ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 6, Issue 2, 122-135

Exploring the Development of a User-Friendly Home Management System with ESP Rainmaker Firmware

Matthew Oluwole AROWOLO, Adekunle Okikioluwa OWOLABI, Henry Onaopemipo BANKOLE-AJAYI, Oluwaseyi Paul AKINYELE

[1]*Department of Mechatronics Engineering, Federal University, Oye, Ekiti State, Nigeria*
arowolo.oluwole@fuoye.edu.ng/adekunle.owolabi.171085@fuoye.edu.ng/henry.bankole-ajayi.171037@fuoye.edu.ng/oluwaseyi.akinyele.171025@fuoye.edu.ng

Corresponding Author: arowolo.oluwole@fuoye.edu.ng, +2348038066827

Date Submitted: 11/10/23
Date Accepted: 22/10/23
Date Published: 02/11/23

Abstract: *With the rapid development of the Internet of Things (IoT), managing living spaces has become more intuitive than ever. This research illustrates the implementation of a user-friendly home management system utilizing the advanced ESP Rainmaker firmware from Espressif Systems. The study focuses on creating a seamless user experience through the efficient integration of ESP Rainmaker's features. Various hardware components are carefully selected to facilitate the creation of user-friendly smart devices, including intelligent lighting systems, switches, and climate control units. The specialized firmware tailored to the unique attributes of each device ensures secure connections with the ESP Rainmaker cloud services. The research delves into comprehensive provisioning techniques, including Bluetooth and SoftAP, to simplify device onboarding and ensure secure connectivity. Leveraging the Rainmaker cloud services, the system emphasizes the enhanced user control and synchronization of device states, contributing to an average success rate of 88% and an average response time of 2.1 seconds. This study provides valuable insights into the development of intuitive and secure home automation systems, underscoring the significance of integrated platforms like ESP Rainmaker in shaping the future of IoT-enabled living environments.*

Keywords*: Home, Automation, ESP rainmaker, System, Devices, Firmware*

1. INTRODUCTION

The rapid evolution of technology has revolutionized the way households interact with their living spaces, giving rise to the integration of home automation as a pivotal element within modern residences [1]. Leveraging the capabilities of the Internet of Things (IoT), this integration has unlocked a plethora of possibilities for enhancing user comfort, convenience, and energy efficiency through the utilization of smart devices and systems [2]. Notably, the implementation of the ESP Rainmaker firmware from Espressif Systems has emerged as a transformative solution in this sphere [5, 6]. By streamlining the complex process of home management, the ESP Rainmaker firmware offers a user-friendly platform for creating a seamless home automation experience. This integration facilitates not only enhanced control and interconnectivity but also provides avenues for increased quality of life for the elderly and disabled individuals, eliminating the need for extensive infrastructure changes commonly associated with advanced home automation systems [4]. However, challenges persist in the design and implementation of such systems, calling for scalable solutions with user-friendly interfaces, diagnostic services, and cost-effective measures to ensure widespread adoption [3, 7]. This paper aims to explore the design and execution of a portable home automation system, emphasizing the utilization of the ESP Rainmaker firmware as a central component in creating a secure, efficient, and user-centered automated home environment.

Reviewing some previous work, [8] designed and implemented a Wi-Fi-based home automation system. The developed prototype allowed users to control and monitor the home through Wi-Fi by using Arduino Mega integrated with an Android-based application known as Virtuino. However, the prototype had limited connectivity, capable of performing only local control. The remote control of the developed system was suggested to be enabled based on IoT, allowing users to control it using a web server even when they were not around their house. A similar system using Arduino Mega with IoT was presented by [9]. The interconnected system, consisting of an Arduino microcontroller, was utilized to connect to an Ethernet shield, which was connected to a modem with an Internet connection. A relay was linked to the devices and through an HTML page with an assigned IP address. However, the system did not consider home surroundings and was not fully implemented.

Gupta and Chhabra designed and implemented an SH intelligent system based on Ethernet to monitor power consumption in real time through tracking devices in the home using an Intel Galileo Gen 2 board, which can be used in homes and communities [10]. This study presented a smart and intelligent system for energy management and security based on IoT with an independent and portable power control, where users can oversee the power management and security of their homes even when they are not around. Kaur et al., also provided information on home automation and security systems using different techniques, such as Arduino and GSM and Android applications to control home appliances [11]. Badabaji and Nagaraju presented a system that managed home appliances through IoT, where the temperatures, fire, and gas were controlled by using different sensors, and their values were displayed on an LCD [12].

According to ElShafee and Hamed, Wi-Fi based home automation system is presented. PC (with built in Wi-Fi card) based web server was made used. It manages all connected home devices [13]. The users can manage and control the system locally (LAN) or remotely (internet). A wide range of home automation devices like power management components and security components was supported by the system. A similar system was proposed in Caytiles and Park where the actions are coordinated by the home agent running on a PC. Other papers such as [15-18] also presented internet-controlled systems consisting of a dedicated web server, database and a web page for interconnecting and managing the devices [14]. These systems utilize a PC which leads to a direct increase in cost and power consumption. On the other hand, the development and hosting of the web page will also result in additional costs.

The design and implementation of a microcontroller-based voice activated wireless automation system is presented in Dutta et al. [19] The user speaks the voice commands through a microphone, which is processed and sent wirelessly via radio frequency (RF) link to the main control receiver unit. The features of the voice command are being extracted using a voice recognition module. This extracted signal is then processed by the microcontroller to perform the expected action. The setback noticed in the system is that, it can only be controlled from within the RF range. Jabbar et al., Presented the design, fabrication, and implementation of a portable, user-friendly, and cost-friendly automation system for Smart homes based on Internet of Things [20]. The developed IoT at home system can be easily implemented in a real house to allow real-time monitoring of home conditions and control of home appliances. Several sensors and actuators were connected to the NodeMCU controller, which updated the data to the IoT server. The obtained data from the sensors (temperature, humidity, motion, gas, and RFID) can be monitored via MQTT Dash mobile application and Adafruit IO Web via laptops/PC. For security and safety purposes, the user receives notifications on their mobile phones about any abnormal condition at home via the IFTTT server. Control of home appliances can be easily and efficiently conducted by using MQTT/Adafruit IO GUI or through voice commands using Google Assistant.

