**Project Report**

**on**

Smart Home Automation using IoT

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING**

IN

**Branch Name**

**Submitted by:**

Kanhaiya Kumar(24BAI70046)

Ansh Uttam(24BAI70321)

Prince Kumar (24BAI70733)

Jatin Malik (24BAI70593)

….

**Under the Guidance of**

**Dr Divneet Singh Kapoor**

**ACADEMIC UNIT-I**

****

**Chandigarh University**

**April 2025**

**Index**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Content** | **Page No.** |
| 1 | Project Overview | 1 |
| 2 | Objective and Problem Statement | 2 |
| 3 | Proposed Solution & Methodology | 3 |
| 4 | Key Findings / Results | 4 |
| 5 | Conclusion & Learnings | 5 |
| 6 | References | 5 |
| 7 | Appendix | 6-9 |

## 1.Project overview

This project focuses on Smart Home Automation using IoT (Internet of Things). The goal is to create an automated system that allows users to control home appliances (such as lights) remotely through the internet using their smartphones.

With increasing reliance on technology, smart homes offer convenience, energy efficiency, and enhanced security. IoT-based automation enables real-time control and monitoring of devices from anywhere, making daily life more efficient.

The project uses basic IoT components to demonstrate a scalable and low-cost smart home solution. Microcontrollers like ESP32 serve as the central hub, connected to the internet via Wi-Fi, allowing communication between the user and household appliances.

Users interact with the system through a dedicated mobile app or a web interface, which sends commands to the microcontroller. These commands are then used to activate or deactivate appliances through relay modules. Feedback from sensors, such as motion detectors or temperature sensors, can also be used to trigger automated responses.

For example, the system can be configured to turn on lights when motion is detected or to switch off devices when no one is in the room, thereby saving electricity. In addition, the system may include voice control integration using virtual assistants like Alexa or Google Assistant, further enhancing user convenience.

Security is also a key aspect of this project. The system can support authentication mechanisms to ensure only authorized users can access or control the devices. Alerts can be sent to the user’s smartphone in case of unusual activity, such as a door opening at odd hours or sudden changes in temperature.

This project serves as a foundational step toward building fully automated homes and can be expanded with features like smart energy metering, scheduling, remote diagnostics, and more. Its low cost and ease of implementation make it ideal for educational purposes as well as real-world applications.

## ****2. Objective and Problem Statement****

**Objective:**  
To develop a smart home automation system that allows users to control household appliances remotely through Wi-Fi using IoT technology. The system should provide a user-friendly interface, ensure secure communication, and support automation features like scheduling and sensor-based triggers. It aims to offer real-time control, status monitoring, and energy efficiency by minimizing unnecessary power usage. The solution should be scalable, low-cost, and easy to implement, making it suitable for both urban and rural households.

The project will also explore integration with smartphones and cloud platforms, enabling seamless connectivity and remote access from anywhere. This objective supports the broader vision of creating intelligent living spaces that enhance the quality of life through automation and technology.

**Problem Statement:**

In conventional homes, manual operation of appliances can be inefficient, time-consuming, and inconvenient, especially for people with mobility issues or during long hours away from home. Forgetting to turn off appliances leads to unnecessary power consumption, increasing electricity bills and contributing to energy waste.

There is a growing demand for smart solutions that can be easily installed and managed without requiring major modifications to existing infrastructure. Most available systems in the market are either too expensive or overly complex for average users.

Therefore, there's a need for a cost-effective, reliable, and user-friendly smart home automation system that enables users to monitor and control their appliances remotely, automate routine tasks, and optimize energy usage. The solution should also provide feedback mechanisms and notifications, improving overall awareness and home management. Addressing this need can contribute to smarter living environments and a more sustainable future.

## ****3. Proposed Solution & Methodology****

### Components Used:

NodeMCU ESP32 Wi-Fi Module

4-Channel Relay Module

4 Bulbs (AC Load)

Male-to-Female Jumper Wires

Power Supply

Ubidots IoT Platform

### Methodology:

1. **System Setup:**

Configure NodeMCU to connect to home Wi-Fi and Ubidots platform.

Relays connected to GPIO pins on the NodeMCU, controlling each bulb.

1. **Cloud Integration (Ubidots):**

Ubidots dashboard used to create switches for each device.

NodeMCU sends and receives data from Ubidots via HTTP/MQTT.

1. **Working Process:**

User logs into Ubidots app/web and toggles switches.

NodeMCU fetches the updated status and controls the relays accordingly.

## ****4. Key Findings / Results****

Successfully implemented IoT-based control using Ubidots.

All bulbs could be switched ON/OFF independently from the Ubidots dashboard.

Real-time device control and monitoring was achieved.

Reliable performance with low latency over Wi-Fi.

**



## ****5. Conclusion & Learnings****

### Learnings:

Hands-on experience with IoT protocols and cloud services.

