Design and Implementation of Smart Home System Based on IoT and Esprainmaker

B. Santhikiran[1], L. Nagaraju[1,*], Sk. Abdul Sattar[1], D. Bharath Chandra[1], Y. Jayasankar[1]

Abstract: Smart home systems have revolutionized the way people interact with their living spaces. The Internet of Things (IoT) has made it possible to connect various devices to a central hub, enabling homeowners to control and automate their homes from a single device. In this paper, we present the design and implementation of a smart home system based on IoT and the Esprainmaker platform, which allows for easy integration of different devices. The system uses Bluetooth and Wi-Fi technology through mobile app to enable homeowners to control and monitor their homes from anywhere. The app provides an intuitive user interface that allows users to control various devices, including lights, cameras, and appliances. The system also includes sensors that detect environmental conditions, such as temperature and humidity, and automatically adjust the home's settings. The implementation of the system involves the use of Esprainmaker, an open-source platform that provides libraries and tools for developing IoT applications. The platform allows for easy integration of various devices and provides a framework for building custom applications. The application is designed to be intuitive and easy to use, with features such as scheduling and automation. The proposed smart home system provides a cost-effective and easy-to-use solution for controlling and monitoring various devices in the home. It is designed to be scalable, and additional sensors and devices can be added to the system as needed. The system can also be integrated with various voice assistants such as Amazon Alexa and Google Assistant, which provides users with even more convenience and control.

Keywords: Smart home, IoT, ESPRainmaker, Bluetooth Low Energy, Sensors.

1. Introduction

In recent years, the Internet of Things (IoT) has become increasingly popular, and it has been widely used in various applications, including smart homes.

Article History
Received: 10-05-2023; Revised: 20-06-2023;
Accepted: 25-06-2023

*Corresponding author: L. Nagaraju

[1]Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, Vijayawada 520008, India.
E-Mail: bsk.aliet@gmail.com
nagarajuram03@gmail.com
abdulsattarshaik4512@gmail.com
bharathchandra309@gmail.com
yjayasankar08092@gmail.com

Smart homes provide a convenient and efficient way to control and monitor home appliances, lighting, temperature, security, and energy usage. The aim of this paper is to present the design and implementation of a smart home system based on IoT and ESPRainmaker. ESPRainmaker is an open-source platform that simplifies the development and management of IOT devices. The smart home system is a network of devices that are connected and controlled by the internet [1]. These devices can communicate with each other and can be remotely controlled using a smartphone or a computer. The system is built on the Internet of Things (IOT) technology, which allows devices to be connected to the internet and communicate with each other without human intervention. ESP32 is a low-cost, low-power microcontroller board with built-in Wi-Fi and Bluetooth capabilities. It is widely used in IOT applications and has gained popularity in recent years.

The board is built on the Ten silica Xtensa LX6 microprocessor, which has a dual-core processor, and a variety of interfaces such as Wi-Fi, Bluetooth, and Ethernet [2]. The board can be programmed using the Arduino IDE or Micro Python, which makes it easy to develop IOT applications. create these components, incorporating the applicable criteria that follow.

ESP Rainmaker is a framework for building IOT applications that run on the ESP32 board. It provides an easy-to-use interface for developing and managing IOT devices. The framework includes a cloud-based platform that allows for remote control of devices, over-the-air firmware updates, and cloud-based analytics. It also provides a set of pre-built components, such as temperature and humidity sensors, light sensors, and relays, that can be easily integrated into IoT applications. The smart home system based on ESPRainmaker and ESP32 board can be used to control various devices such as lights, thermostats, and security systems. The system can be customized to meet specific needs and preferences, and can be integrated with other smart home platforms. It provides a convenient and efficient way to manage and automate home devices, making life easier and more comfortable. One of the key advantages of this system is its flexibility. The ESP32 board can be programmed to control a wide variety of devices, and the ESPRainmaker framework provides a cloud-based platform for managing and monitoring these devices. This makes it easy to create a custom smart home system that meets the needs of the user [3].

In summary, the smart home system based on IoT and ESPRainmaker using the ESP32 board is a powerful and flexible platform for building custom smart home systems. It provides users with a centralized platform to control and monitor their home appliances remotely using their smartphones or tablets. It provides an easy-to-use interface for developing and managing IoT devices, and can be customized to meet specific needs and preferences. It is a cost-effective and efficient solution for building a custom smart home system.

2. Literature review

There have been several related works on the design and implementation of smart home systems based on IOT. One study proposed a system architecture that used sensors to monitor the environment and control various devices in the home, while another focused on developing a user-friendly mobile app for controlling smart home devices. Additionally, there have been several works exploring the use of machine learning algorithms to improve the efficiency and effectiveness of smart home systems. We did a good research on the papers of Home Automation and the various technologies implemented on the past. Some of the designs briefed here [4].

K Lova Raju in his paper implemented home automation using NodeMCU with IOT. The appliances are controlled by using the Blynk application. In his project Blynk application is used as the transmitting device and the NodeMCU which is equipped in the electrical appliances acts as receiving device. Based on the commands received from the Blynk application to the NodeMCU through the internet the electrical appliances should be on required condition [5]. The draw back of his system is to control the appliances through manual mode only from the longer distance through the internet. Swetha Amit in her paper implemented the home automation system, the commands are given through either Alexa or mobile to the Node MCU controller.

3. Proposed system

The idea behind a smart home system based on IoT (Internet of Things) and ESP32 microcontroller running the ESPRainMaker firmware is to provide homeowners with a more efficient and convenient way of controlling various home appliances and systems using the internet. A Smart Home is able to control and monitor the home even through the person is not available in the home. The internet of things system can be formed by collaborating the ESP32 microcontroller board with the other components like DHT11 and LDR connected to the microcontroller board which is used to collect the data from the surroundings like humidity, Temperature and light intensity [6]. This collected data send to the esp32 microcontroller board and then send it to the ESPrainmaker application through the internet. The ESPrainmaker application is a cloud-based platform for building and managing IoT devices. It provides a set of tools and services that enable developers to easily build and deploy IoT solutions. One of the key features of the ESPRainmaker platform is the ability to create timer schedules and automations. Timer

schedules allow the user to set specific times for your device to perform certain actions [7]. For example, user can set your device to turn on every day at 7am and turn off at 8pm. User can create multiple schedules for different days of the week or for specific dates. Automation allows the user to create rules for the device to perform actions based on certain conditions like temperature, light intensity, Humidity etc.

3.1 Block diagram

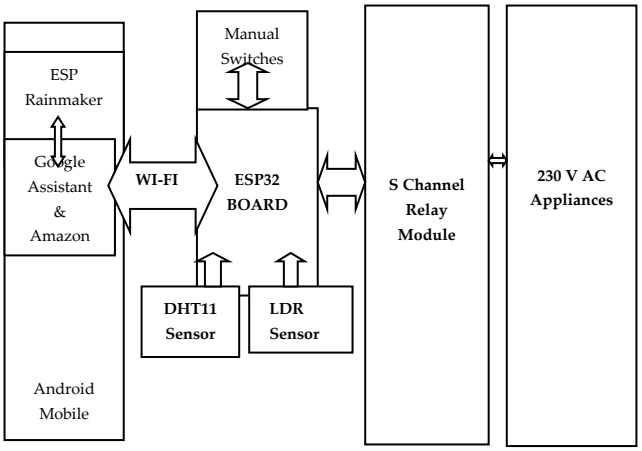

Fig. 1: Schematic of the Proposed System

User can create an automation that turns on your device when the temperature in a room reaches a certain level. You can create multiple automations for different conditions. ESPRainmaker supports voice services through integration with Amazon Alexa and Google Assistant. With this feature, users can control their devices by using simple voice commands. For example, a user can ask Alexa or Google Assistant to turn on/off a particular device or change its settings. ESPRainmaker provides various notification features that keep the users informed about the status of their devices. These notifications can be received in the form of push notifications on the user's smartphone. This platforms supports the various notifications like Device Notification, Users can receive notifications about their devices' status, such as whether a device is online or offline. Event Notification can configure notifications based on specific events, such as when a device's battery is low or when a particular sensor's value crosses a threshold [8]. Schedule Notifications, Users can receive schedule notifications to at a particular time, such as a daily report of their devices' status.

3.2 Circuit Diagram

ESP32 Module

The ESP32 is a powerful microcontroller board that was developed by Espressif Systems, a company based in Shanghai, China. It is widely used in various projects ranging from IoT devices to robotics and automation. The ESP32 board features a dual-core processor with a clock speed of up to 240 MHz, making it significantly faster than its predecessor, the ESP8266.

