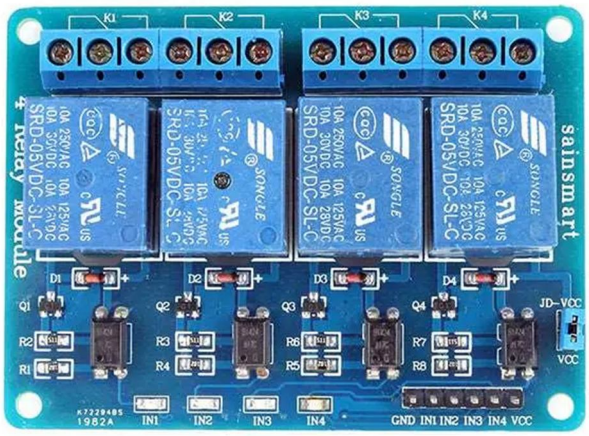
Memory resources including 16 KB of RTC SRAM, 520 KB of on-chip SRAM, and 448 KB of ROM.

The ESP32 can be conveniently powered through a micro-USB port, commonly using a standard mobile phone charger or USB power source.

In this project, the ESP32 serves as the central processing unit (CPU), responsible for managing all control logic and communications. The user-written program is uploaded to the ESP32, which interacts with peripheral devices such as relays and sensors through its GPIO pins. The relays are used to control household appliances, while sensor data inputs are processed to enable real-time decision-making and cloud synchronization, as depicted in Fig. 2.

A. Relay

A relay is an electromechanical or electronic switch that can open and close a circuit in response to an electrical signal. It operates based on the principle of electromagnetic induction, where a current passing through a coil generates a magnetic field that activates a mechanical switch to connect or disconnect another circuit. The relay module used in this project is shown in Fig. 3.



In this setup, the relay module is interfaced with the GPIO pins of the ESP32 microcontroller, which sends the necessary control signal to activate or deactivate the relay. This allows for the switching of connected appliances or devices, such as light bulbs, based on real-time inputs or pre-configured scenarios.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

B. Light Bulb

The light bulb is connected to the relay, but in this system, it is not directly controlled through user input. Instead, the light bulb will turn on or off automatically based on ambient light levels detected by the Light Dependent Resistor (LDR). If the LDR detects insufficient natural light (e.g., in the evening or when there is no external light), the system will trigger the relay to turn the light bulb on. This automatic control is based on environmental conditions, rather than manual activation.

C. Air Purifier

Similar to the light bulb, the air purifier is connected to the relay but does not require manual control by the user. The air purifier is automatically activated when the air conditioner (AC) is turned on. This setup ensures that whenever the AC is started, the air purifier also begins its operation to enhance indoor air quality. The relay, controlled by the ESP32, manages the power supply to the air purifier in sync with the AC's operation.

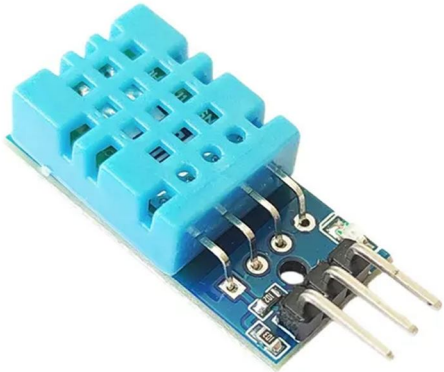
D. Air Conditioner

The air conditioner (AC) is the primary device controlled by the system. When the AC is turned on, it will automatically set its temperature based on the readings from the DHT11 temperature and humidity sensor. The ESP32 processes the data from the sensor and adjusts the AC's settings to maintain a comfortable room temperature. As soon as the AC is activated, both the light bulb and air purifier will be automatically started, creating a more comfortable and controlled indoor environment.

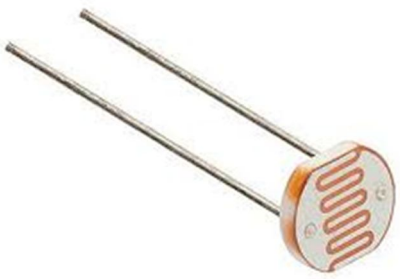
E. Sensors

The project integrates several key sensors to monitor the environment and provide data for controlling the system:

- DHT11 (Temperature and Humidity Sensor): This sensor measures the ambient temperature and humidity. It serves as the primary input for regulating the AC's temperature. When the temperature exceeds a set threshold, the ESP32 activates the relay to power the AC and adjusts the temperature automatically.