In the context of this study, the predominant challenges lie in ensuring portability, cost-effectiveness, responsiveness, flexibility, and user-friendliness of the implemented home automation system. The study aims to address these challenges directly. Many households in Nigeria require extensive modifications to their sensors and actuators before integrating any home automation solution. However, this novel system is designed to enable a seamless integration of emerging technologies at a minimal cost and reduced risk, thereby benefiting numerous households. Notably, the system demonstrates the capability to control multiple devices and automate functionalities based on parameters such as temperature, motion, and light intensity.

2. METHODOLOGY

This section describes the methodology adopted in this study, which includes systematic organization of different research phases in conjunction with the detailed design and implementation of the IOT home automation system and prototype. In addition, the selection of components and their integration are explained to fulfill the design objectives.

2.1 Conceptual Framework of the System

The research starts by identifying the problems encountered in existing home automation systems. Most considerable problems of the available systems in the market are their high initial implementation costs and unfriendly user interfaces. The modeling phase focuses on the selection of materials and components for building the house model prototype and developing the Automation system. The house model plan is designed on Adobe software, and the house model is fabricated using fine plastic material and flexi glass. The design and implementation of the IoT Home automation system are conducted. Wiring and connection between different components (bulb, fan, motor, and sensor) in the house model prototype to the attached home automation system (ESP32, relay board, DC source, and others) are installed and tested. After the connection of the microcontroller and components, coding is performed to realize the required tasks and then flashed into the microcontroller. The system is then linked up to the ESP Rainmaker cloud service by scanning the QR code generated. After that, Wifi Provisioning was done either by Bluetooth (BLE) or SoftAP. The design is then reviewed again to see whether the system's functionalities are affected by any issues. To confirm the efficacy of the system, testing is done. When an issue is discovered, the system goes back to the improvement and optimization phase. When the system performs well, it is considered complete.

2.2 Components and Materials Used

i. ESP-WROOM-32 Micro-controller: The ESP32 microcontroller was successfully used in industrial automation, primarily for the deployment of embedded systems and numerous IoT applications.
ii. 16-channel Relay Module: To manage high voltage, high current loads such motors, solenoid valves, lights, and AC loads. Majorly used to switch the devices.
iii. Water-Level sensor: Detects an excessively high or low liquid level in a stationary container.
iv. DHT11 Temperature and Humidity Sensor: Measures the surrounding air and temperature.

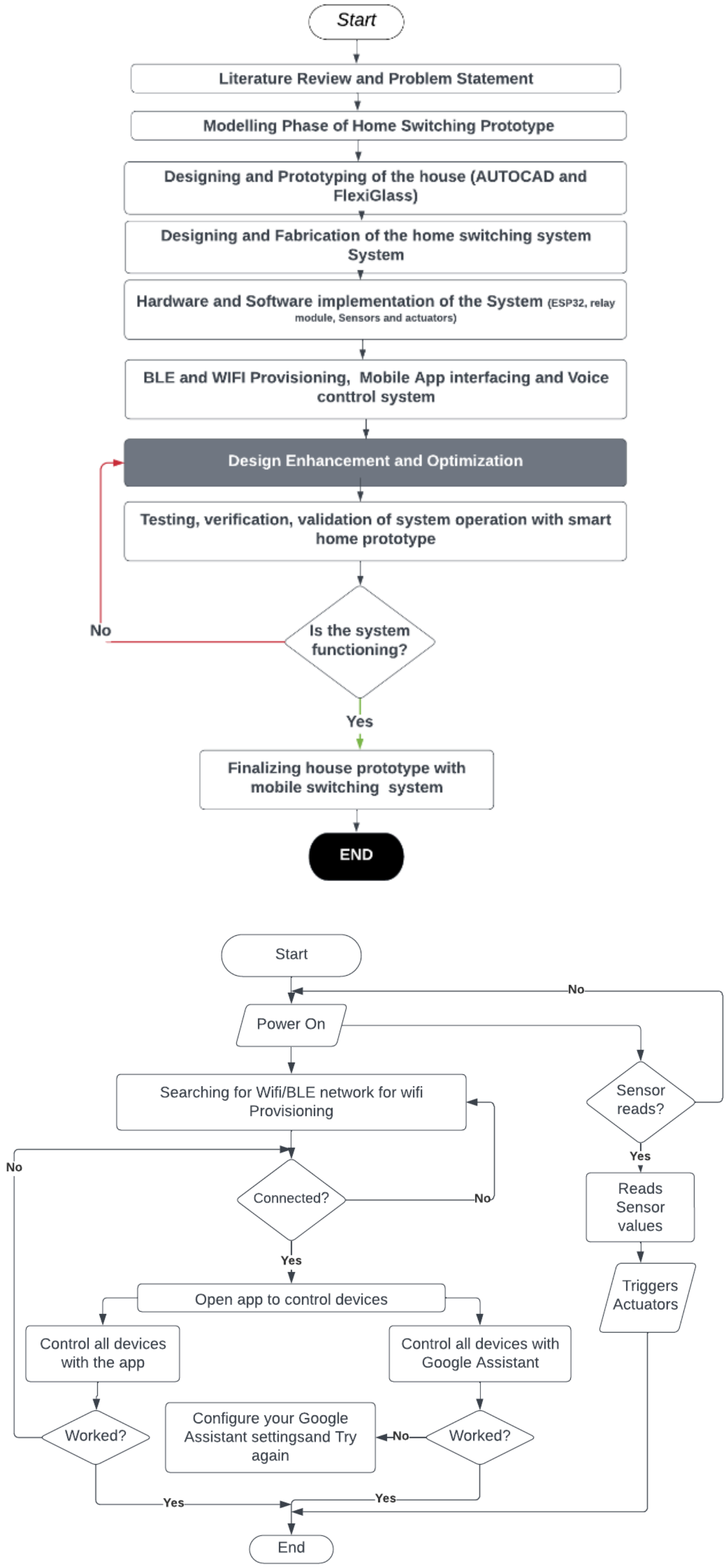

Figure 1: The research activities and working principle flowchart of this study.

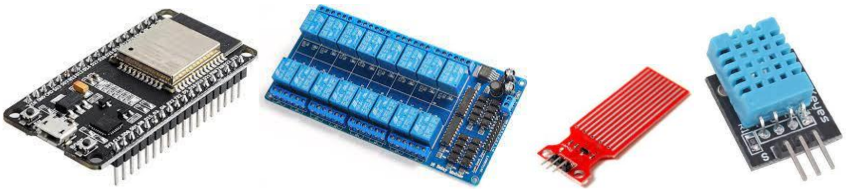

Figure 2: ESP-WROOM-32, 16-channel relay module, water-level sensor, DHT11 sensor

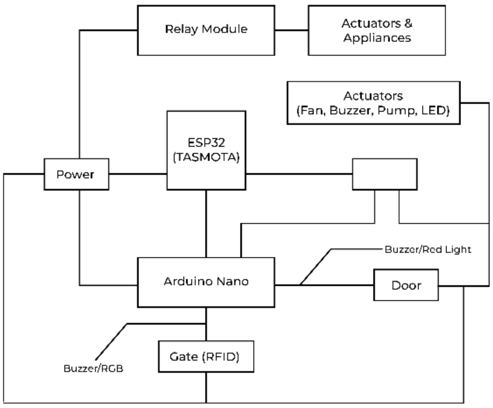

Figure 3: General block diagram

Figure 4: Home ground plan

Figure 5: Pictures of the house model

2.3 Overall System Model

The System Architecture (General Block Diagram) is as shown in Figure 2.2. The setup involves the physical assembly and configuration required to integrate them into the home automation ecosystem. This process involves several steps:

i. Device Assembly: Physically assembling the devices involves connecting components, sensors, and actuators according to the manufacturer's instructions. Ensuring proper wiring and connections are essential to device functionality and user safety.
ii. Firmware Flashing: Loading the ESP RainMaker firmware onto the microcontrollers of the selected devices is a critical step. The firmware provides the devices with the necessary software framework to communicate, connect to the network, and interact with the RainMaker cloud services.
iii. Network Configuration: Configuring network settings, such as Wi-Fi credentials, is vital for the devices to connect to the home network. Proper configuration ensures seamless communication between devices and cloud services.
iv. Provisioning: Utilizing the provision methods supported by the ESP Rainmaker firmware, such as Bluetooth and SoftAP provisioning, to securely onboard devices to the network and cloud services.
v. Cloud Service Integration: Linking the devices to the Rainmaker cloud services to enable functionalities like device discovery, remote control, and state synchronization.
vi. Testing and Calibration: Verifying the functionality of each device and ensuring it responds as expected to user commands and interactions. Calibrating sensors and actuators, if necessary, to achieve accurate and consistent performance.

Figure 2.4 signifies the ground floor plan of the model including the device and sensor positioning and the dimension is 70cm x 65cm. Figure 2.5 includes pictures of the final model design with several architectural materials for its aesthetics.

2.4 Device Control

2.4.1 Hardware control

For the 8-channel relay module, D23, D22, D21, D19, D18, D5, D25 & D26 GPIO was used to control the 8 relays. And the GPIO D13, D12, D14, D27, D33, D32, D15 & D4 are connected with switches to control the relay module manually. The D2 (GPIO-2) pin was used as the Wi-Fi indicator. (D2 pin connected with the blue LED, mounted on ESP32 board).

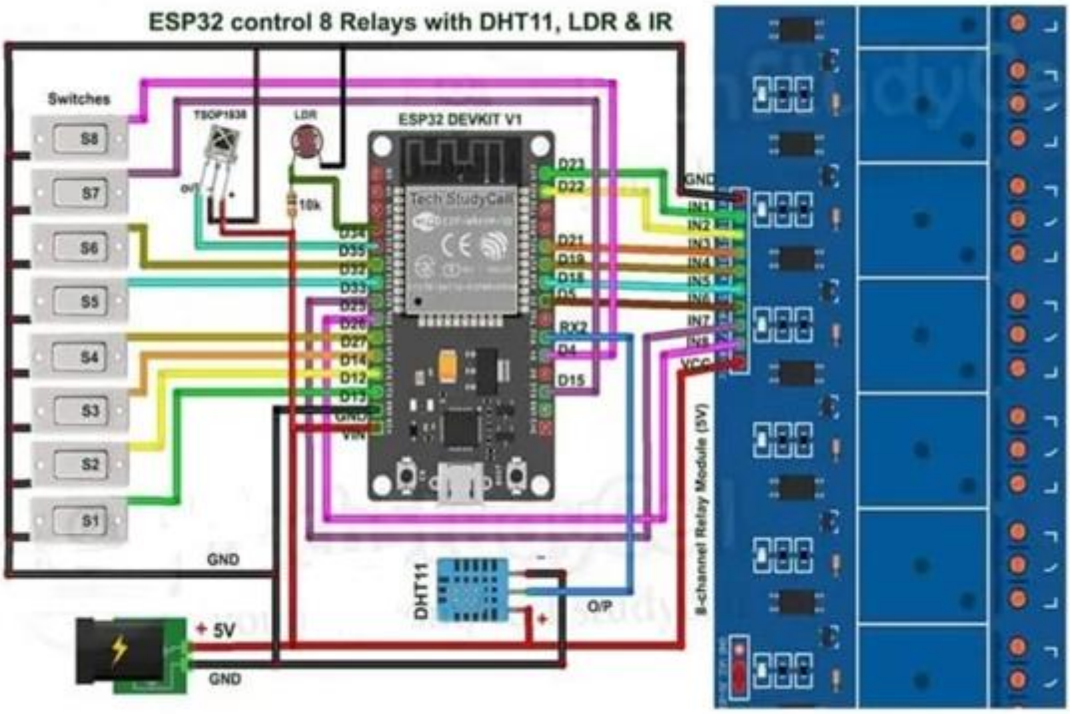

Figure 6: Schematic diagram of the connection

2.4.2 App control

ESP Rainmaker app is a lightweight IoT Cloud app, fully integrated into the AWS server less architecture, which allows customers to build, develop and deploy customized IoT solutions with a minimum amount of code and maximum security. They are Open source, offering a reference design and development guidance. It has features like User Management, User-device association, Device/Group Sharing, Local/Remote Control, Scenes and automations, Time-series Data, Grouping, Voice-Assistant Integration, Scheduling and Triggers.

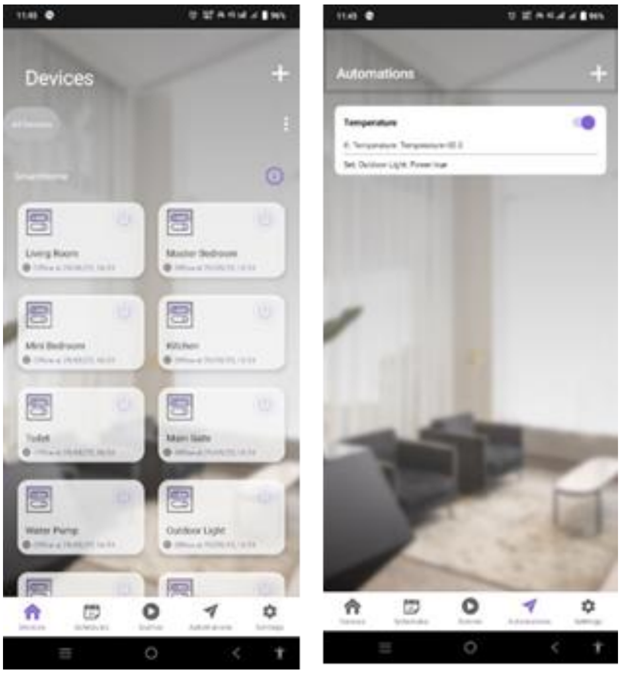

Figure 7: Mobile app interface

2.4.3 Gate Control

The RFID (Radio Frequency Identification) security gate system is a crucial part of the smart home infrastructure and is intended to improve access control and security on the property. This system makes use of RFID technology to give anybody trying to get in an easy and secure way to authenticate. RFID tags or cards, RFID readers, a backend system, and the security gate mechanism itself make up the core components of the RFID security gate system. In order to maintain the system's efficiency and smooth functioning, each component is essential. Authorized persons are given RFID tags or cards that have a unique identification linked to their access credentials. These tags are frequently compact and lightweight, making it easy for users to carry them. The RFID readers placed at the entrance point are in charge of finding RFID tags that are within their detection range. A person presents their RFID tag or card to the RFID scanner as they approach the security barrier. When reading a tag, the reader extracts its unique identification and transmits it to the backend system for validation. The received identifier is compared by the backend system to a database of authorized users and their access privileges.

The backend system sends a signal to activate the security gate mechanism, enabling the person to enter the building if the identify matches an authorized user. On the other hand, admission is refused and the gate stays locked if the identify is not recognized or is not allowed. The RFID security gate system has a number of advantages. First of all, it offers a hassle-free and contactless mode of verification, enabling customers to enter quickly by merely swiping their RFID tag or card over the reader. This increases user convenience by doing away with the requirement for manual key entry or direct contact with the gate. The technology also improves security by lowering the possibility of unwanted access. RFID tags are challenging to copy or fabricate because of their distinctive identifiers, which lowers the possibility of uninvited visitors entering. The backend system may also be set up to record and monitor access events, creating a useful audit trail for security reasons.

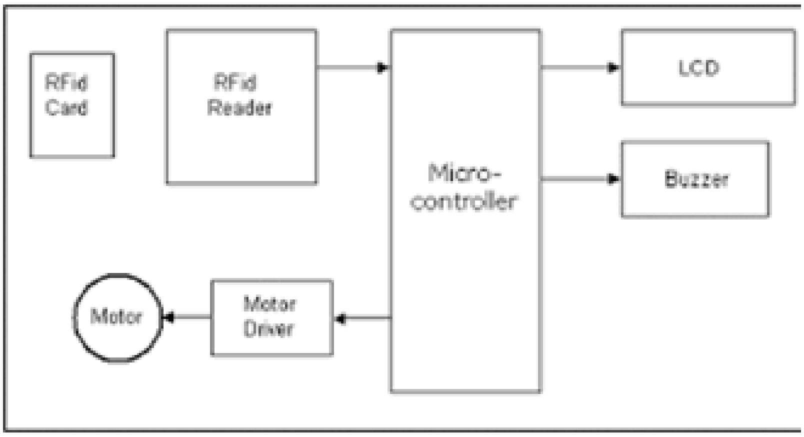

Figure 8: RFID gate control system

2.4.4 Water-level management system

A key component of a smart home system is water level control, especially for applications like irrigation systems, water tanks, and water monitoring. It entails keeping an eye on and managing the water level to guarantee effective use and avoid waste. Water-level management often uses actuators (such as valves or pumps) and microcontrollers for data processing and control in the hardware infrastructure. To correctly detect the water level, water-level sensors are positioned at key locations. Depending on the specific needs and the type of water storage or monitoring system, these sensors may be capacitive, ultrasonic, or pressure-based. To calculate the current water level, the microcontroller analyses the data it receives from the water-level sensors. The microprocessor transmits instructions to the actuators to open or close valves, switch on or off pumps, or engage other water control mechanisms based on established thresholds or user-defined parameters. This makes it possible for the smart home system to control water flow, set irrigation schedules, and optimize water use depending on current circumstances. A smart home system that controls water levels provides advantages including effective irrigation, water saving, and the avoidance of problems like overflow or shortage. Users may automate and monitor water consumption, assuring optimal use and environmental sustainability, by integrating water-level sensors, actuators, and microcontrollers into the smart home infrastructure.

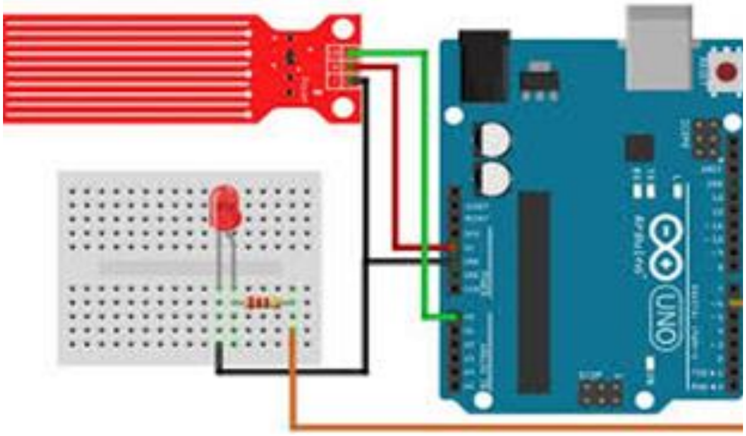

Figure 9: Water-Level circuit

2.5 Firmware Development and Integration with Rainmaker

A crucial step in making smart devices that communicate successfully, react to human input, and enhance the user experience is developing the firmware for those devices. The firmware development process begins with defining the intended functionality of the smart device. Then the design and implementation of the programming logic that governs the device's behavior. Firmware development includes mechanisms for configuring device-specific settings using QR code.

The functionality and user experience of the home automation system are largely dependent on the effective integration of smart devices with the RainMaker cloud services. This integration makes it easier for customers to connect with their smart home environment through a single interface by facilitating smooth communication, remote control, and synchronization among devices.

ESP RAINMAKER

Provisioning Code: Please scan with your ESP RainMaker app.

Payload: {"ver":"v1","name":"PROV_1234RB8","pop":"12345RB8","transport":"ble"}

This QR code is unique for your device. You may print a copy of this for subsequent use.

This QR code is generated using qrcodejs

Figure 10: ESP rainmaker QR code provisioning

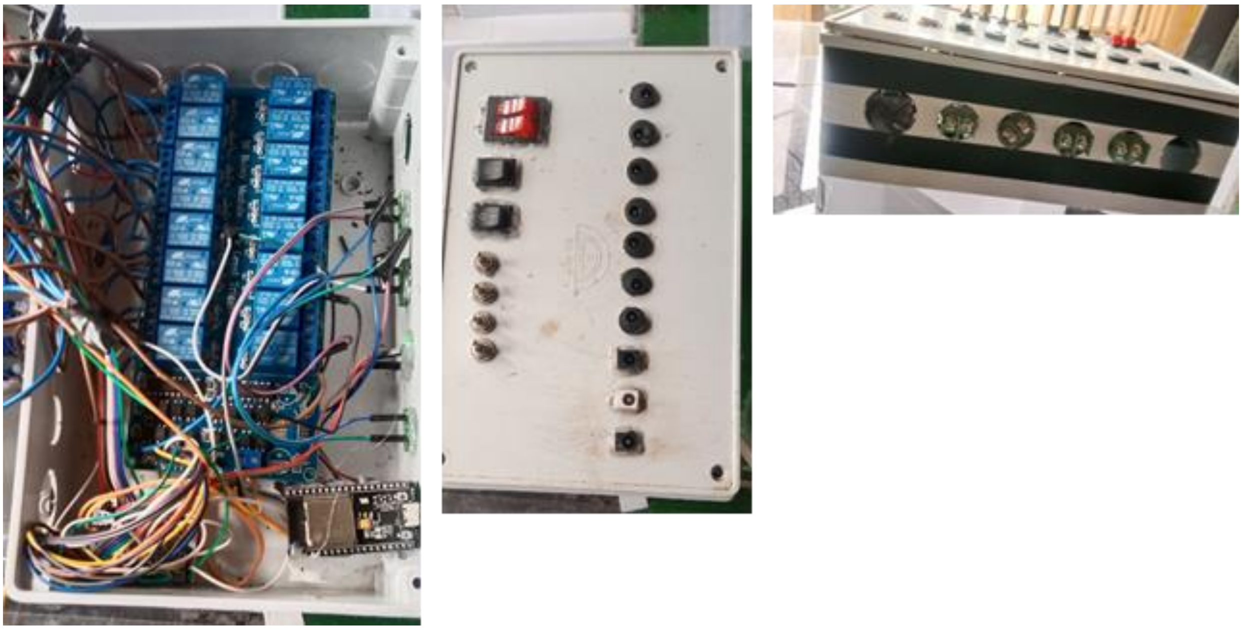

Figure 11: Pictures of the portable home automation system

2.6 Device Implementation

The main contribution of this work is the development of an innovative market-ready portable controller that can be implemented in real houses to continuously monitor home conditions and comfortably manage home appliances through the Internet regardless of time and place. In addition to the proposed algorithm (Algorithm 1) for home automation and control that implemented in ESP32 microcontroller. The employment of several sensors, including motion, gas, RFID, ultrasonic, and temperature and humidity sensors, ensures the technology is safe and intelligent. Before beginning the final implementation and production of the created system, several experiments for individual sensors and actuators are carried out on the breadboard. As a result, the right materials and parts are chosen for the system. Because the controller is portable, it may be installed (plug and play) in the built prototype and is appropriate to be attached to a real house. Connectors are utilized on the wire terminals for simple plug-and-play. After the hardware implementation test, the controller system is installed in the SH prototype and put through another round of testing. When an error occurs, the system is improved and optimized until it can perform as it is intended. The sensor and relay in the created portable automation system are operating with the two microcontrollers without any issues.

Algorithm1- Home Switching system including voice control

```
Variables:
- livingRoomStatus
- masterRoomStatus
- miniRoomStatus
- toiletStatus
- kitchenStatus
- waterPumpStatus
- mainGateStatus
- fanStatus
- currentTemperature
Main Loop:
  Repeat indefinitely:
    Read currentTemperature
    if currentTemperature >= 30:
      TurnOnFan()
    Read userVoiceCommand
    Parse userVoiceCommand
    if userVoiceCommand contains "living room":
      ToggleLivingRoom()
```

Algorithm2- Sensor Control

```
Variables:
motionSensorStatus
buzzerStatus
waterLevelStatus
waterPumpStatus
buzzerStatus
gasSensorStatus
buzzerStatus
rfidStatus
gateStatus
Main Loop:
  Read motionSensorStatus
    if motionSensorStatus == MOTION_DETECTED:
  TurnOnBuzzer()
  WaitForTime(DELAY_DURATION)
  TurnOffBuzzer()
  Read waterLevelStatus
    if waterLevelStatus == LOW:
```

Algorithm1- Home Switching system including voice control

```
    else if userVoiceCommand contains "master room":
      ToggleMasterRoom()
    else if userVoiceCommand contains "mini room":
      ToggleMiniRoom()
    else if userVoiceCommand contains "toilet":
      ToggleToilet()
    else if userVoiceCommand contains "kitchen":
      ToggleKitchen()
    else if userVoiceCommand contains "water pump":
      ToggleWaterPump()
    else if userVoiceCommand contains "main gate":
      ToggleMainGate()
    else:
      Print "Command not recognized"
Functions:
- ToggleLivingRoom():
  If livingRoomStatus == OFF:
    TurnOnLivingRoom()
  else:
    TurnOffLivingRoom()
- ToggleMasterRoom():
  If masterRoomStatus == OFF:
    TurnOnMasterRoom()
  else:
    TurnOffMasterRoom()
- ToggleMiniRoom():
  If miniRoomStatus == OFF:
    TurnOnMiniRoom()
  else:
    TurnOffMiniRoom()
- ToggleToilet():
  If toiletStatus == OFF:
    TurnOnToilet()
  else:
    TurnOffToilet()
- ToggleKitchen():
  If kitchenStatus == OFF:
    TurnOnKitchen()
  else:
    TurnOffKitchen()
- ToggleWaterPump():
  If waterPumpStatus == OFF:
    TurnOnWaterPump()
  else:
    TurnOffWaterPump()
- ToggleMainGate():
  If mainGateStatus == OFF:
    TurnOnMainGate()
  else:
    TurnOffMainGate()
- TurnOnFan():
  Set fanStatus to ON
  Turn on the fan
- TurnOnLivingRoom():
  Set livingRoomStatus to ON
```

Algorithm2- Sensor Control

```
  TurnOnBuzzer()
  TurnOnWaterPump()
  WaitForTime(PUMP_DURATION)
  TurnOffWaterPump()
  WaitForTime(DELAY_DURATION)
  TurnOffBuzzer()

else if waterLevelStatus == HIGH:
  TurnOnBuzzer()
  WaitForTime(DELAY_DURATION)
  TurnOffBuzzer()

Read gasSensorStatus
   if gasSensorStatus == GAS_DETECTED:
  TurnOnBuzzer()
  WaitForTime(DELAY_DURATION)
  TurnOffBuzzer()
Read rfidStatus
   if rfidStatus == CORRECT_RFID:
  OpenGate()
WaitForTime(GATE_OPEN_DURATION)
  CloseGate()
Functions:
TurnOnBuzzer(): Set buzzerStatus to ON Turn on the buzzer
TurnOffBuzzer(): Set buzzerStatus to OFF Turn off the buzzer
TurnOnWaterPump(): Set waterPumpStatus to ON Turn on the water pump
TurnOffWaterPump(): Set waterPumpStatus to OFF Turn off the water pump
OpenGate(): Set gateStatus to OPEN Open the gate
CloseGate(): Set gateStatus to CLOSED Close the gate
WaitForTime(duration): Wait for the specified duration before proceeding
```

Algorithm1- Home Switching system including voice control	**Algorithm2- Sensor Control**

```
    Turn on the living room lights
- TurnOnMasterRoom():
    Set masterRoomStatus to ON
    Turn on the master room lights
- TurnOnMiniRoom():
    Set miniRoomStatus to ON
    Turn on the mini room lights
- TurnOnToilet():
    Set toiletStatus to ON
    Turn on the toilet lights
- TurnOnKitchen():
    Set kitchenStatus to ON
    Turn on the kitchen lights
- TurnOnWaterPump():
    Set waterPumpStatus to ON
    Turn on the water pump
- TurnOnMainGate():
    Set mainGateStatus to ON
    Open the main gate
- TurnOffFan():
    Set fanStatus to OFF
    Turn off the fan
- TurnOffLivingRoom():
    Set livingRoomStatus to OFF
    Turn off the living room lights
- TurnOffMasterRoom():
    Set masterRoomStatus to OFF
    Turn off the master room lights
- TurnOffMiniRoom():
    Set miniRoomStatus to OFF
    Turn off the mini room lights
- TurnOffToilet():
    Set toiletStatus to OFF
    Turn off the toilet lights
- TurnOffKitchen():
    Set kitchenStatus to OFF
    Turn off the kitchen lights
- TurnOffWaterPump():
    Set waterPumpStatus to OFF
    Turn off the water pump
- TurnOffMainGate():
    Set mainGateStatus to OFF
    Close the main gate
```

3. Results and Discussion

3.1 Model Testing Results

3.1.1 Accuracy

Accuracy analysis is a systematic process used to evaluate how closely a system's actual performance matches its expected or intended behavior. It involves comparing the observed outcomes of a system or algorithm with the expected outcomes to determine the degree of accuracy in achieving the desired results.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Observed\ outcome} \times \mathbf{100}}{\boldsymbol{Expected\ Outcome}} \tag{1}$$

Table 1 gives the results obtained from performing accuracy test on Algorithm 1 (Home Switching System) on the 8 devices for a frequency of 10.

Table 1: Accuracy analysis for algorithm 1

S/N	Expected Outcome	Observed Outcome	Accuracy (Success Rate) %
1	8	6	75
2	8	7	87.5
3	8	8	100
4	8	5	62.5
5	8	8	100
6	8	8	100
7	8	7	87.5
8	8	8	100
9	8	8	100
10	8	7	87.5
	Average Success Rate		**90%**

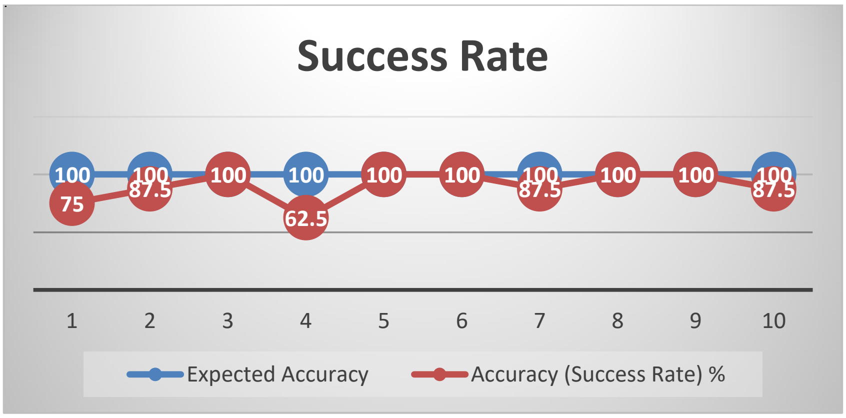

Figure 12: Success rate of model

From Table 12, it is observed that all devices were tested 10 times to determine the success rate of the model and the average Success rate gotten is 90%, which shows the accuracy level of the model, based on these results, it can be deduced that 90 out of 100 attempts will be correctly actualized.

Table 2 gives the results obtained from performing accuracy test on Algorithm 2 (Sensor Control) on all sensor performance for a frequency of 10.

Table 2 Accuracy analysis of algorithm 2

S/N	Sensor	Expected Outcome	Observed Outcome	Success Rate%
1	Motion	10	10	100
2	DHT	10	7	70
3	Water-level	10	8	80
4	RFID	10	10	100
5	Gas	10	9	90
		Average Success Rate		**88%**

Figure 13: Graph showing the accuracy analysis for algorithm 2

From Table 2, it is noticed that all sensors were tested 10 times to determine the success rate of the model and the average Success rate gotten is 88%, which shows the accuracy level of the model, based on these results, it can be inferred that 88 out of 100 attempts will be correctly triggered.

3.1.2 Response time

Response time is a critical factor in assessing the efficiency, usability, and overall performance of systems and applications. Table 3 gives are response rate i.e., measuring the time taken from the moment the user taps the app icon to the moment there is a response from the device or the time it takes for an actuator to actuate when a sensor senses.

Table 3 Response Time analysis

S/N	Device	Response Time
1	Lights	2.45seconds
2	Buzzer	1.7seconds
3	Water Pump	3.6seconds
4	Gate	2.1seconds
5	Buzzer(gas)	2.3seconds

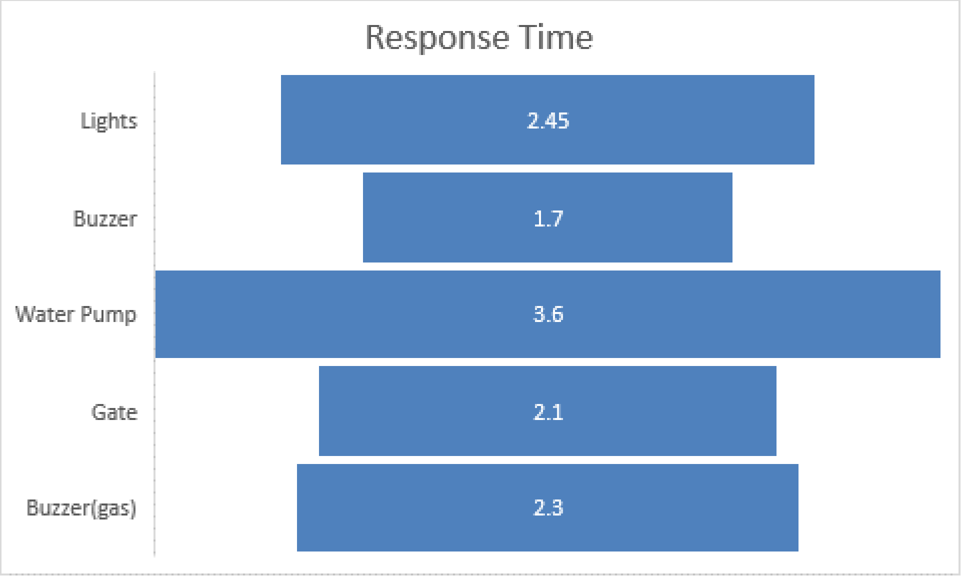

Figure 14: Graph showing the response time for the model

3.1.3 Failure analysis

Failure analysis was conducted for key components of the home automation system which involves identifying the causes of failures, assessing their impact, and proposing corrective and preventive measures.

Table 4 Failure analysis

S/N	Failure	Frequency	Cause	Solution
1	Gas Sensor Algorithm Failure	2	a bug in the algorithm logic that caused it to skip the buzzer activation step	Updated algorithm code to include the missing edge case handling.
2	Water Pump Motor Failure	2	Insufficient power supply to the water pump motor.	Rectified power supply issues to ensure adequate voltage for the motor.
3	Fan Failure	3	Loose connection in the wiring leading to the motor.	Conducted thorough inspection and fixed loose wiring connections.

The failure analysis highlighted the critical importance of robust testing and thorough code reviews to identify and rectify edge cases that might lead to failures. By addressing this issue promptly, the gas sensor algorithm's performance and reliability were significantly improved. The failure analysis underscored the significance of not only software but also hardware components in system reliability. Regular maintenance and monitoring are crucial to prevent unexpected failures and ensure uninterrupted system operation.

3.2 Discussions

The results emphasize the success of the home automation system using ESP RainMaker firmware. The outcomes validate the sensor control algorithms' accuracy, responsiveness, and reliability. This study not only advances the field of home automation but also lays the groundwork for future developments that align with the evolving demands of smart living. This study's implementation of the ESP Rainmaker firmware on the ESP-WROOM-32 microcontroller showcased the potential of transforming a conventional home into a smart and automated one. As part of the limitations of the study, the control can only be configured to one Wi-Fi and Bluetooth network. Also, there is a sensor device limit that can work or be compatible with ESP Rainmaker.

