INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 526-531
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

HOME AUTOMATION USING ESP32 AND ACCESS POINT

Himanshi Dhokane[1], Vedant Kasar[2], Hariom Patil[3], Dr. S. I. Bakhtar[4]

[1,2,3]Third Year Student, Department of Electronic and Telecommunication Engineering, PRMCEAM, Badnera, Amravati, Maharashtra, India.

[4]Assitant Professor, Department of Electronic and Telecommunication Engineering, PRMCEAM, Badnera, Amravati, Maharashtra, India.

ABSTRACT

Home automation has emerged as a pivotal aspect of contemporary smart living, propelled by innovations in wireless communication technologies and the Internet of Things (IoT). This paper introduces a home automation system that employs the ESP32 microcontroller alongside an Access Point (AP) to enable remote control and monitoring of various household devices. The ESP32, known for its robust and adaptable features, includes integrated Wi-Fi and Bluetooth capabilities, acting as the central hub for managing connected devices such as lighting, fans, temperature sensors, and security systems. By incorporating an Access Point, the system facilitates smooth communication among devices and ensures secure, localized wireless control, thereby minimizing reliance on external cloud services and enhancing user privacy. Control of the system is achieved through an intuitive mobile application or a web interface, allowing for real-time monitoring and management of the home environment. Notable features of the proposed system encompass remote device management, sensor data collection, automated scheduling, and security notifications. This paper elaborates on the hardware and software architecture of the home automation system, the ESP32's role in promoting efficient communication, and the benefits of utilizing an Access Point for secure and localized network connectivity. Additionally, the paper explores the system's scalability for larger residences and the potential for incorporating further IoT devices. The proposed solution presents a cost-effective, dependable, and adaptable home automation platform suitable for a range of smart home applications.

Keywords: Home Automation, IoT, Access Point, ESP32

1. INTRODUCTION

The swift progression of technology has profoundly altered our interactions with our living environments, leading to the emergence of home automation. This innovation allows for the management and oversight of various home systems, including lighting, heating, cooling, security, and entertainment, through devices connected to the internet. The transition towards smart homes is largely fueled by the growing availability of Internet of Things (IoT) devices, which facilitate smooth communication among different appliances and systems within a home In recent times, the ESP32 microcontroller has attracted significant interest in the realm of home automation, owing to its affordability, high efficiency, and built-in Wi-Fi and Bluetooth features. As a robust and adaptable microcontroller, the ESP32 acts as the core component for a diverse array of smart home solutions. By incorporating an Access Point (AP), the ESP32 can establish local wireless communication among home devices, ensuring a secure and dependable connection without dependence on external cloud services. This setup fosters an effective, private, and scalable home automation network that can be conveniently managed via smartphones, tablets, or web interfaces. The objective of this research is to develop and implement a home automation system utilizing the ESP32 microcontroller in conjunction with an Access Point, facilitating local control and monitoring of various household appliances. The envisioned system seeks to improve the convenience and security of daily life by enabling users to remotely manage lighting, temperature, security cameras, and other devices, while also incorporating automation features for energy efficiency and user comfort. The combination of the ESP32 with an Access Point presents numerous benefits, including the removal of reliance on internet connectivity and cloud services, enhanced communication speed, and increased privacy by keeping data within the home network. This paper commences with an examination of the essential components and technologies integral to the system, emphasizing the function of the ESP32 microcontroller and the Access Point in facilitating device communication. It also outlines the necessary hardware and software architecture for constructing an effective and scalable home automation platform. Furthermore, the paper will investigate key system features, such as remote device management, real-time monitoring, sensor data collection, and automated scheduling. It will also assess the potential applications, limitations, and scalability of the system for residences of varying sizes In light of the growing interest in smart home technologies, the proposed system implementing home automation across diverse settings, thereby promoting more accessible and sustainable smart living solutions provides a cost-efficient, secure, and adaptable platform.