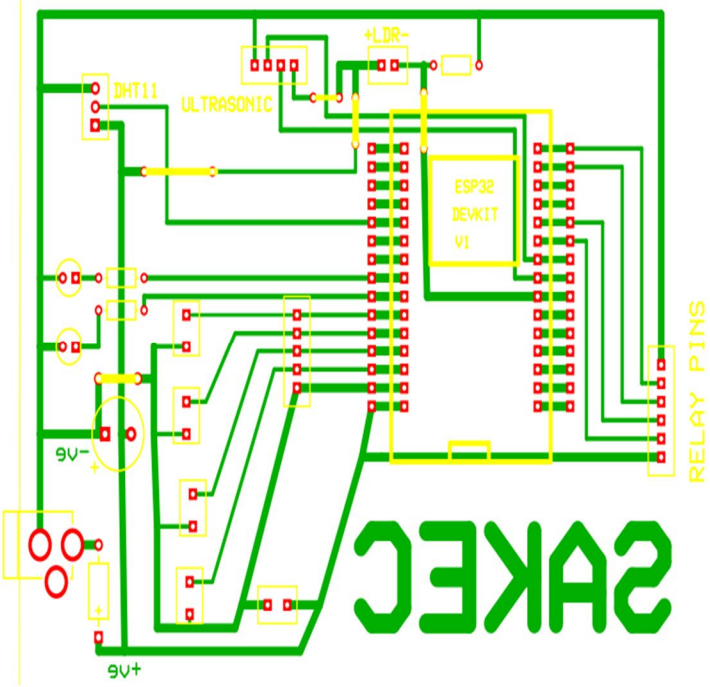
- LDR (Light Dependent Resistor): The LDR detects the ambient light levels. It ensures that the light bulb is turned on automatically when natural light levels fall below a certain threshold, such as at night or in dark environments.



- Ultrasonic Sensor (Distance Sensor): This sensor measures distance and detects the presence of objects or people in the room. It can trigger the system to perform actions, such as activating the light or turning off the AC when no one is present.

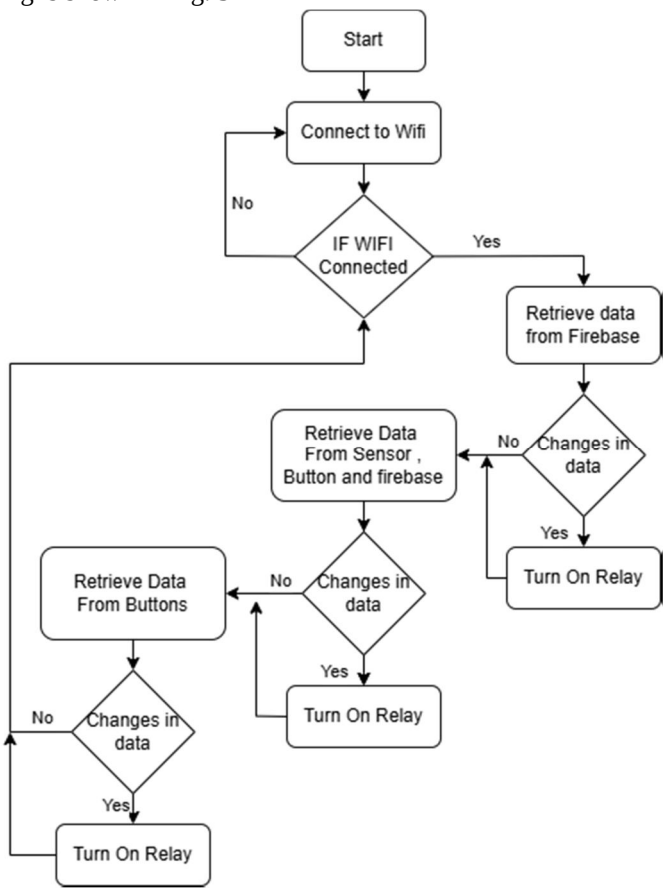
F. PCB

The PCB design features an ESP32 microcontroller connected to a DHT11 sensor for temperature and humidity, an IR sensor for motion detection, an ultrasonic sensor for distance measurement, and tactile buttons for manual input. Components are arranged to ensure compactness, stability, and efficient signal flow, making the board suitable for smart home and automation applications.



