IOT BASED SWITCHBOARD

MICROCONTROLLERS AND INTERFACING
PROJECT REPORT

Submitted by

DHARSHINEE SS	**927623BEC036**
ELAKKIYA K	**927623BEC052**
ELAMATHI S	**927623BEC053**
GOPIKA S	**927623BEC059**

BACHELOR OF ENGINEERING

in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639 113

MAY 2025

M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR

BONAFIDE CERTIFICATE

Certified that this report **"IOT BASED SWITCHBOARD"** is the Bonafide work of **DHARSHINEE.S.S (927623BEC036), ELAKKIYA.K (927623BEC052), ELAMATHI.S (927623BEC053), GOPIKA.S (927623BEC059)** who carried out the project work under my supervision in the academic year 2024 -2025 **EVEN SEMESTER**.

SIGNATURE	**SIGNATURE**
Dr.A.KAVITHA, B.E., M.E., Ph.D.,	**Dr.E.DINESH,B.E., M.E., Ph.D.,**
HEAD OF THE DEPARTMENT,	**SUPERVISOR,**
Professor,	Associate Professor,
Department of Electronics and	Department of Electronics and
Communication Engineering,	Communication Engineering,
M.Kumarasamy College of Engineering,	M.Kumarasamy College of Engineering,
Thalavapalayam,	Thalavapalayam,
Karur-639113.	Karur-639113.

This project report has been submitted for the ECB1223 – Microcontrollers and Interfacing viva voice examination held at M. Kumarasamy College of Engineering, Karur on ___________________________.

INTERNAL EXAMINER

INSTITUTION VISION AND MISSION

Vision

To emerge as a leader among the top institutions in the field of technical education.

Mission

M1: Produce smart technocrats with empirical knowledge who can surmount the global challenges.

M2: Create a diverse, fully -engaged, learner -centric campus environment to provide quality education to the students.

M3: Maintain mutually beneficial partnerships with our alumni, industry and professional associations

DEPARTMENT VISION, MISSION, PEO, PO AND PSO

Vision

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

Mission

M1: Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.

M2: Inculcate the students in problem solving and lifelong learning ability.

M3: Provide entrepreneurial skills and leadership qualities.

M4: Render the technical knowledge and skills of faculty members.

Program Educational Objectives

PEO1: **Core Competence:** Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering

PEO2: **Professionalism:** Graduates will provide feasible solutions for the challenging problems through comprehensive research and innovation in the allied areas of Electronics and Communication Engineering.

PEO3: **Lifelong Learning:** Graduates will contribute to the social needs through lifelong learning, practicing professional ethics and leadership quality

Program Outcomes

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1: Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.

PSO2: Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfil the industrial expectations.

Abstract	Matching with POs, PSOs
IoT (Internet of Things), Smart Switchboard, Remote Appliance Control, NodeMCU (ESP8266), Web-based Interface, Relay Module, Home Automation, Real-time Monitoring.	**PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8,PO9, PO10,PO11,PO12, PSO1,PSO2**

ACKNOWLEDGEMENT

We gratefully remember our beloved **Founder Chairman, (Late) Thiru. M.Kumarasamy**, whose vision and legacy laid the foundation for our education and inspired us to successfully complete this project.

We extend our sincere thanks to **Dr. K. Ramakrishnan, Chairman, and Mr. K. R. Charun Kumar, Joint Secretary**, for providing excellent infrastructure and continuous support throughout our academic journey.

We are privileged to extend our heartfelt thanks to our respected **Principal**, **Dr. B. S. Murugan, B.Tech., M.Tech., Ph.D.,** for providing us with a conducive environment and constant encouragement to pursue this project work.

We sincerely thank **Dr. A. Kavitha, B.E., M.E., Ph.D.,** Professor and **Head, Department of Electronics and Communication Engineering**, for her continuous support, valuable guidance, and motivation throughout the course of this project.

Our special thanks and deep sense of appreciation go to our Project Supervisor, **Dr. E. Dinesh, B.E., M.E., Ph.D.,** Associate Professor, **Department of Electronics and Communication Engineering** , for her exceptional guidance, continuous supervision, constructive suggestions, and unwavering support, all of which have been instrumental in the successful execution of this project.

We would also like to acknowledge **our Class Advisor, Dr. K. Karthikeyan, B.E., M.E., Ph.D.,** Associate Professor, **Department of Electronics and Communication Engineering,** and **our Project Supervisor , Dr.K.Sivanandam, B.E., M.E., Ph.D.,** Associate Professor,, Department of Electronics Engineering (VLSI Design and Technology), for their constant encouragement and coordination that contributed to the smooth progress and completion of our project work.