Practical understanding of NodeMCU ESP32 integration.

Use of Ubidots dashboard to create smart user interfaces.

Real-time data handling between devices and cloud.

### Future Enhancements:

Include sensors (e.g., motion, temperature) for automation.

Add features like scheduling or voice control.

Expand the system to control fans, locks, or alarms.

## ****6. References****

https://ubidots.com/docs

<https://randomnerdtutorials.com>

ESP32 Documentation – <https://www.espressif.com>

<https://tinyurl.com/dt2-2024>

## ****7. Appendix****

### Sample Code Snippet (Using HTTP to Control Relays via Ubidots):

*/\**

*\* Board: DOIT ESP32 DEVKIT v1*

*\* Description: 4 Bulb (Relay) Smart Home Automation with Ubidots*

*\*/*

*#include <WiFi.h>*

*#include <UbidotsESPMQTT.h>*

*#define RELAY1 16  // Use proper GPIO pins*

*#define RELAY2 17*

*#define RELAY3 18*

*#define RELAY4 19*

*#define TOKEN "BBUS-j4Ij5ErnqfoCx8yMRZsvpuhCTMMqyp"  // Your Ubidots TOKEN*

*#define WIFISSID "vivoT35G"                          // Your SSID*

*#define WIFIPASS "12345678"                          // Your Wifi Password*

*Ubidots client(TOKEN);*

*// Callback to handle incoming messages*

*void callback(char\* topic, byte\* payload, unsigned int length) {*

*Serial.print("Message arrived [");*

*Serial.print(topic);*

*Serial.print("] ");*

*String message = topic;*

*if (message.substring(25) == "red/lv") {*

*Serial.println("Red Bulb Command");*

*Serial.println("text:");*

*for (int i = 0; i < length; i++) {*

*Serial.print((char)payload[i]);*

*}*

*Serial.println();*

*Serial.print("Command: ");*

*bool command = \*payload - 48;*

*Serial.println(command);*

*digitalWrite(RELAY1, command);*

*}*

*if (message.substring(25) == "yellow/lv") {*

*Serial.println("Yellow Bulb Command");*

*Serial.println("text:");*

*for (int i = 0; i < length; i++) {*

*Serial.print((char)payload[i]);*

*}*

*Serial.println();*

*Serial.print("Command: ");*

*bool command = \*payload - 48;*

*Serial.println(command);*

*digitalWrite(RELAY2, command);*

*}*

*if (message.substring(25) == "blue/lv") {*

*Serial.println("blue Bulb Command");*

*Serial.println("text:");*

*for (int i = 0; i < length; i++) {*

*Serial.print((char)payload[i]);*

*}*

*Serial.println();*

*Serial.print("Command: ");*

*bool command = \*payload - 48;*

*Serial.println(command);*

*digitalWrite(RELAY3, command);*

*}*

*if (message.substring(25) == "white/lv") {*

*Serial.println("white Bulb Command");*

*Serial.println("text:");*

*for (int i = 0; i < length; i++) {*

*Serial.print((char)payload[i]);*

*}*

*Serial.println();*

*Serial.print("Command: ");*

*bool command = \*payload - 48;*

*Serial.println(command);*

*digitalWrite(RELAY4, command);*

*}*

*}*

*void setup() {*

*Serial.begin(9600);*

*Serial.println("Init... Smart Home Automation");*

*// Set relay pins as output*

*pinMode(RELAY1, OUTPUT);*

*pinMode(RELAY2, OUTPUT);*

*pinMode(RELAY3, OUTPUT);*

*pinMode(RELAY4, OUTPUT);*

*// Initialize all relays as OFF*

*digitalWrite(RELAY1, LOW);*

*digitalWrite(RELAY2, LOW);*

*digitalWrite(RELAY3, LOW);*

*digitalWrite(RELAY4, LOW);*

*// Connect to Wi-Fi*

*Serial.print("Connecting to SSID: ");*

*Serial.println(WIFISSID);*

*client.wifiConnection(WIFISSID, WIFIPASS);*

*// Connect to Ubidots*

*client.ubidotsSetBroker("industrial.api.ubidots.com");*

*client.setDebug(true);*

*client.begin(callback);*

*// Subscribe to each variable*

*client.ubidotsSubscribe("smart-home", "blue");*

*client.ubidotsSubscribe("smart-home", "red");*

*client.ubidotsSubscribe("smart-home", "yellow");*

*client.ubidotsSubscribe("smart-home", "white");*

*Serial.println("Setup done");*

*}*

*void loop() {*

*if (!client.connected()) {*

*client.reconnect();*

*client.ubidotsSubscribe("smart-home", "blue");*

*client.ubidotsSubscribe("smart-home", "red");*

*client.ubidotsSubscribe("smart-home", "yellow");*

*client.ubidotsSubscribe("smart-home", "white");*

*}*

*client.loop();*

*delay(1000);*

*}*