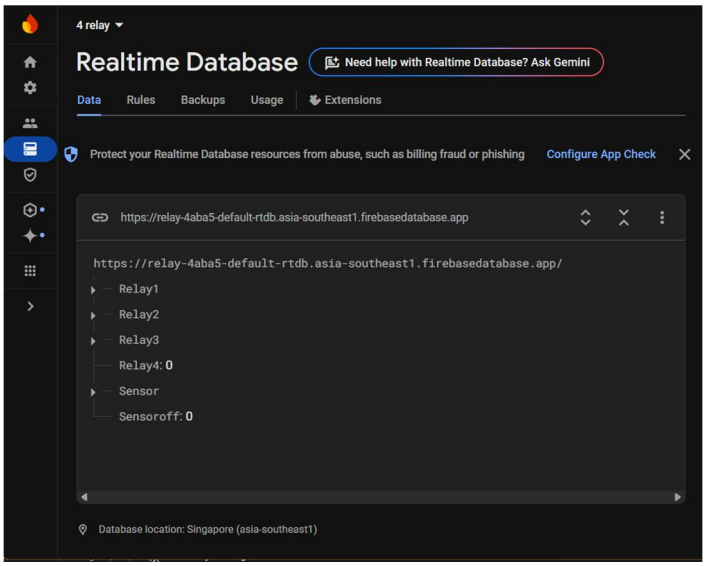
IV. WORKING

The flow chart explaining the working is shown in Fig. 5



The hardware connection or circuit diagram showing connections between the relay module , Sensors , Buttons and the ESP32 board is shown in above image.

The ESP32 is connected to the Firebase and clicking on the virtual button on the website changes the data in the database. This way the bulb turns on/off as desired. The program is written in ESP32. Firstly check for Internet connection, if it is not connected then it will keep on checking for the connection. Once connected to the internet, it will retrieve data from firebase and perform action based on the data received from fire base.

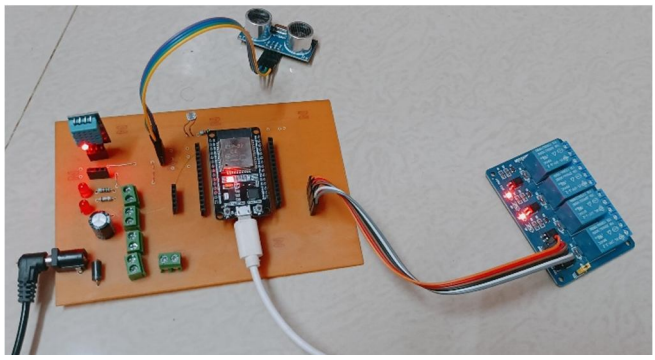


Here, a database is created in the firebase as shown in Fig. 7 to store the values of different relays which will be read to turn on or off the relays respectively.

Next, the virtual switches as shown in Fig. 8 are created and with the help of these virtual relays, one can control the relays remotely.

V. RESULTS

In the results, it shows that when we start any switch from firebase , Button , Serial input , Through Sensors and mit app it gives output as required as shown in figure below



The smart home automation enables us to access high-tech functionality and elegance that was previously unattainable. Almost all house appliances may be controlled remotely and usually from a single app with home automation. We can do a lot of things with home automation, like, we can customize certain features in our way we like and thus making things more convenient. This project aims at enabling remote access to our home appliances while also maintaining the manual controls available for there is no internet available.

I will allow us to control appliances from Distance like if we are out we can water plants from distance , before coming home cooling room to specified temperature .

The following are the precautions that need to be taken to ensure smooth functioning of the device