2. METHODOLOGY

The methodology involves hardware and software integration for Home Automation Using Esp32 and Access Point

The ESP32 is set up in Soft AP mode to establish a local Wi-Fi network. Users can connect to this network using a smartphone or computer to access a web-based control panel. This web application features buttons that allow users to manage four distinct appliances. When a button is pressed, an HTTP GET request is dispatched to the ESP32, activating the corresponding relay to turn the appliance on or off.

- The system architecture is organized into three primary layers:
- Device Layer – This layer includes the ESP32 and relay module, which interface with home appliances.
- Network Layer – The ESP32 functions as an access point and hosts the web server.
- Application Layer – A user-friendly web interface enables the control of appliances.

2.1 Hardware Components

ESP32 Module

The ESP32 is an economical and energy-efficient system-on-chip (SoC) microcontroller developed by Espressif Systems. Its robust capabilities, along with built-in Wi-Fi and Bluetooth connectivity, make it a popular choice for numerous applications, such as Internet of Things (IoT) devices, wearables, and smart technology, all while maintaining a compact design.

Overview of ESP32

The ESP32 is a 32-bit microcontroller that features built-in Wi-Fi and Bluetooth capabilities. As the successor to the widely-used ESP8266, it provides enhanced processing power, increased memory, and a greater variety of peripheral interfaces. The ESP32 includes a dual-core CPU and offers comprehensive support for various wireless communication protocols, positioning it as an excellent choice for Internet of Things (IoT) applications and other wireless connectivity solutions.

- **Key Features**
- Processor: Dual-coreXtensa® 32-bit LX6 processor, capable of clock speeds reaching 240MHz.
- Memory: Generally equipped with 520 KB of SRAM.
- Storage: Flash memory capacity of up to 16 MB, varying by model.
- Wireless Communication: Supports Wi-Fi (802.11 b/g/n) and Bluetooth (both Classic and BLE).
- General Purpose Input/Output: Accommodates up to 34 GPIOs for various functions including digital I/O, ADCs, DACs, PWM, and more.
- Security Features: Includes hardware encryption, secure boot, and flash encryption.
- Power Management: Offers several power modes suitable for low-power applications.
- Peripheral Interfaces: Features SPI, I2C, UART, PWM, ADC/DAC, CAN, SD card interface, among others.
- Dimensions: Typically measures 5mm x 5mm or smaller in module and development board formats

2.2 Software Components

Arduino IDE: Used for programming the ESP32 microcontroller.

- Arduino IDE with ESP32 Board Integration
- Wi-Fi Library for Configuring Access Points
- Web Server Library for Managing HTTP Requests
- HTML, CSS, and JavaScript for Web Interface Development

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 526-531
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

3. MODELING AND ANALYSIS

3.1 Flowchart

1. Power On

Activate the ESP32 microcontroller.

2. Configure ESP32 as an Access Point

Set the ESP32 to function in Access Point mode. Establish the SSID (network name) and password.

3. Establish Wi-Fi Network

The ESP32 generates its own local Wi-Fi network, which does not require internet access.Example SSID: ESP32_AutoHome

4. Initiate Web Server

A lightweight web server is set up on the ESP32. It is designed to respond to HTTP requests from connected devices.

5. Connect Client Device

A smartphone, tablet, or computer connects to the ESP32's Wi-Fi network. Default IP address: 192.168.4.1

6. Access Web Page

The user opens a web browser and goes to 192.168.4.1. An HTML interface is retrieved from the ESP32.

7. Send Command (e.g., Toggle LED or Relay)

The user clicks a button on the web page (e.g., "Turn ON Light"). The ESP32 receives an HTTP GET or POST request.

8. Process Command

The ESP32 analyzes the request to determine the intended action. It translates this into GPIO control logic.

9. Execute Action

The system activates or deactivates a relay, LED, fan, or other household devices.

10. Provide Feedback to User

The ESP32 sends a response (e.g., “Light Turned ON”).

The web page refreshes to reflect the change.

11. Continue or Disconnect

The user has the option to send additional commands or disconnect from the Access Point.