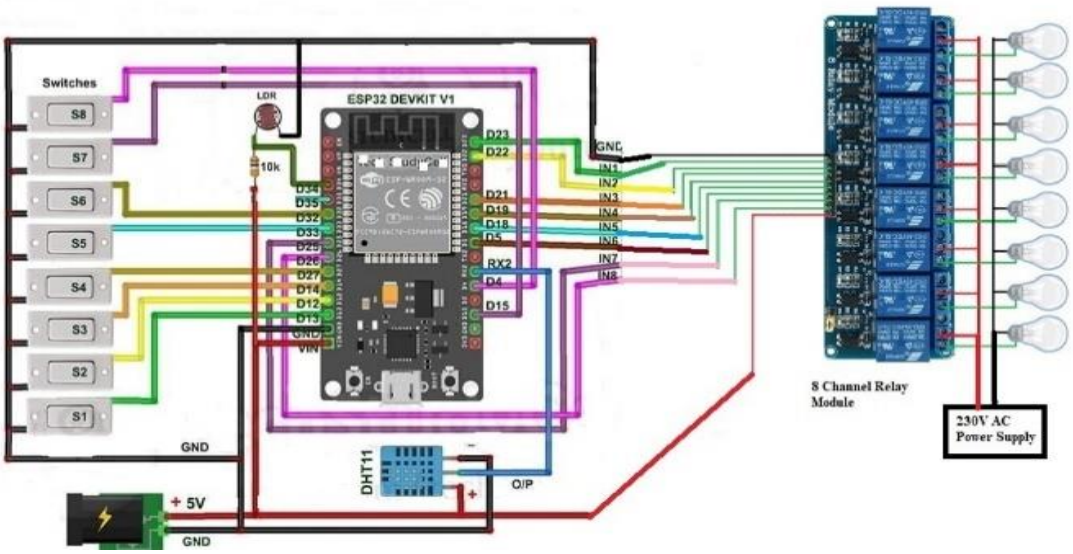

Fig. 2: Circuit Diagram

Fig. 3: ESP32 Board

The board also comes equipped with Wi-Fi and Bluetooth capabilities, making it an excellent choice for projects that require wireless connectivity. It also has a range of input/output pins, including digital and analog pins, as well as support for protocols such as SPI, I2C, and UART, making it highly versatile. Additionally, the ESP32 board features a wide range of built-in sensors, including an accelerometer, a gyroscope, and a temperature sensor. This makes it excellent choice for projects that require the use of these sensors. One of the most significant advantages of the ESP32 board is its low power consumption. It has several power-saving modes that allow it to operate for extended periods on battery power, making it an ideal choice for remote or low-power applications.

Relay

A relay is an electrically operated switch that allows a low-power signal to control a larger electrical circuit or device. Relays are commonly used in many applications where it is necessary to switch a high voltage or high current circuit using a low voltage or low current signal. In IoT projects, relays can be used to control a wide range of electrical devices such as lights, motors, heaters, and more.

Fig. 4: Relay Module

It can be used to turn devices on and off remotely or automatically in response to sensor readings or other events. Relays work by using an electromagnet to switch a set of contacts on or off. When a low voltage signal is applied to the coil of the relay, the coil generates a magnetic field that attracts a set of contacts, causing them to close. This completes the circuit and allows a larger current to flow through the relay and on to the connected device. When the signal is removed, the contacts return to their original position and the circuit is broken. IoT projects can use relays in a variety of ways, such as controlling the power supply to a device, turning lights on and off, or controlling the heating or cooling of a room. With the help of relays, IoT devices can be designed to be more flexible, efficient, and responsive to changing conditions.

DHT11 Sensor

DHT11 is a low-cost digital temperature and humidity sensor that is commonly used in various electronic projects. It is a basic sensor that is easy to use and is capable of measuring temperature between 0°C to 50°C (32°F to 122°F) with an accuracy of ±2°C, and relative humidity between 20% to 90% RH with an accuracy of ±5%.

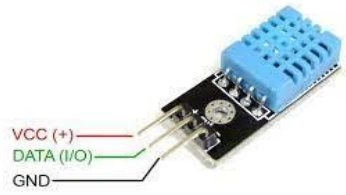

Fig. 5: DHT11 Sensor

The DHT11 sensor is made up of two parts - a capacitive humidity sensor and a thermistor, which work together to provide digital output for temperature and humidity. The sensor has a single-wire digital interface, which makes it easy to connect to microcontrollers such as Arduino, Raspberry Pi, NodeMCU and esp32 etc.

LDR Sensor

An LDR (Light Dependent Resistor) sensor is a type of electronic component that changes its resistance in response to changes in the intensity of light falling on it. LDRs are commonly used in various applications such as automatic streetlights, outdoor lighting systems, burglar alarms, and many more.

ESPrainmaker Application

The ESP Rainmaker application is a popular open-source framework used in the Internet of Things (IoT) domain to facilitate home automation. This application is built on top of the popular ESP-IDF (Espressif IoT Development Framework) and provides a set of APIs that allow developers to easily build smart home solutions. One of the key features of the ESP Rainmaker application is its ease of use. The framework provides a user-friendly interface that allows users to quickly set up and configure their smart home devices. One of the most significant benefits of using the ESP Rainmaker application is its ability to integrate with various IoT devices. The application provides support for a wide range of devices, including light bulbs, switches, and sensors, making it easy for developers to create complex home automation systems. Additionally, the application supports multiple communication protocols, including Wi-Fi and Bluetooth, which allows it to work with a wide range of devices. Another important feature of the ESP Rainmaker application is its ability to provide remote access to home automation systems. This means that users can control their home devices from anywhere in the world, as long as they have an internet connection. This feature is particularly useful for people who travel frequently and want to keep an eye on their home while they are away.