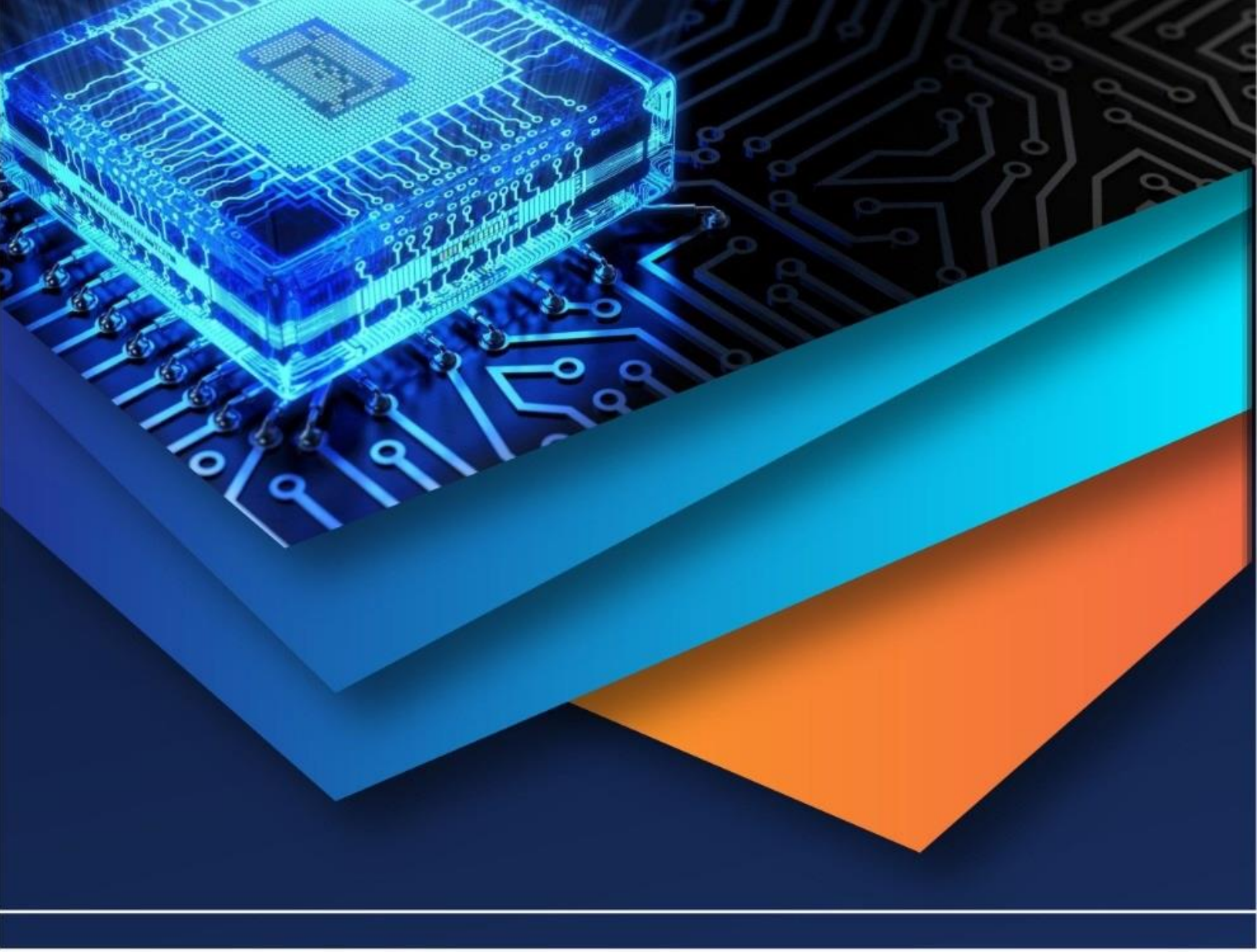
- In case of a power outage, we need to operate these manually.
- The automation system should have access to a reliable local area network, or any other private network, to ensure seamless functionality.

VI. CONCLUSION

This project successfully demonstrates a scalable and real-time smart home automation system using Firebase and ESP32. With sensor integration and a mobile interface, it enables intelligent monitoring and appliance control. Future improvements may include machine learning for predictive automation and integration with voice assistants.

REFERENCES

[1] Srivastava, "IoT Based Home Automation using Firebase and ESP32," International Journal of Advanced Research in Computer Science, vol. 11, no. 4, pp. 142-146, 2023.
[2] M. Patel and R. Singh, "Design of Smart Home System Using LDR and DHT11," IEEE Conference on IoT Systems, 2022.
[3] D. Giusto, A. Iera, G. Morabito, and L. Atzori, The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications, Springer, 2010.
[4] Espressif Systems, “ESP32 Technical Reference Manual,” [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
[5] Google Firebase, “Firebase Realtime Database,” [Online]. Available: https://firebase.google.com/docs/database
[6] M. Collina, “MQTT Essentials – A Lightweight IoT Protocol,” IEEE Internet Computing, vol. 22, no. 1, pp. 14–18, Jan.–Feb. 2018.
[7] MIT App Inventor, “MIT App Inventor Official Documentation,” [Online]. Available: https://appinventor.mit.edu/explore/documentation
[8] S. Misra, A. Mukherjee, and M. S. Obaidat, “Smart Environment Monitoring Using Wireless Sensor Networks in Internet of Things,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1328–1339, Oct. 2017.
[9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
[10] R. Piyare, “Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone,” International Journal of Internet of Things, vol. 2, no. 1, pp. 5–11, 2013.
[11] M. J. Khan, S. M. Rizvi, and F. Anwar, "Design and Implementation of a Cloud Based Real-time Smart Home Automation System," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2020, pp. 923–928.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY