IoT-Based Smart Home Automation System

Final Year Project Report

Name: Musfirah Noor

Department: BS Information Technology

Project Title: Smart Home Automation System (IoT-Based Smart Control)

Session: 2018-2022

1. Abstract

With the rapid advancement of Internet of Things (IoT) technologies, smart home automation has become a major focus in modern engineering systems. This project presents the design and development of a cost-effective, Wi-Fi-enabled Smart Home Automation System using NodeMCU (ESP8266), DHT11/DHT22 sensors, PIR motion sensor, and relay modules.

The system enables real-time monitoring of temperature, humidity, and motion detection while allowing remote control of appliances via the Blynk mobile application. Communication is established using MQTT and HTTP protocols.

The prototype demonstrates efficient automation, low power consumption, and scalable IoT architecture. The project contributes toward energy efficiency, home security, and intelligent infrastructure development.

2. Introduction

Modern homes are increasingly shifting toward intelligent and automated environments due to rapid advancements in Internet of Things (IoT) technologies. Traditional manual control of household appliances often leads to energy wastage, lack of remote accessibility, and limited security monitoring. Users must be physically present to operate devices, which reduces convenience and efficiency. Moreover, conventional systems do not provide real-time environmental monitoring such as temperature tracking or motion detection, which are essential for modern smart living and enhanced home security.

To overcome these limitations, this project proposes an IoT-based Smart Home Automation System that integrates embedded systems, wireless communication, and mobile application control. Using NodeMCU (ESP8266), environmental sensors, relay modules, and Wi-Fi communication via MQTT and HTTP protocols, the system enables real-time monitoring and remote control of appliances. The proposed solution aims to create a cost-effective, scalable, and energy-efficient automation framework that enhances convenience, security, and intelligent infrastructure development.

2.1 Problem Statement

Traditional home environments rely heavily on manual control systems that lack automation, remote accessibility, and intelligent monitoring capabilities. Such systems often result in unnecessary energy consumption, limited security monitoring, and reduced user convenience. Moreover, conventional setups do not provide real-time environmental data such as temperature, humidity, or motion detection, making it difficult to ensure energy optimization and safety within residential spaces.

3. Objectives

1. To design a low-cost IoT-based smart home system.

2. To enable real-time temperature and humidity monitoring.

3. To implement motion detection for security purposes.

4. To allow remote control of appliances using a mobile dashboard.

5. To integrate MQTT and HTTP communication protocols.

4. Literature Review

IoT-based smart home systems have gained attention for improving convenience, security, and energy efficiency. Previous studies show that NodeMCU (ESP8266) is widely used for low-cost, Wi-Fi-enabled automation [3]. Sensors like DHT11/DHT22 provide temperature and humidity data, while PIR sensors detect motion for security [4]. MQTT and HTTP protocols allow real-time monitoring and remote appliance control [2]. Mobile platforms like Blynk make it easy to manage devices from anywhere [5].

Traditional systems lack automation, remote control, and real-time monitoring. IoT-based solutions, like the one proposed in this project, provide a cost-effective, scalable approach for automated home control, energy optimization, and enhanced security.

5. Methodology / Design Approach

The development process included:

- Literature review on IoT-based automation systems
- Design of system architecture and block diagram
- Selection of hardware components (NodeMCU ESP8266, DHT11/DHT22 sensor, PIR motion sensor, Relay module)
- Circuit design and GPIO mapping
- Embedded C/C++ programming
- Development of Python automation scripts
- Integration with Blynk mobile dashboard
- Testing and validation of system performance

6. System Architecture

The architecture consists of four layers:

Sensing Layer: DHT sensor (Temperature & Humidity), PIR sensor (Motion detection)
Processing Layer: NodeMCU ESP8266 microcontroller with embedded C/C++ firmware
Communication Layer: Wi-Fi module using MQTT and HTTP protocols
Application Layer: Blynk mobile dashboard for real-time monitoring interface

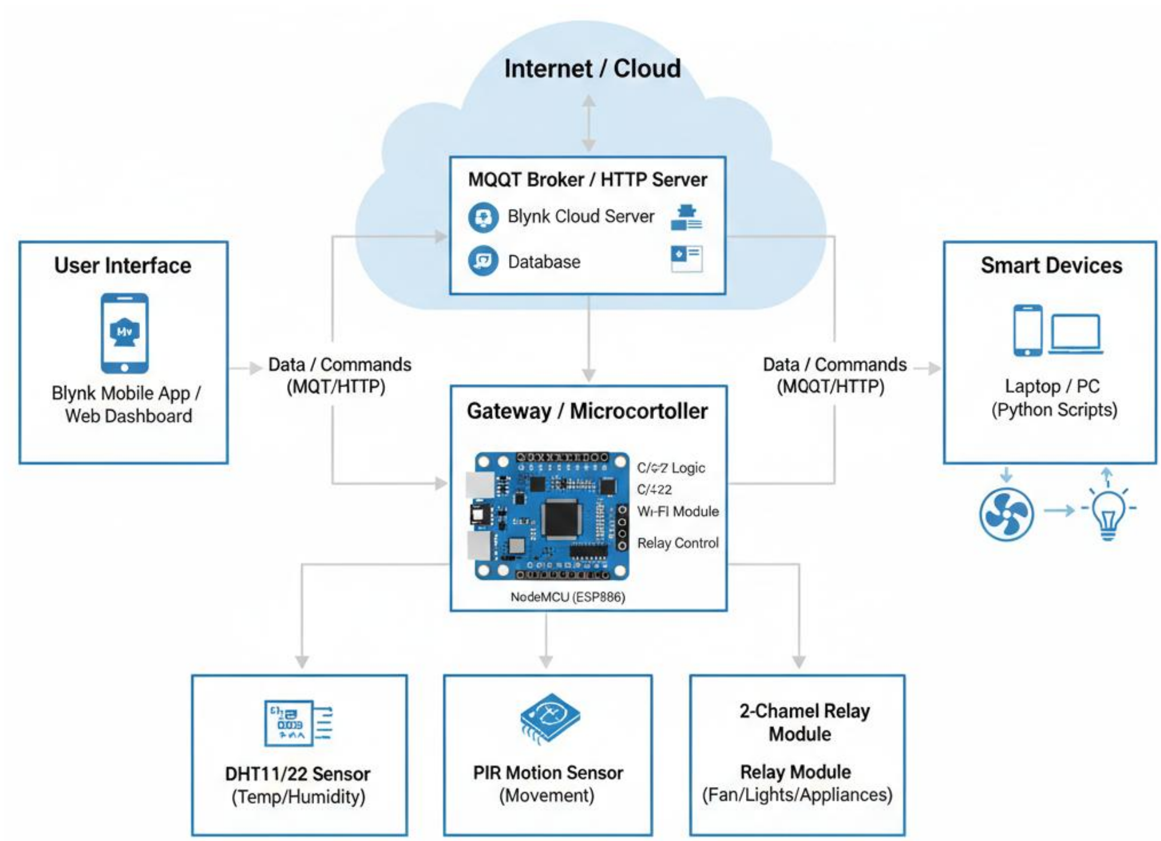

Figure 6.1: System Architecture Diagram

7. Hardware Components

NodeMCU (ESP8266): Built-in Wi-Fi capability, low power consumption, GPIO digital pins.
DHT11 / DHT22: Digital temperature & humidity sensor for environmental monitoring.
PIR Sensor: Detects human motion for security enhancement.
Relay Module: Controls high-voltage appliances with electrical isolation.

Additional Components

Power Supply: 5V regulated supply for stable operation of NodeMCU and sensors.

Jumper Wires & Breadboard: For easy prototyping and connections.

LED Indicators: Optional for visual status of device operation.

7.1 NodeMCU (ESP8266)

Built-in Wi-Fi capability, low power consumption, GPIO digital pins.

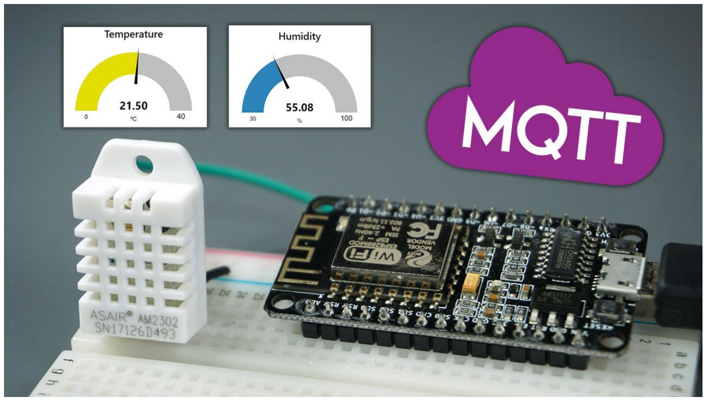

Figure 7.1: NodeMCU (ESP8266) Module

7.2 DHT11 / DHT22 Sensor

Digital temperature & humidity sensor for environmental monitoring.

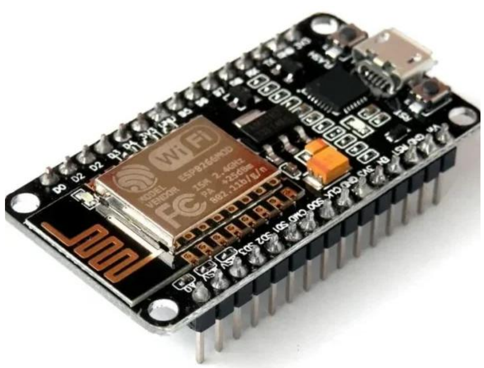

Figure 7.2: DHT11 / DHT22 Sensor

7.3 PIR Motion Sensor

7.4 Detects human motion for security enhancement.

Figure 7.3: PIR Motion Sensor

7.5 Relay Module

Controls high-voltage appliances with electrical isolation.

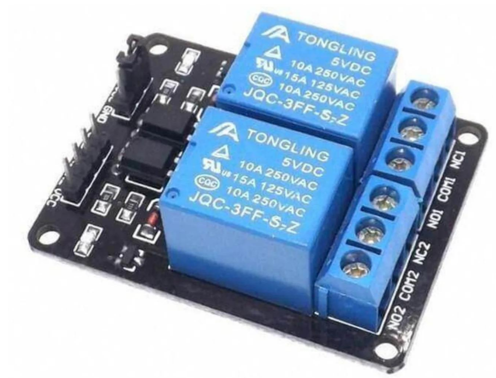

Figure 7.4: Relay Module

8. Circuit Design

The circuit integrates sensors and relay modules with NodeMCU GPIO pins.
DHT sensor connected to digital input pin.
PIR sensor connected to interrupt pin.
Relay module connected to digital output pin.
5V regulated power supply.

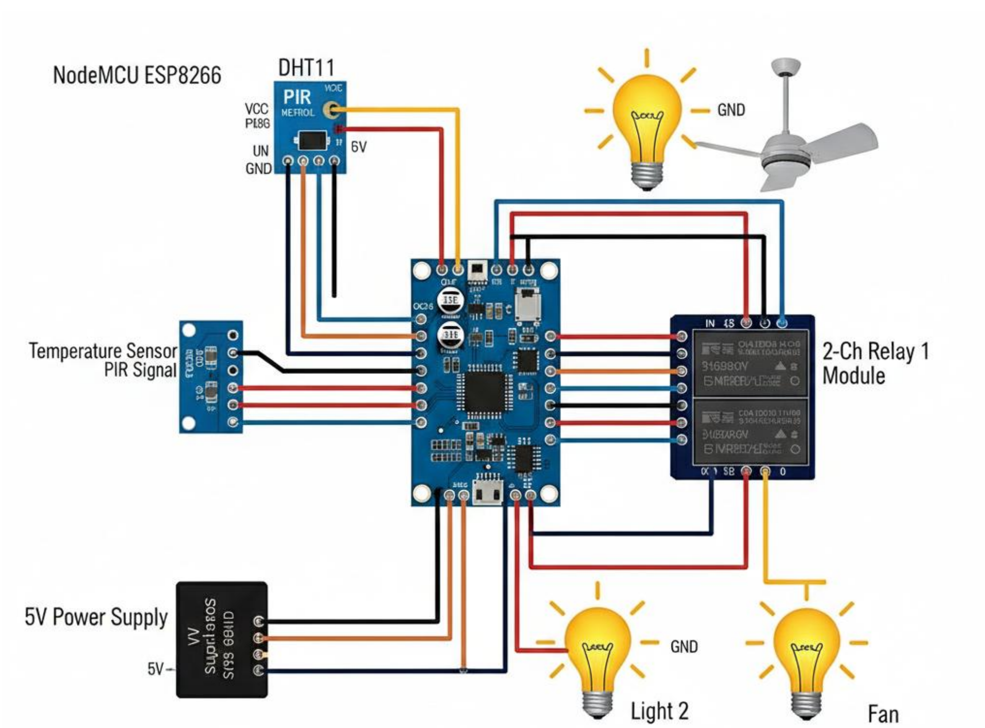

Figure 8.1: Complete Circuit Diagram of the Proposed System

9. Software Implementation

9.1 Embedded Programming (C/C++)

The NodeMCU (ESP8266) was programmed using C/C++ to interface with sensors and relay modules. The firmware handles Wi-Fi connection, reads temperature, humidity, and motion data, and controls appliances automatically. MQTT protocol is used to publish sensor data in real-time.

Wi-Fi initialization, sensor data acquisition, automation logic, MQTT publishing.

```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>


#define DHTPIN D4

#define DHTTYPE DHT11

#define RELAY_PIN D1


const char* ssid = "Your_WiFi_Name";

const char* password = "Your_WiFi_Password";

const char* mqtt_server = "broker.hivemq.com";


WiFiClient espClient;

PubSubClient client(espClient);

DHT dht(DHTPIN, DHTTYPE);


void setup() {

  Serial.begin(9600);

  pinMode(RELAY_PIN, OUTPUT);

  dht.begin();


  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);
```

```
 }

 client.setServer(mqtt_server, 1883);
}

void loop() {
 float temperature = dht.readTemperature();
 float humidity = dht.readHumidity();

 if (temperature > 30) {
   digitalWrite(RELAY_PIN, HIGH);
 } else {
   digitalWrite(RELAY_PIN, LOW);
 }
 client.publish("home/temperature", String(temperature).c_str());
 delay(2000);
}
```

9.2 Python Automation Script

Python scripts were developed to subscribe to MQTT topics, monitor sensor data, and log it into files for analysis. This allows real-time monitoring, data storage, and further automation possibilities.

```
import paho.mqtt.client as mqtt
from datetime import datetime
broker = "broker.hivemq.com"
topic = "home/temperature"

def on_connect(client, userdata, flags, rc):
   print("Connected to MQTT Broker")
   client.subscribe(topic)
def on_message(client, userdata, msg):
```

```
    temperature = msg.payload.decode()

    print("Received Temperature:", temperature)


    with open("data_log.txt", "a") as file:

        file.write(f"{datetime.now()} - Temp: {temperature}\n")

client = mqtt.Client()

client.on_connect = on_connect

client.on_message = on_message

client.connect(broker, 1883, 60)

client.loop_forever()
```

9.3 Mobile Dashboard

A Blynk mobile dashboard was configured to display sensor readings and provide remote control of home appliances. Users can turn devices ON/OFF, view temperature and humidity, and receive alerts for motion detection, all in real-time.

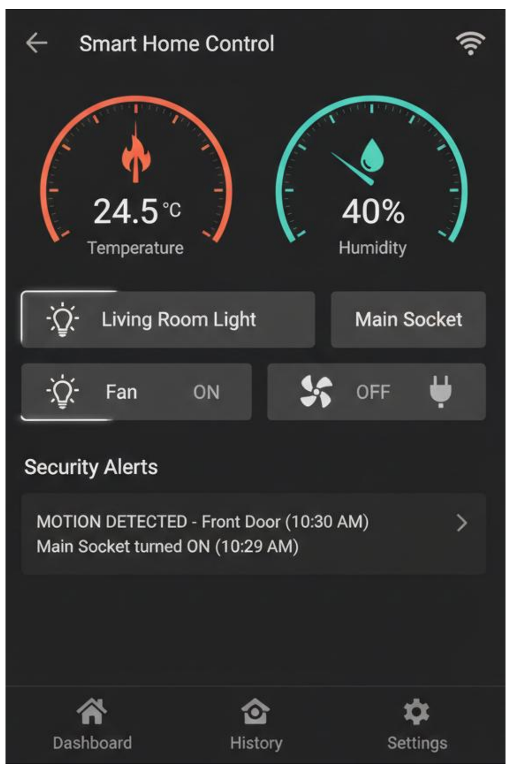

Figure 9.3: Mobile Dashboard

10. Prototype & Results

The developed prototype of the IoT-Based Smart Home Automation System was successfully implemented and tested under real-time conditions. The system demonstrated stable performance in monitoring environmental parameters and controlling household appliances remotely. The NodeMCU (ESP8266) maintained reliable Wi-Fi connectivity, while sensor readings were transmitted efficiently through MQTT protocol. The automation logic responded accurately to predefined temperature thresholds and motion detection events, ensuring both energy efficiency and enhanced security.

Key Features

- Real-time temperature and humidity monitoring using DHT sensor
- Motion detection alerts using PIR sensor
- Remote appliance ON/OFF control via Blynk mobile application
- Stable Wi-Fi communication using MQTT/HTTP
- Low power consumption and cost-effective design

11. Performance Evaluation

The IoT-Based Smart Home Automation System was tested under real-time conditions.

Sensor Accuracy: DHT11/DHT22 sensors provided reliable temperature and humidity readings; PIR sensors detected motion accurately.

Response Time: Appliances responded to threshold events within 1–2 seconds; MQTT ensured low-latency communication.

Remote Control: Blynk dashboard allowed reliable real-time monitoring and control of appliances.

Energy Efficiency: Automated control reduced unnecessary appliance usage.

Stability: Wi-Fi connectivity remained stable, and the system ran continuously without errors.

Parameter	Traditional Manual System	Proposed IoT Prototype
Cost	Moderate to High	Low
Remote Control	Not Available	Available
Energy Efficiency	Low	High
Monitoring	Manual	Real-time

Parameter	Traditional Manual System	Proposed IoT Prototype
Automation	Not Available	Threshold-based

Figure 11.1: Performance Comparison Table

12. Impact

The proposed IoT-Based Smart Home Automation System contributes to improved energy efficiency, enhanced home security, and cost-effective automation. By enabling real-time monitoring and automated control of appliances based on environmental conditions, the system reduces unnecessary energy consumption and minimizes electricity wastage. The integration of motion detection and remote access through a mobile application enhances user convenience and security. Furthermore, its scalable IoT architecture allows future integration with Artificial Intelligence, cloud-based analytics, and smart city infrastructures, making it a sustainable and forward-looking solution for modern intelligent living environments.

13. Future Scope

The proposed IoT-Based Smart Home Automation System provides a strong foundation for future technological enhancements and intelligent system integration. One major extension of this project includes the implementation of Artificial Intelligence-based predictive automation, where machine learning algorithms can analyze user behavior patterns and automatically optimize appliance control. Integration with cloud databases would enable long-term data storage, real-time analytics, and remote system management from multiple devices. Additionally, incorporating energy consumption monitoring modules can help users track electricity usage and improve energy efficiency through data-driven insights.

Furthermore, the system can be expanded by integrating voice assistant technologies such as Google Assistant or Amazon Alexa for hands-free control of appliances. Advanced cybersecurity mechanisms, including encrypted MQTT communication, secure authentication protocols, and firewall-based protection, can also be implemented to enhance data security and system reliability. With these improvements, the system can evolve into a fully intelligent, scalable smart home framework suitable for smart city ecosystems and next-generation IoT infrastructure.

14. Conclusion

The Smart Home Automation System successfully integrates embedded hardware, wireless communication, and mobile application control into a scalable IoT architecture.

This project demonstrates strong technical competence in embedded systems, IoT communication, automation engineering, and intelligent infrastructure development, forming a solid foundation for advanced studies in Artificial Intelligence, Embedded Systems, and Smart Technologies.

References

[1] K. Ashton, 'That Internet of Things Thing,' RFID Journal, 2009.
[2] MQTT Version 3.1.1 Specification, OASIS Standard.
[3] ESP8266 Technical Reference Manual, Espressif Systems.
[4] DHT11 Sensor Datasheet.
[5] Blynk IoT Platform Documentation.