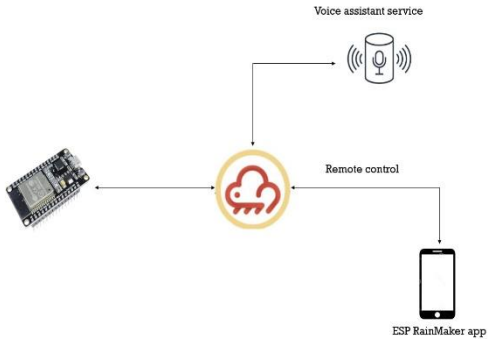

Fig. 6: Representation of entire Project Using ESPRainmaker.

The ESP Rainmaker application also supports voice control through popular voice assistants such as Amazon Alexa and Google Assistant. This allows users to control their smart home devices using their voice, making it even more convenient to use. In summary, the ESP Rainmaker application is a powerful tool for home automation. Its ease of use, support for a wide range of devices and communication protocols, remote access, and voice control make it an ideal choice for building complex smart home solutions.

Amazon Alexa Application

Amazon Alexa is a voice-controlled personal assistant application developed by Amazon. It is capable of performing various tasks, such as playing music, providing information, setting alarms, and controlling smart home devices. The application is compatible with a wide range of devices, including smartphones, tablets, and smart speakers. When it comes to IoT projects, Amazon Alexa can be an incredibly useful tool. It can be integrated with smart home devices, allowing you to control your lights, thermostat, and other appliances with voice commands. This can be particularly helpful for people with disabilities or those who have difficulty using

Fig. 7: Preview of Amazon Alexa App.

To use Amazon Alexa in your IoT projects, you'll need to have compatible devices and set up the Amazon Alexa application on them. You'll also need to ensure that your smart home devices are compatible with Amazon Alexa and that they have been set up correctly. Once everything is set up, you can start using Amazon Alexa to control your IoT devices. For example, you can ask Alexa to turn on the lights in your living room or adjust the temperature on your thermostat. You can also create custom routines that allow you to control multiple devices with a single voice command.

Google Assistant

Google Home is a smart speaker developed by Google that allows users to control their home devices with voice commands. The speaker is equipped with

Google Assistant, a virtual assistant that can answer questions, play music, and control smart home devices. Google Assistant is also available on other devices, such as smartphones and tablets, making it a versatile tool for interacting with the internet of things (IoT). Using Google Home and Google Assistant in IoT projects is relatively easy. First, you need to ensure that your smart home devices are compatible with Google Assistant. Most devices these days are compatible, but it's worth checking to make sure. Once you've confirmed compatibility, you can link your devices to your Google Home account through the Google Home app.

Fig. 8: Preview of Google Assistant.

Once your devices are linked, you can use voice commands to control them. For example, you could say "Hey Google, turn on the living room lights" or "Hey Google, set the thermostat to 70 degrees." You can also create routines, which are sets of actions that are triggered by a single command. For example, you could create a routine called "Goodnight" that turns off all the lights, sets the thermostat to a comfortable temperature, and plays soothing music. Overall, Google Home and Google Assistant are powerful tools for controlling and interacting with the internet of things. By linking your smart home devices to your Google Home account, you can create a seamless and intuitive experience for controlling your home devices with voice commands.

4. Design and implementation

Connect the ESP32 board to the internet via Wi-Fi. Connect the DHT11 sensor to the ESP32 board to measure temperature and humidity. Connect the LDR sensor to the ESP32 board to measure light intensity. Connect the appliances that need to be controlled to the relays which are connected to the ESP32 board. Install the ESPRainmaker library and the required dependencies in the Arduino IDE. Write code in the Arduino IDE to configure the ESP32 board to connect to Wi-Fi and use the ESPRainmaker library to register the board on the ESPRainmaker app and to read the sensor values and control the appliances based on the sensor readings is known as Automation in the ESPRainmaker.