The ESP32 remains ready to receive new inputs.

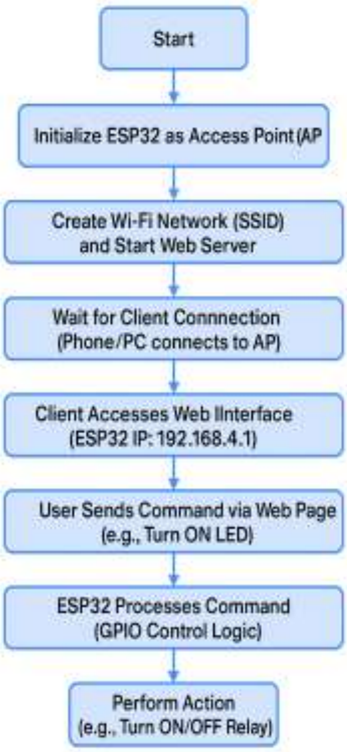

Figure 1: Flow-chart.

Vol. 05, Issue 04, April 2025, pp : 526-531
e-ISSN : 2583-1062
Impact Factor : 7.001

3.2 Block Diagram

The home automation system utilizing the ESP32 in Access Point (AP) mode enables users to manage devices such as lights and fans independently of a router or internet connection. In this configuration, the ESP32 establishes its own Wi-Fi network, allowing users to connect through a smartphone or tablet. After establishing a connection, users can navigate to a web-based control interface hosted on the ESP32 to operate appliances using buttons. The ESP32 interprets these commands and activates relays linked to the devices. Additionally, optional sensors, such as temperature or motion detectors, can be incorporated for automated control. This system provides a cost-effective, low-latency, and secure solution for smart home automation, particularly in locations lacking internet connectivity.

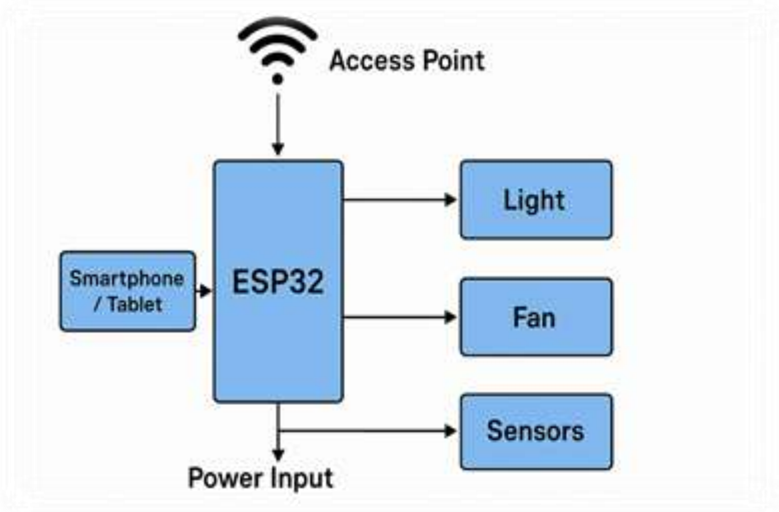

Figure 2: Block diagram.

4. ADVANTAGES DISADVANTAGES AND APPLICATIONS

4.1 Advantages

1. Independent Functionality Without Internet Access

A key benefit of utilizing the ESP32 for local communication is its capability to function independently of a central router or internet connection. Protocols like ESP-NOW, Wi-Fi Direct, and Bluetooth facilitate direct peer-to-peer connections between devices, allowing for data exchange and command execution without reliance on external infrastructure. This minimizes dependence on third-party services and boosts system independence, making it particularly suitable for remote settings, off-grid applications, or critical systems that must remain operational during network failures.

2. Decreased Latency

Local communication systems greatly minimize data transmission delays by removing the necessity to send information through external servers or cloud services. In ESP32 networks, devices communicate directly with one another within the same network or mesh. This direct exchange leads to reduced latency, which is vital for real-time applications like home automation, robotics, and industrial monitoring. The enhanced responsiveness of the system allows for quicker sensor feedback and control actions, which is crucial for applications that require timely responses.

