TechNest

Sayli Shinde [1], Shruti Singh [2], Salman Patel [3], Manjiri Pathak [4]

[4] Associate Professor,
[1, 2, 3, 4] Department of Computer Engineering, Vasantdada Patil Pratishthan's College of Engineering, Mumbai, Maharashtra, India.

Published in IJIRMPS (E-ISSN: 2349-7300), Volume 12, Issue 2, (March-April 2024)

Abstract

This study examines how comfort, convenience, and energy efficiency can all be significantly improved by IoT-based home automation. Significant energy savings and enhanced user experiences are revealed by our studies. We present an in-depth analysis of the modern IoT technologies, complete with a working prototype system that makes use of ESP32-enabled switches that are managed via a rainmaker application. The suggested method makes IoT-based home automation accessible and advantageous for modern families by streamlining switch operation and providing a financially viable substitute for traditional smart switches. Additionally, our system includes IR remote control and voice assistant features, making it easier for disabled individuals to use. By incorporating these features, our solution optimizes energy use and enhances convenience for all users, contributing to inclusive and efficient smart homes.

Keywords: Home Automation, Smart Light, Smart Fan, IR Remote, Voice Assistant, ESP32, ESP Rainmaker

1. Introduction

The home automation market has grown quickly in recent years, bringing advanced technology that makes life more convenient. As people shift from manual to automated systems, the Internet of Things (IoT) is playing a major role in this change. With its origins in home automation, the IoT is known for its ease of data sharing and transmission across networks as well as its ability to provide remote management through the internet. Physical engagement is needed to activate or deactivate the electricity of each appliance when using analog switches, which are typically situated on walls and present a laborious operational issue. This study presents an advanced approach that makes use of a clever trick that uses a Web browser on a PC or mobile device, IR Remote and Voice Assistant to operate a switch. In contrast to existing market solutions, which are frequently expensive and require additional hubs, our technique uses a Web Browser, IR Remote and Voice Assistant to simplify the control of manually controlled switches. Unlike other systems that use a variety of communication protocols, this study purposefully chooses to use the widely used IEEE 802.11 (Wi-Fi) and Bluetooth standards. By using this cutting-edge technology, users may easily interact with the processor through the Web Browser, improving user comfort and opening the door to a more effective and convenient home automation experience.

The next sections examine the additional research. The associated research encompasses the literature review conducted on numerous journal articles, conference proceedings, and websites. The next section III contains the proposed system. In Section IV, the necessary hardware and software are listed. The block diagram is shown in the V section. The section on Design and Development shows the process flow. The Expected Outcome section contains details on the generated results. The section under "Conclusion and Future Scope" outlines the conclusion and the work to be done in the future. All of the reference materials for our project are listed in the Reference Section.

2. Related Work

The authors have explained the home automation system using the ESP32 microcontroller in [1]. Users can remotely operate household appliances from cellphones through an Android app that uses JavaScript, while physical switches are available as a backup. Through IoT connectivity, the NodeMCU (ESP32) microcontroller and relays allow for remote switch control from the server, improving home efficiency and convenience. In [2], the authors have described a study that focuses on an IoT powered, affordable home automation system. Relay technology is used in this system to provide remote control of household appliances through the use of the Blynk IoT server and the Esp32 module. Users that have internet connectivity can utilize it to manage their gadgets from anywhere. The solution ensures effective communication between the connected appliances and the IoT server by using the Esp32 module. The motive of this research is to help a larger audience benefit from home automation by increasing its accessibility and usability. In [3], the authors have indicate that most of home automation systems currently in use concentrate on a specific task, such as remote appliance control, utilizing Bluetooth technology. Even though things have gotten better over time, new arrangements and technology are still welcome. The motive of this article is to create a complete home automation system that includes an Internet of Thingsbased appliance control setup, a smart energy meter, and a smart surveillance system.

The authors have discussed in [4], the home automation is becoming more and more common as technology advances, making daily life easier. In order to Ease home automation activities, this paper presents qToggle, a system that links sensors, actuators, and other devices. For basic device-to-device communication, qToggle uses an adaptable Application Programming Interface (API), usually with Raspberry Pi boards or ESP8266/ESP8285 chips. Users may effortlessly operate home appliances and sensors with a smartphone application. In [5], the authors have discussed the qToggle system that provides flexible and easy-to-use automation options. It utilizes the NodeMCU-ESP8266 microcontroller board for real-time sensing and processing, in addition to the EmonCMS platform for data collection and visualization. In paper [6], the authors have suggested a single paradigm for smart home systems. They examine how learning algorithms can improve security in smart homes and talk about possible risks and countermeasures.

3. Proposed System

The Android operating system provides the benefit of using open-source resources in the suggested system, together with simple access to the device's integrated sensors. A number of functions are included in the control application to ease system management. The Android phone uses socket programming to communicate data as a client. The program provides users with 4 unique ways to interact with it:

- **Switch Mode**: This mode lets you operate household appliances with radio buttons. Switching the radio buttons enables users to control the appliances by changing the status of the switches.
- **Voice Mode**: With this mode, users can use voice commands to operate home appliances. Using the inbuilt microphone of Smartphone, the application creates an intent that retrieves the speech data to the Google server which responds with a string data. The string data are further analyzed and then processed.
- **Manual Mode**: Using this mode, users can operate home appliances with actual switches.
- **IR Remote Mode**: In this mode, users can control compatible home appliances using an infrared remote control.

4. Hardware and Software Requirements

The required hardware must consist ESP32 Microcontroller boards, soldering kit, Sensors as temperature and humidity sensor, Actuators, PCB, Relay module, LED, Resistors and IR Remote. Software requirements include AI assistant and User Interface.

- **Microcontroller Board**

The ESP32 microcontroller board is used for processing data from sensors and transporting outputs to user interface. It includes multiple digital input/output pins, flash memory for storage, and pins for wireless transmission and reception and GND and VCC pins for power supply and grounding purposes as explained in [10].

- **Sensors**

DHT11 Temperature and Humidity Sensor: This sensor module provides the real-time humidity and temperature of the surrounding. The temperature is measured in Celsius Degrees and Fahrenheit as explained in [11].

- **Actuators**

The actuators that are used for implementing the circuit are relay module. The actuator such as LED gives an output to the system.

5. System Architecture

The block diagram mentioned below in Fig.1. gives an overview of the functionality of the overall project. The ESP32 unit functions as the microcontroller or the main controlling unit of the system. The user uses the mobile application to set commands for the functioning of the appliances. The mobile app interprets user commands in various modes (voice, switch, manual, or IR) and sends them wirelessly to the ESP32 unit via Wi-Fi. The ESP32's built-in Wi-Fi module enables it to communicate with devices and receive commands from the app over a wireless network. The ESP32, upon further receiving the signal, then turns on or off the appliance with the help of a relay. There is a power supply unit that powers the microcontroller, the relay, and the final appliances. There is a display unit that displays the status of the application. [7]

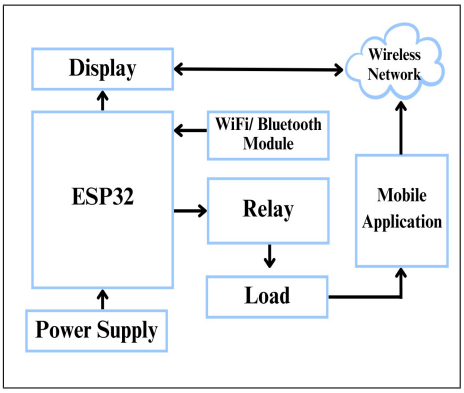

Fig 1: Proposed System Block Diagram [7]

The block diagram mentioned below in Fig. 2 gives an overview of the functionality of the ESP Rainmaker application. The user can develop and interact to system using this application. ESP RainMaker includes tools like a device-agent SDK, a cloud app, and cellular apps for iOS/Android. It facilitates developers to easily upload device settings to their firmware and sync them with the cloud. This allows instant control of gadgets through cellphone apps or other services, while not having to address cloud code as explained in [9].