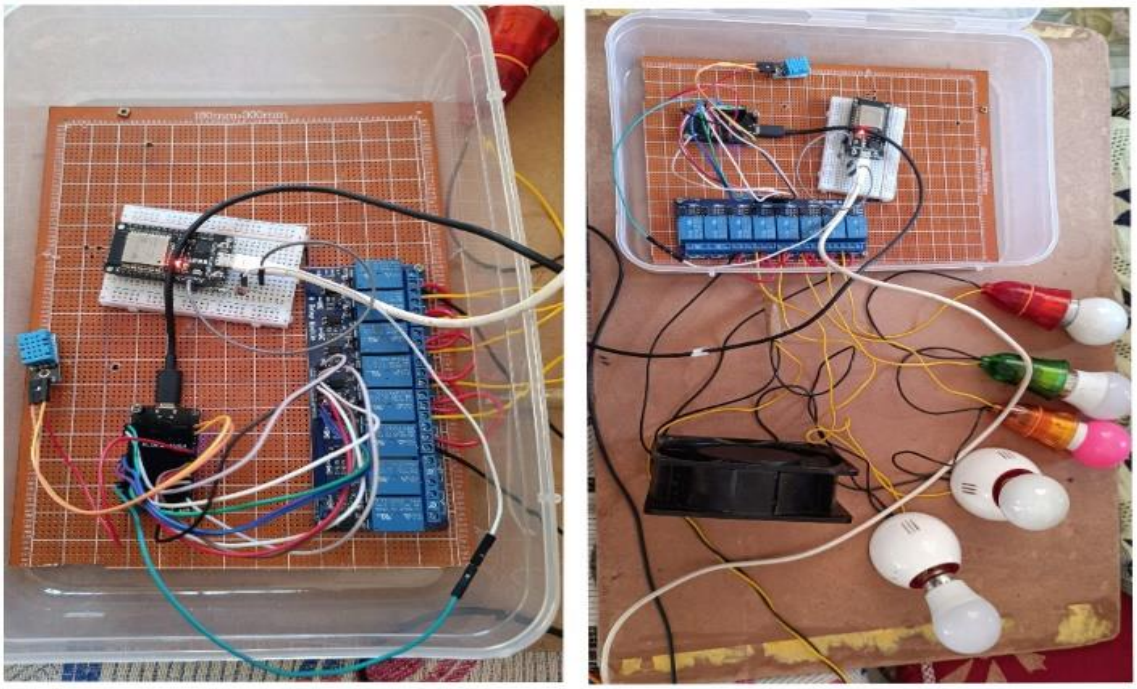

Fig. 7: Complete Setup of the Project

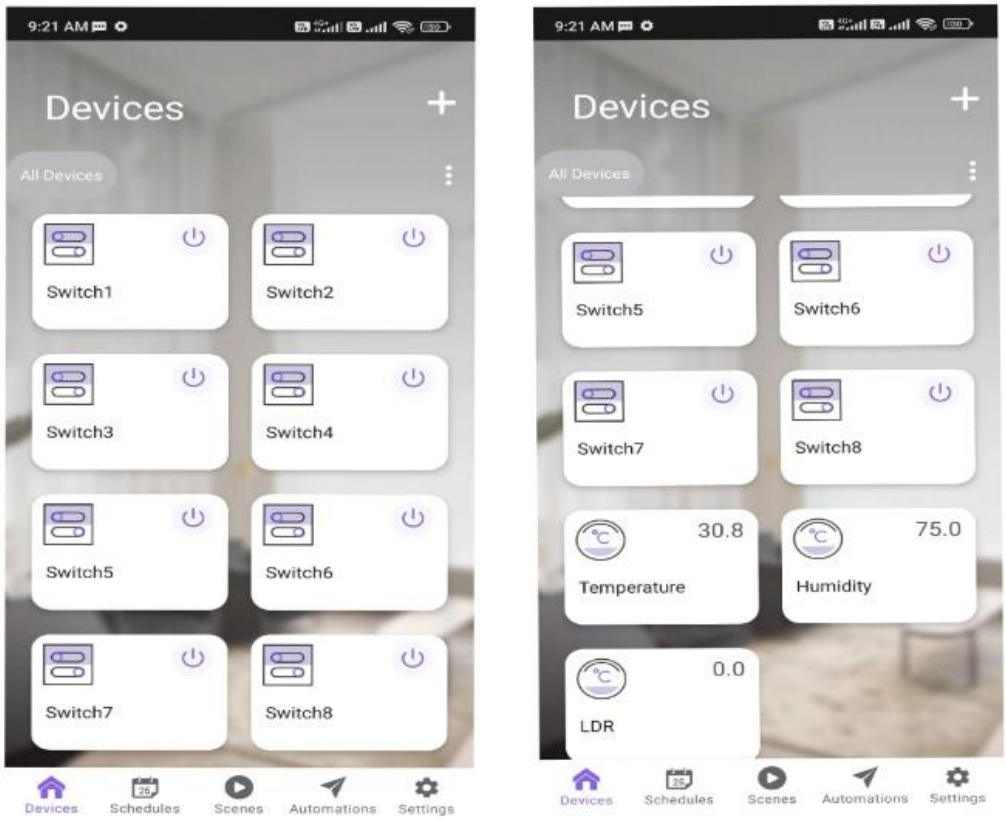

Fig. 8: Preview of ESPRainmaker Application

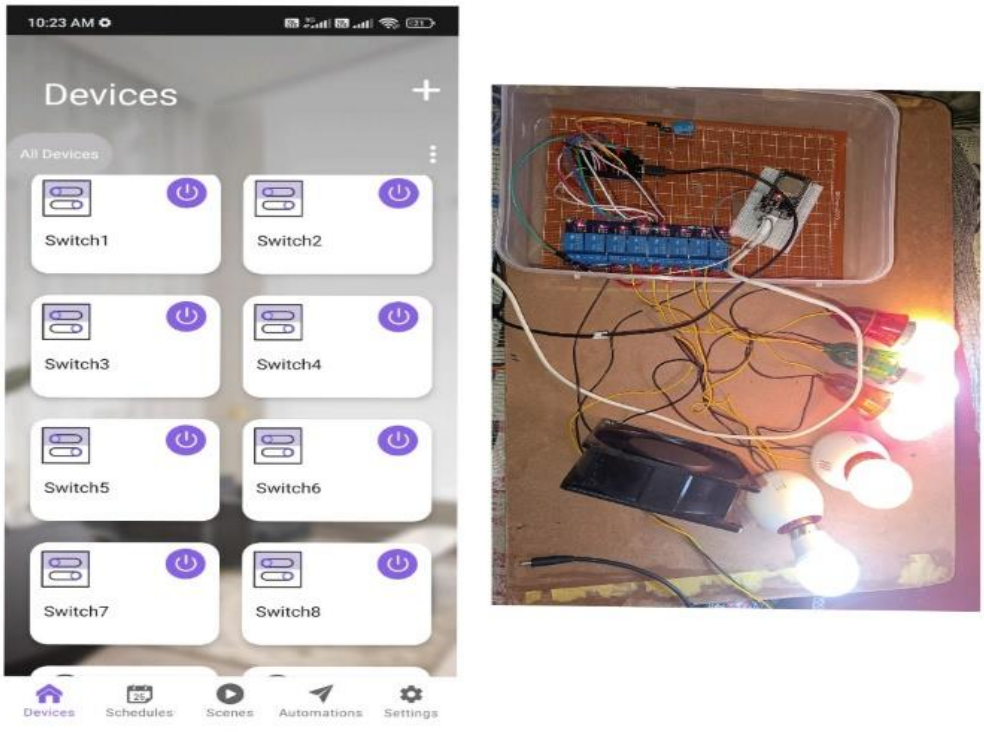

Fig. 9: Control the appliances through ESPRainmaker app.

To add the timer schedules in the ESPRainmaker then the appliances should be on the required conditions based on the timer schedules. Use the Amazon Alexa and Google Assistant to enable voice control of the appliances through these platforms. The ESPRainmaker app sends feedback to the user on the status of the appliances.

5. Result and Discussion

The main aim of this project is to control the home appliances like lights, fans, Ac and forth remotely by using the smart phone. By using the ESPRainmaker application one can control the electrical and electronic appliance from anywhere around the world. All the persons in that family can share the ESPRainmaker app and control the appliances simultaneously. So that, one person can turn on the appliance and other can do turn off that particular appliance. So everyone in the family can know the status of the Home.

In the above Fig 9 represents the control the hardware from the ESPRainmaker app running on our mobile device through the internet. The commands send from the ESPRainmaker app through the internet received by the ESP32 Board which equipped in the electrical appliances and appliances should be on the required condition. In the Fig 10 represents the control the devices at home through Google assistant and Alex app and thus we could implement home automation successfully. Timer schedules allow the user to set specific times for your device to perform certain actions. The above Fig 11 represents the by 11:05 AM switch 4 should be on. So every day at 11:05 AM switch 4 should be ON and get the Notification to the mobile until user can change the Event and action in the ESPRainmaker app. These features in the project is very use full for the users to save the time and the power of the Household appliances.