4. CONCLUSION

This study culminated in a comprehensive set of findings that highlight the successful development, testing, and evaluation of the home automation system using ESP Rainmaker firmware. Through meticulously crafted sensor control algorithms, there was accurate and reliable performance across various scenarios, effectively enabling motion detection, water level sensing, gas detection, and RFID-based access control. Algorithm optimization efforts further fine-tuned response times, ensuring prompt and efficient system responses while accommodating diverse user needs. The findings affirm the system's accurate algorithms, responsive performance, reliability, and practicality. The study's outcomes contribute to the advancement of smart living, offering users a secure, customizable, and efficient home automation solution.

ACKNOWLEDGMENT

This study would not have been possible without the extraordinary assistance and support provided by numerous individuals and organizations. We sincerely thank all these contributors. Additionally, sincere gratitude is extended to senior mentors and other research colleagues whose important contributions were crucial to the successful publication of this study.

REFERENCES

[1] Kumar, S., Tiwari, P. & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, *6*(111). https://doi.org/10.1186/s40537-019-0268-2

[2] Alahi, M. E. E., Sukkuea, A., Tina, F. W., Nag, A., Kurdthongmee, W., Suwannarat, K. & Mukhopadhyay, S. C. (2023). Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends. *Sensors*, *23*(11), 5206 https://doi.org/10.3390/s23115206

[3] Bodea, M. B., Anghelescu, P., Gavriloaia, B. M. & Vizireanu, D. N. (2019). Design and Implementation of a Home Automation System. *Proceedings of the 11th International Conference on Electronics, Computers and Artificial Intelligence*, *ECAI 2019*(*10)*, 1–5. https://doi.org/10.1109/ECAI46879.2019.9041956

[4] El Jaouhari, S. & Bouvet, E. (2022). Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions. *Internet of Things*, *18*, 100508. https://doi.org/10.1016/J.IOT.2022.100508

[5] Pereira, G. & Chaari, M. Z. (2023). Comparison of Blynk IoT and ESP Rainmaker on ESP32 as beginner-friendly IoT solutions. *International Conference on Internet of Things*. https://doi.org/10.1007/978-3-031-23582-5_9

[6] Pereira, G. P., Chaari, M. Z. & Daroge, F. (2023). IoT-Enabled Smart Drip Irrigation System Using ESP32. *IoT*, *4*(3), 221–243. https://doi.org/10.3390/iot4030012

[7] Stolojescu-Crisan, C., Crisan, C. & Butunoi, B. P. (2021). An iot-based smart home automation system. *Sensors, 21*(11), 1–23. https://doi.org/10.3390/s21113784

[8] Jabbar W. A., Alsibai M. H., Amran N. S. S. & Mahayadin S. K. (2018). Design and Implementation of IoT-Based Automation System for Smart Home, *Proceedings of the 2018 International Symposium on Networks, Computers and Communications*, 1–6.

[9] Imran S. S, Vignesh J., Singh V. K. & ArunPrasath D. T. (2016). Smart Home Automation Based on IoT using Arduino Mega, *Proceedings of the International Conference on Current Research in Engineering Science & Technology*, 2348–8379.

[10] Gupta P. & Chhabra J. (2016), IoT-based Smart Home Design using Power and Security Management, *Proceedings of the 2016 IEEE International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 6– 10.

[11] Kaur, S., Singh, R., Khairwal, N. & Jain, P. (2016). Home Automation and Security System. *Advanced Computational Intelligence: An International Journal (ACII)*, *3*(3), 17–23. https://doi.org/10.5121/acii.2016.3303

[12] Badabaji, S. & Siva Nagaraju, V. (2018). An IoT-Based Smart Home Service System. *International Journal of Pure and Applied Mathematics*, *119*(16), 4659–4667. http://www.acadpubl.eu/hub/

[13] Elshafee, A. & Hamed, K. A. (2012). Design and Implementation of a WiFi Based Home Automation System. *World Academy of Science, Engineering and Technology*, *6*(8), 1856–1862.

[14] Caytiles, R.D. & Park, B. (2012). Mobile IP-Based Architecture for Smart Homes, *International Journal of Smart Home*, 6, 29-36.

[15] Alkar, A. Z. & Buhur, U. (2005). An Internet Based Wireless Home Automation System for Multifunctional Devices. *IEEE Transactions on Consumer Electronics*, *51*(4), 1169–1173. https://doi.org/10.1109/TCE.2005.1561840

[16] Liang, N. S., Fu, L. C. & Wu, C. L. (2002). An Integrated, Flexible, and Internet-Based Control Architecture for Home Automation System in the Internet Era. *Proceedings - IEEE International Conference on Robotics and Automation*, *2*(May), 1101–1106. https://doi.org/10.1109/robot.2002.1014690

[17] Rajabzadeh, A., Manashty, A. R. & Jahromi, Z. F. (2010). A Mobile Application for Smart House Remote Control System. *World Academy of Science, Engineering and Technology*, *62*. http://arxiv.org/abs/1009.5557

[18] Sharma U. & Reddy S. R. N. Design of Home/Office Automation using Wireless Sensor Network. *International Journal of Computer Applications* 43(22), 53-60.

[19] Dutta, K. P., Rai, P. & Shekher, V. (2012). Microcontroller Based Voice Activated. *VSED International Journal of Electrical, Electronics & Communication Engineering*, *2*(8), 642–649.

[20] Jabbar, W. A., Kian, T. K., Ramli, R. M., Zubir, S. N., Zamrizaman, N. S. M., Balfaqih, M., Shepelev, V. & Alharbi, S. (2019). Design and Fabrication of Smart Home with Internet of Things Enabled Automation System. *IEEE Access*, *7*(September), 144059–144074. https://doi.org/10.1109/ACCESS.2019.2942846