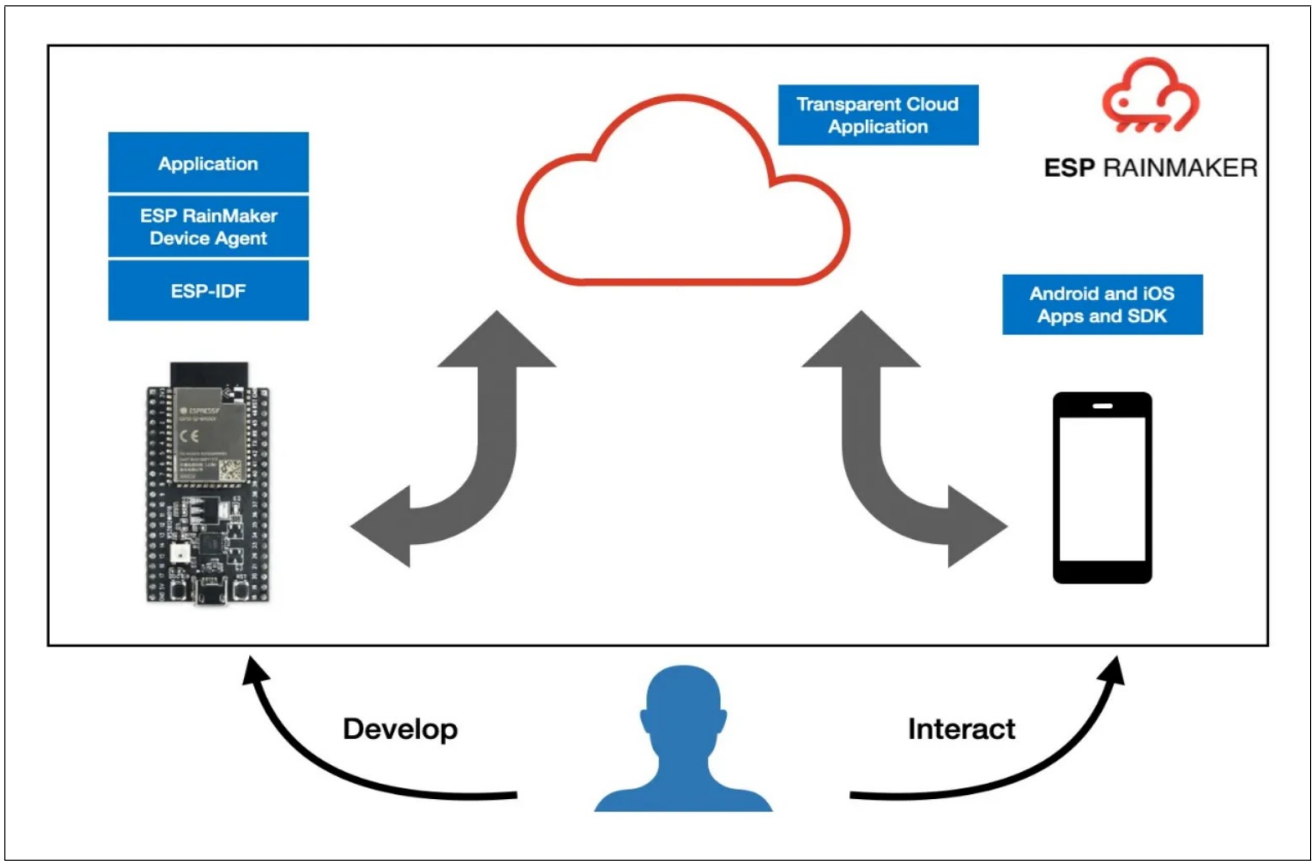

Fig 2: ESP RainMaker Block Diagram [9]

6. Hardware Implementation

The ESP32, shown in Fig.3, is just like the mind of our machine. It's a small but effective laptop that controls everything. When you use the mobile app to inform your appliances what to do, the ESP32 listens and makes certain they comply with your instructions. [10]

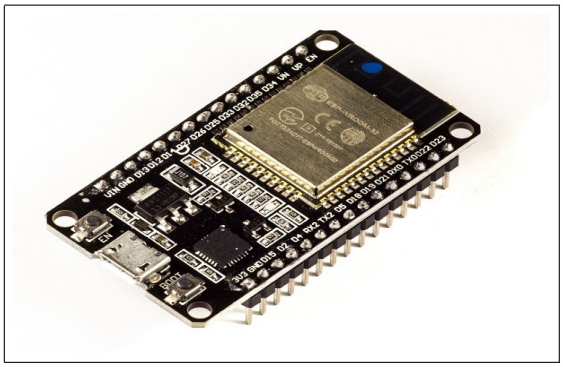

Fig 3: ESP32 [11]

We use ESP32 in preference to the Raspberry Pi because it's smaller, inexpensive, and uses less strength. ESP32 is perfect for controlling simple duties like turning appliances on and off, even though the Raspberry Pi is more desirable for more complex obligations. ESP32 receives the activity performed correctly without the extra capabilities that the Raspberry Pi offers, making it a more realistic choice for our task. [7]

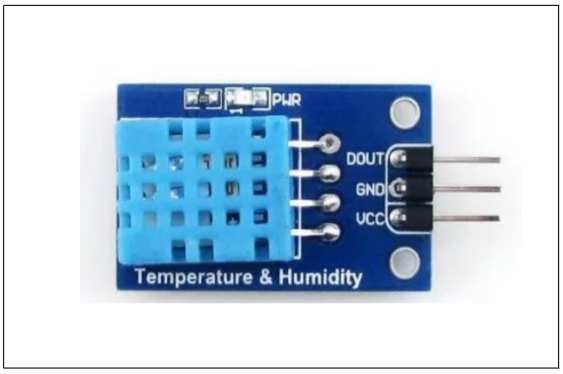

Fig 4: DHT11 Temperature and Humidity Sensor

The DHT11 temperature and humidity sensor shown in Fig. 4. is a weather sensor for our system. It measures the humidity and temperature of the surrounding air and sends that information to the ESP32. This enables the system to determine if it's too warm, too cold, too moist, or simply right, so it can modify as a consequence. Not only can we monitor the humidity and temperature, but we can also adjust the fan speed. This manner that if the sensor detects that the room is just too warm or too cold, the system can alter the fan pace to optimize the climate. It is sort of a clever thermostat in your fan, creating conditions the best types in your home. [12]

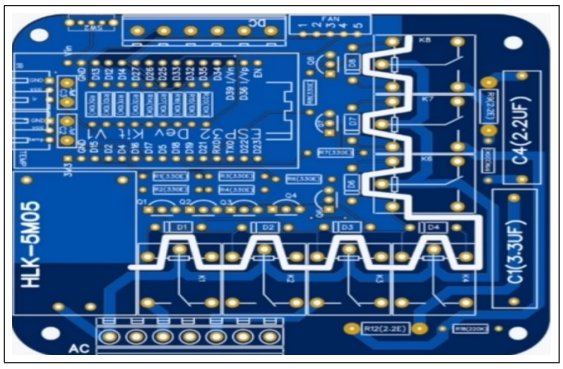

Fig 5: Printed Circuit Board

To give the project a competent look and make the circuit compact, we have designed a PCB for this ESP32 IoT based home automation project and the necessary components for relay module PCB are:

- DHT11 Sensor
- IR Receiver
- Terminal Connector
- 5v Relay
- Diode IN4007
- Transistor BC547
- Resistor 330 Ω 1/4W
- Resistor 220kΩ 1/4W
- Resistor 2.2Ω 1/2W
- Resistor 10kΩ 1/4W
- Slide Switch
- Male Connector
- Capacitor 2.2uF 250v
- Capacitor 3.3uF 250v [8-9]

This project allows to control home appliances and fan speed using ESP Rainmaker application, IR remote and manual switches, even without internet. Using the ESP32, it connects to the ESP RainMaker IoT cloud if Wi-Fi is available, allowing the global control of lights and fan speed via the internet and monitoring feedback in apps like Amazon Alexa, Google Home, and the ESP RainMaker App. The circuit is straightforward, using GPIO D13, D12, D14, and D27 to control the 4-channel relay module. GPIO D33, D32, D15, and D4 are connected with switches, and a 4-step selector switch controls the relay module manually. Instead of pull-up resistors, the Arduino IDE's INPUT_PULLUP function is used. The relay module turns on when receiving a LOW signal and off with a HIGH signal. An IR remote receiver (TSOP1838) is connected to D35, and a DHT11 sensor is linked to RX2 (GPIO16). A 5V 5Amp mobile charger powers the circuit. [9]

The circuit diagram for connecting the AC appliances and switches of the system is shown in Fig. 6.

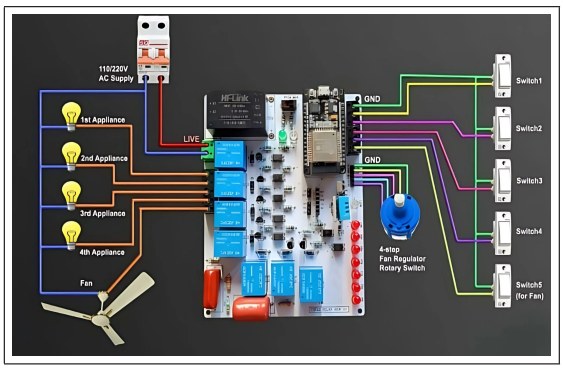

Fig 6: System Circuit Diagram [8]

7. Software Implementation

ESP Rainmaker is an IoT platform developed by Rapid IoT, which makes it easier to control and manage IoT devices. The user-friendly interface enables users to easily install, configure, and manage devices from the main dashboard. It is platform-independent and convenient to use. We are using ESP Rainmaker application to manage and control the system. It is an innovative platform that enables

firmware developers to effortlessly connect devices that may be controlled via phone apps, other offerings, or voice assistants, all without having to worry about handling infrastructure.

7.1. Add Devices to the ESP RainMaker Application

To Connect Appliances to the system, follow these steps:

- **Download and Install ESP RainMaker App**: Go to Google Play Store or App Store to download ESP RainMaker app. After downloading, install it on your smartphone as shown in Fig 7.

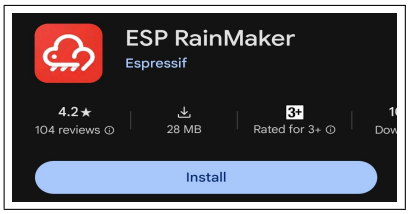

Fig 7: Download and Install

- **Open the App and Sign In:** Launch the app on your smartphone and sign in if you already have an account. If you haven't already, you'll need to register for one as shown in Fig 8.

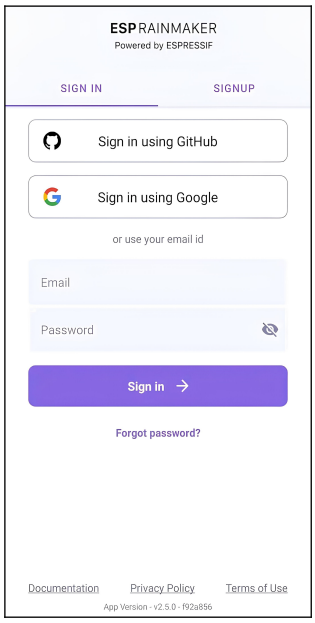

Fig 8: Sign Up

- **Turn Bluetooth on:** Enable Bluetooth on the smartphone.
- **Add Device:** If you haven't added a device before, you'll see a home screen with an "Add Device" button. Click here.

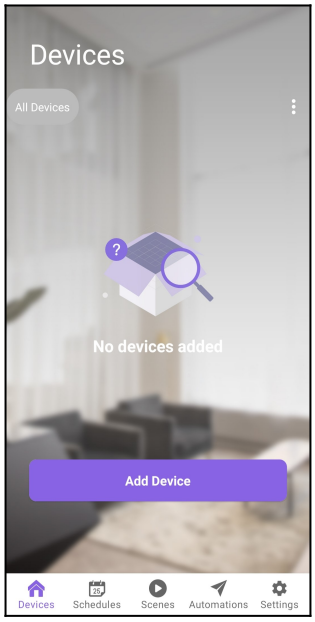

Fig 9: Add Devices

- **Scan QR Code:** Use the app to scan the QR code that appears on your device. This initiates communication with the BLE connection and the ESP32 board.