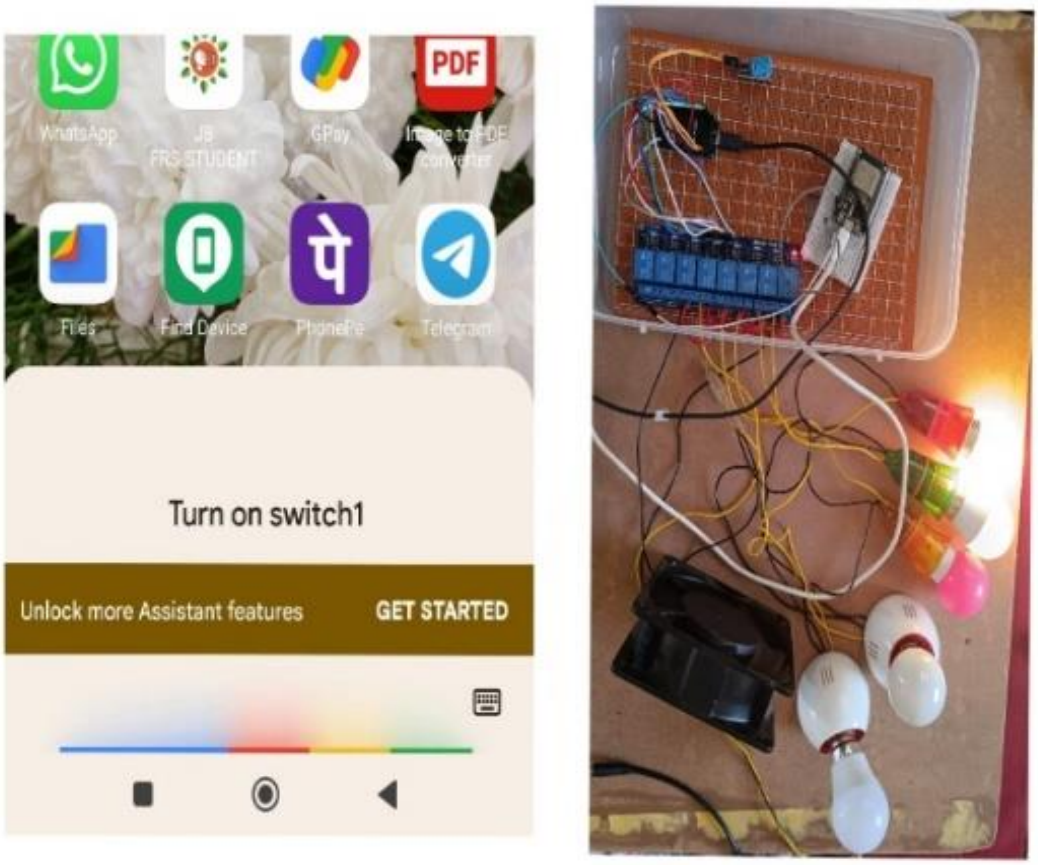

Fig. 10: Control the appliances through the Google assistant

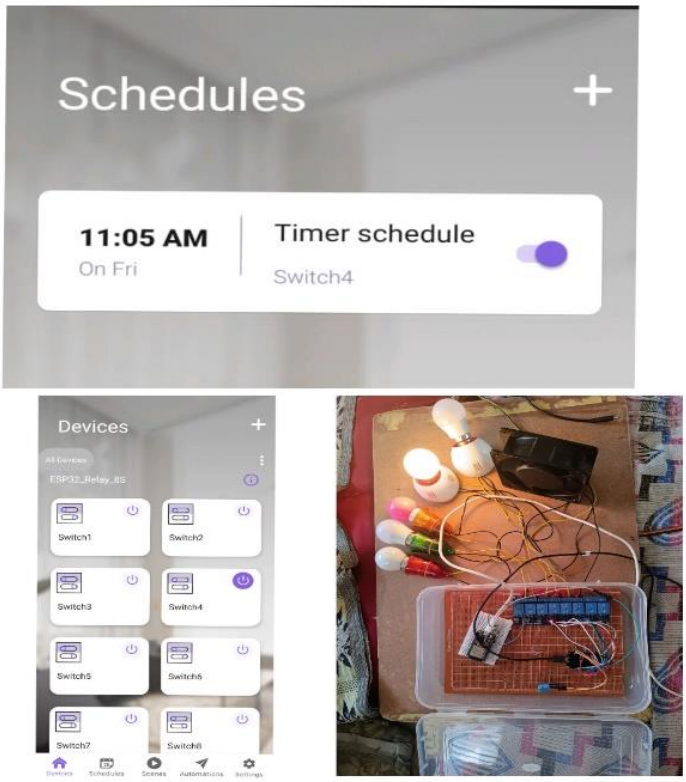

Fig. 11: Timer schedules

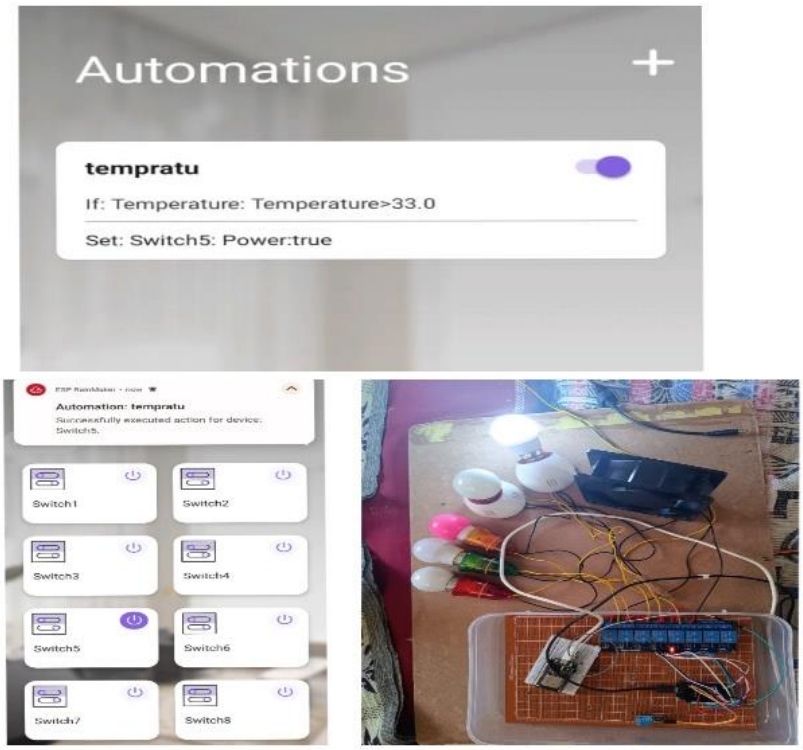

Fig. 12: Automation Based on Temperature value

Automation allows the user to create rules for the device to perform actions based on certain conditions like temperature, light intensity, Humidity etc. User can create an automation that turns on your device when the temperature in a room reaches a certain level. The above Fig 12 represents the if temperature greater than 330C switch 5 should be ON, whenever temperature reaches greater than 330C switch 5 is ON and get the Notification to the mobile.

6. Conclusion and future scope

In conclusion, the design and implementation of a smart home system based on IoT and the ESPRainmaker application has several advantages. It provides users with a centralized platform to control and monitor their home appliances remotely using their smartphones or tablets. The system also allows for automation of tasks and can help users save energy and reduce their utility bills. Overall, the future Scope of smart home systems based on IoT and the ESPRainmaker application is promising, and we can expect to see more innovative and sophisticated applications of this technology in the years to come.

Acknowledgment

The authors thank the Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, for providing the necessary facilities to carry out this research work.

Conflict of Interest

The authors declared "No conflict of interest"

References

[1] E. A. Mertens, L. F. S. Dias, F. A. Fernandes, B. D. Bonatto, J. P. G. Abreu and H. Arango, "Evaluation and trends of power quality indices in the distribution system," *9th International Conference on Electrical Power Quality and Utilization*, Barcelona, pp. 1-6, 2007.