3. Improved Security

Local communication significantly minimizes the risk of sensitive data being compromised by external threats, as it confines data transmission to a secure, closed network. In contrast to cloud-based systems that depend on internet connectivity, locally managed ESP32 networks ensure that all data remains within the physical confines of the system. Additionally, the ESP32 microcontroller is equipped with sophisticated security features, including AES encryption, secure boot, and flash encryption, which offer an extra layer of defense against unauthorized access and manipulation. This makes it an excellent option for applications where the confidentiality and integrity of data are of utmost importance.

4. Energy Efficiency

The ESP32 is engineered for low power consumption, making it ideal for applications that prioritize energy efficiency, such as battery-operated devices, wearables, and environmental monitoring sensors. In local communication modes like ESP-NOW or Bluetooth Low Energy (BLE), the device utilizes considerably less power compared to when it is linked to a complete Wi-Fi network or cloud services. Additionally, the ESP32 features multiple sleep modes, enabling it to function effectively for prolonged durations, even in settings with limited energy resources.

e-ISSN : 2583-1062
Impact Factor : 7.001
Vol. 05, Issue 04, April 2025, pp : 526-531

5. Functionality in Isolated Locations

The capacity to function without reliance on the internet or centralized systems renders ESP32-based solutions particularly ideal for use in remote or rural environments. In areas where network connectivity is sparse or nonexistent, ESP32 devices can continue to execute essential functions such as data collection, sensor observation, and automated management. This capability paves the way for advancements in sectors such as precision agriculture, wildlife tracking, disaster response, and off-grid solutions, where conventional communication methods are either unfeasible or challenging to sustain.

6. Cost-Effective

Establishing local networks using ESP32 technology proves to be significantly more economical than conventional IoT systems that depend on cloud subscriptions, routers, or cellular modules. The ESP32 microcontroller is both affordable and robust, adept at managing intricate tasks without the necessity for supplementary components. By eliminating the requirement for constant internet connectivity and external services, operational expenses are greatly minimized. Consequently, local ESP32 systems present an attractive option for budget-conscious projects, educational initiatives, and scalable industrial applications.

4.2 Disadvantages

1. Limited Range and Connectivity

The ESP32 is equipped with Wi-Fi and Bluetooth capabilities; however, its communication range is naturally restricted. Wi-Fi generally achieves a range of approximately 30 to 50 meters indoors and 100 to 150 meters outdoors, whereas Bluetooth has an even more limited range. The presence of physical obstacles like walls, metal items, and electronic interference can further weaken the signal. In situations where devices are spread across a vast area, ensuring consistent communication is difficult without employing repeaters, mesh networks, or supplementary hardware. Consequently, ESP32-based systems may not be ideal for extensive deployments without considerable adjustments.

2. Lack of Internet Access

While functioning without internet access can improve privacy and lower expenses, it also restricts capabilities. The absence of internet connectivity renders features like remote monitoring, cloud-based data analytics, Over-the-Air (OTA) firmware updates, and third-party integrations (such as with Google Assistant or Alexa) unavailable. This limitation can hinder the overall scalability, manageability, and integration possibilities of the system, particularly for applications that depend on real-time remote interaction or centralized data management.

3. Limited Connection Capacity

The ESP32 has a constrained capacity for managing multiple concurrent connections. When operating as a Wi-Fi Access Point, it generally accommodates between 4 to 10 devices effectively, contingent upon the complexity of the tasks and the demands on bandwidth. Exceeding this limit may lead to network congestion, connection drops, or delays in data transmission. Consequently, the ESP32 is not the most suitable choice for applications that necessitate high-density networking, such as smart buildings, sensor networks, or industrial IoT systems with numerous nodes.