Fig 10: Scan QR Code

- **Select Wi-Fi Connection:** After clicking, the app will prompt you to select a Wi-Fi connection. If your network is not listed, you can enter the Wi-Fi credentials manually.

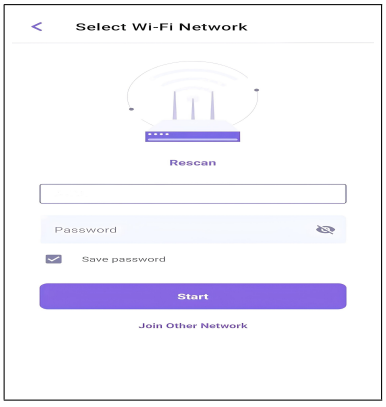

Fig 11: Select Wi-Fi Connection

- **Provisioning Process:** The app will guide you through several verification steps.

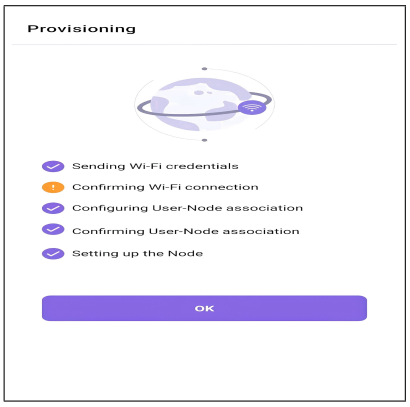

Fig 12: Provisioning Process

- **Closing:** After all the steps are tick click "Done". The ESP device will then link to the Wi-Fi network and appear as another device on the app's home screen called "Switch"

Fig 13: Added Devices

- **Setup Devices:** Now, you can use the switch buttons in the app to control the LED or any other device connected to the ESP32 board. [8]

7.2. Connect ESP RainMaker with Alexa

To Connect ESP RainMaker with Alexa App, follow these steps:

- **Download and Install the Amazon Alexa Application:** First, get the Amazon Alexa app. It is available on Google Play Store or the App Store. Download and set up it on your phone.
- **Add devices to ESP RainMaker:** Before connecting to Amazon Alexa, make sure you have already added devices to the ESP RainMaker app. Otherwise, follow the steps mentioned within the applicable post or the commands for connecting devices to ESP RainMaker.
- **Connect the ESP RainMaker to Alexa:** Open the ESP RainMaker app and go to "Settings". Tap on "Voice Services". Select "Amazon Alexa" to select. Click on "Link with Amazon Alexa".
- **Complete communication plan:** Click "LINK". Log in to your ESP RainMaker account or select "Continue with Google" if desired. Next, launch the Amazon Alexa app on your smartphone. Go to "Devices" then "Switches". You should see all the devices listed there that you have added to the ESP RainMaker app.

- **Control devices from the Alexa app:** If your ESP32 device is connected to Wi-Fi, you can now control the device from the Alexa app. Just select on the device you want to control in the Alexa app.
- **Voice Control:** Voice commands can also be used to control devices. For example, when controlling devices with your voice, speak "Alexa, Turn ON Switch." [8]

7.3. Connect ESP RainMaker with Google Home App

To Connect ESP RainMaker with Google Home App, follow these steps:

- **Download and Install Google Home App:** You can get the app from Google Play Store, then download and install it on your device.
- **Create a New Home in Google Home App:** Open the Google Home app. In the upper left corner, there's a "+" icon you can click on. Choose "Create new home" on the top menu. Then add a nickname along with your home address and click "Continue" to create one.
- **Add Devices in ESP RainMaker:** Before connecting to Google Home connect to ESP RainMaker it is required that you add your devices. Otherwise, follow the steps mentioned within the applicable post or the commands for the installation of an ESP Rain maker device.
- **Connect ESP RainMaker with Google Home App:** Go to settings in the google home app and follow the instructions. Click "Works with Google" option for seamless sharing and synchronization. Enter "ESP RainMaker" into search and choose it. In the ESP RainMaker account, log in or click "continue". Now go to the homepage of Google Home. Here, all the devices which we added in the ESP RainMaker app will appear.
- **Control Appliances from Google Home App:** After your ESP32 has been connected to the internet to the Wi-Fi, you basically engage the control of your devices by tapping on the appliances you want to control via the Google Home app.
- **Voice Control with Google Assistant:** In addition to that, you may control your appliances via voice commands with the Google Assistant. Simply say "Okay, Google, Turn ON Switch". [8]