[2] Y. Baris, A. A. Kayalar, M. B. Tosun, M. K. Ozcan and A. Z. Alkar "A GSM, internet and speech controlled wireless interactive home automation system", IEEE Transactions on Consumer Electronics, Vol. 52, No. 3, pp. 837-843, 2006.

[3] R. K. Lova, V. Chandrani, SK. S. Begum and M. P. Devi "Home automation and security system with node MCU using internet of things", In 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking, pp. 1-5, 2019.

[4] N. V. Sathya and S. Gayathri. "Design of Wireless Home automation and security system using PIC Microcontroller", International Journal of Computer Applications in Engineering Sciences, Vol. 3, No. 13, pp. 153-140, 2013.

[5] A. A. Waheed, N. Jan, S. Iqbal and C. Lee "Implementation of ZigBee-GSM based home security monitoring and remote control system", *In 2011 IEEE 54th International Midwest Symposium on Circuits and Systems* pp. 1-4, 2011.

[6] W. Eddie "A phone-based remote controller for home and office automation", IEEE Transactions on Consumer Electronics, Vol. 40, No. 1, pp. 28-34, 1994.

[7] P. Ravi, A. Saxena and B. Chanda "Implementation of smart home through fpga using verilog hardware descriptive language", *In 2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation*, pp. 1-6, 2020.

[8] G. T. Surya, I. R. H. Yaldi, M. Kartiwi, N. Ismail, "Prototype design of smart home system using internet of things", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 7, No. 1, pp. 107-115, 2017.

[9] K. Rohit, P. Mahamuni and Y. Parikh "Smart home system", *International Journal of Innovative Research in Advanced Engineering*, Vol. 2, No. 1, pp. 81-86, 2015.

Copyright: © 2023 by the authors, Licensee ITEECS, India. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

***