4. Performance and Processing Constraints

Although the ESP32 is a robust microcontroller for its cost, it possesses limited processing capabilities, memory (usually 520 KB SRAM), and storage when compared to more sophisticated microprocessors or embedded systems. It struggles to efficiently perform complex operations like image processing, machine learning, or extensive data analysis without the aid of external modules or cloud services. Consequently, the ESP32 is primarily suited for simpler, low-complexity tasks unless paired with additional hardware.

5. Security Risks

While the ESP32 offers strong security capabilities, numerous real-world applications fail to leverage these features completely, often due to insufficient knowledge or attention. Inadequately configured firmware, unencrypted data transfers, and inadequate authentication measures can make systems susceptible to local threats, including spoofing, man-in-the-middle (MITM) attacks, and unauthorized control of devices. Additionally, the absence of internet connectivity complicates remote security monitoring and patch management, such as the implementation of firmware updates, which may gradually increase the vulnerability of systems.

6. Power Consumption

The ESP32 offers several low-power modes; however, its power consumption during active Wi-Fi communication is significantly greater than that of ultra-low-power options such as LoRa or Zigbee. Continuous data transmission, regular wake-ups, and elevated CPU activity can rapidly deplete batteries, particularly in scenarios where energy efficiency is paramount. In these instances, it may be necessary to incorporate supplementary hardware for energy harvesting, power optimization, or external sleep controllers, which can complicate the design and raise costs.

4.3 Applications

1. Intelligent Lighting Management:

Remotely control LED lights or relays Implement dimmable LED lighting using PWM (Pulse Width Modulation). Automate lighting with timers, motion detectors, or smartphone commands.

2. Intelligent Fan and Climate Regulation

Use temperature sensors (DHT11, DHT22, DS18B20) to adjust fan speed. Enable automatic control based on temperature or humidity.

3. Intelligent Door Lock System

Control door locks via RFID, keypad, or smartphone. Ensure secure access with password authentication.

4. Intelligent Power Outlets

Remotely turn appliances on and off with relays. Monitor energy consumption using current sensors like ACS712.

5. Home Security and Monitoring

Utilize PIR motion sensors for intrusion detection. Stream live video through an ESP32-CAM for surveillance. Add door and window sensors for security alerts.

6. Voice-Activated Automation

Integrate with voice assistants (Google Assistant, Alexa) via MQTT. Employ speech recognition modules for hands-free control.

5. CONCLUSION

Home automation using the ESP32 microcontroller in Access Point (AP) mode offers a flexible and cost-effective way to control household appliances without relying on an external Wi-Fi network. By configuring the ESP32 as an access point, users can connect their smartphones or other devices directly, enabling access to a web interface for managing various home devices like lights and fans.

This setup is ideal for remote or offline locations where Wi-Fi is not available, allowing for easy deployment and mobility. It ensures a user-friendly experience with low latency and direct control. However, the range and number of connectable devices are limited compared to traditional Wi-Fi or cloud-based systems.

In conclusion, the ESP32 in AP mode provides a practical solution for small-scale home automation, allowing users to operate devices independently of internet connectivity while ensuring ease of use.

6. REFERENCES

[1] S. Kumar, " Ubiquitous Smart home System Using Android Application, "International Journal of Computer Networks & Communications, vol. 6, pp. 33-43, January 2014.

[2] R. Piyare,“ Ubiquitous Home Control and Monitoring System using Android based smart phone,” International Journal of Internet of Things, volume. 2,pp. 5-11,2013.

[3] A. Rajabzadeh, A. R. Management, and Z. F. Jahromi, “A Mobile Application for Smart House Remote Control System,” World Academy of Science, Engineering and Technology, vol. 62,2010.

[4] N-S. Liang, L-C. Fu, and C.- L. Wu, “An integrated, flexible, and Internet-based control architecture for home automation system in the Internet era,” in IEEE International Conference on Robotics and Automation, Washington, DC 2002, open. 1101 – 1106.

[5] Mamata Khatu, Neetu Kaimal, Pratik Jadhav, Syedali Adnan Rizvi, “Implementation of Internet of Things for Home Automation”, International Journal of Technology, Volume 3, Issue 2, February 2015.