8. Future Scope

Given the current scenario, we can develop a cross platform system that can be used on different platforms like iOS and Windows. Ideally, sensors can be added to control home appliances such as; PIR for motion detection and used for alarm buzzing for safety purposes. The scope of this project can be expanded to more areas by confining it to small offices, college labs rather than the house.

9. Conclusions

It is obvious from this project work that an individual control home automation system can be made from affordably available components and can be used to control multiple home appliances ranging from the lights to the cooling system and even the entire house lighting system. And better still, the necessary components are minimal and few that they can be packaged into a small unobtrusive container. The designed home automation system was tested a number of times to manage different home appliances used in the lighting system, air conditioning system, and many more. Hence, this system is scalable and flexible.

Acknowledgment

I acknowledge and convey my deep sense of gratitude towards all the authors of research papers whose contribution in this field helped me to prepare my paper. I thank all those who have contributed towards preparation of the same. I thank the developers which helped in developing block diagrams, circuit

diagrams and platforms. Additionally, special thanks to Prof. Manjiri Pathak who were our project guide and co-guide, for their support, guidance, and expertise throughout this project.

References

[1] Mamta, A. Paul and R. Tiwari, "Smart Home Automation System Based on IoT using Chip Microcontroller", 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 564-568, https://doi.org/10.23919/INDIACom54597.2022.9763287

[2] R. Niranjana, A. S, V. M and V. S, "Effectual Home Automation using ESP32 NodeMCU", 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 1-5. https://doi.org/10.1109/ICACRS55517.2022.10028992

[3] S. Dash and P. Choudekar, "Home Automation using Smart Devices and IoT", 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5. https://doi.org/10.1109/ICRITO51393.2021.9596533

[4] U. Singh and M. A. Ansari, "Smart Home Automation System Using Internet of Things", 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 2019, pp. 144-149. https://doi.org/10.1109/PEEIC47157.2019.8976842

[5] M. Al-Kuwari, A. Ramadan, Y. Ismael, L. Al-Sughair, A. Gastli and M. Benammar, "Smart-home automation using IoT-based sensing and monitoring platform", 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-PowerEng 2018), Doha, Qatar, 2018, pp. 1-6. https://doi.org/10.1109/CPE.2018.8372548

[6] Mao, J., Lin, Q., & Bian, J. (2018). Application of learning algorithms in smart home IoT system security. Mathematical Foundations of Computing, 1(1), 63-76.

[7] M. Das, P. Das, S. Das, & E. Biswas. (n.d.). "IoT Based Home Automation System". Technical Report, RCCIIT, 2020, GR3. [Retrieved December 22, 2023]. https://www.rcciit.org/students_projects/projects/ee/2020/GR3.pdf

[8] S. (2023, April 20). "ESP32 IoT Project using Alexa Google Home app – 2024", IoT Circuit Hub. https://iotcircuithub.com/esp32-iot-project-using-alexa-google-home/ [Accessed on November 17, 2023].

[9] Sovani, K. (2021, December 14). "Introducing ESP RainMaker - The ESP Journal", Medium. https://blog.espressif.com/introducing-esp-rainmaker-8fa968bbb2de [Accessed on February 20, 2024].

[10] "Introduction to ESP32". (n.d.). Tutorials Point. https://www.tutorialspoint.com/esp32_for_iot/esp32_for_iot_introduction.htm [Accessed on September 12, 2023].

[11] DHT11–Temperature and Humidity Sensor. (n.d.). Components101. https://components101.com/sensors/dht11-temperature-sensor [Accessed on September 12, 2023].