We gratefully thank all the **faculty members of the Department of Electronics and Communication Engineering** for their timely assistance, valuable insights, and constant support during various phases of the project.

Finally, we extend our profound gratitude to our **parents and friends** for their encouragement, moral support, and motivation, without which the successful completion of this project would not have been possible.

ABSTRACT

The IoT-based Switchboard project aims to create a smart and remotely controllable electrical switchboard system. Traditional switchboards rely on manual operations to control electrical appliances, which can be inefficient and cumbersome. This project leverages Internet of Things (IoT) technology to enable users to control and monitor their home or office appliances via a web application, enhancing convenience, energy efficiency, and automation.

The system consists of a web interface that communicates with embedded IoT hardware connected to electrical devices. Through this interface, users can switch devices on or off, monitor energy consumption, and receive notifications on their device status. The IoT-based switchboard also integrates features like scheduling, where users can set timers for appliances, and remote control, which allows for managing devices from anywhere with an internet connection.

This project uses sensors, microcontrollers, and Wi-Fi modules to connect appliances to the IoT network. The system also ensures security with encryption and authentication mechanisms to protect user data and prevent unauthorized access. The proposed solution is an efficient and scalable system that provides greater control over electrical appliances, improving energy management and overall convenience.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1. Introduction

With the rapid advancement of technology, the concept of smart homes is becoming increasingly popular across the globe. One of the key components of smart home automation is the ability to control electrical appliances such as lights, fans, and sockets remotely, without relying on traditional mechanical switches. The Internet of Things (IoT) has made it possible to create intelligent systems that connect physical devices to the internet, allowing for seamless remote control, automation, and monitoring.

The **IoT-Based Smart Switchboard** project is designed to replace conventional switchboards with a modern, efficient, and connected system. This system allows users to control their electrical appliances using a smartphone, tablet, or computer through Wi-Fi connectivity. This is particularly useful for situations where physical access to a switch is not possible, or for users who want to control appliances while away from home.

At the heart of the system is a microcontroller like the **ESP8266 (NodeMCU)** or **ESP32**, which connects to the internet via Wi-Fi. This microcontroller is programmed to interact with a mobile app (such as **Blynk**, **MIT App Inventor**, or a custom-built app) or a web interface. When a user sends a command from the app, it is transmitted over the internet to the microcontroller, which then activates the appropriate **relay module** to turn a device ON or OFF.

1.2. Problem Statement

Traditional electrical switchboards require manual operation, which limits user convenience and control, especially in scenarios where physical access to the switches is restricted or inconvenient. This reliance on physical interaction becomes a significant drawback in modern lifestyles where automation and remote access are increasingly expected.

There is a lack of remote operability in conventional systems, which poses a challenge for users who wish to control appliances while they are away from home. This limitation is particularly problematic in cases where users forget to switch off appliances, potentially leading to energy waste or safety hazards.

Conventional systems do not support integration with modern smart technologies such as smartphones, Wi-Fi connectivity, or voice assistants. This lack of compatibility hinders the adoption of smart home features and makes traditional switchboards feel outdated in the context of today's interconnected digital ecosystems.

1.3. Objective

The primary objective is to develop an IoT-based smart switchboard that allows users to control electrical appliances remotely through a mobile application or web interface. This remote control feature will enable users to operate devices regardless of their physical location, providing greater flexibility and convenience.

1.4. Description

An IoT-based switchboard is a smart system designed to control and monitor electrical appliances using the Internet of Things (IoT) technology. It allows users to operate home devices like lights, fans, and plugs remotely via a smartphone application, web interface, or voice assistant. The system uses microcontrollers (such as NodeMCU or ESP8266), relay modules, sensors, and Wi-Fi connectivity to communicate with a cloud platform. Unlike traditional switchboards, this system enables automation, real-time status updates, and energy-efficient management of household devices. Its purpose is to increase convenience, ensure safety, and promote energy conservation by integrating smart control features into regular electrical switchboards.

1.5. Methodology Overview

The methodology for developing an IoT-based switchboard begins with a clear analysis of the system requirements, including the number of appliances to be controlled, desired control methods (such as mobile apps or voice commands), and safety considerations. Once the requirements are established, the hardware design phase involves selecting and assembling components like microcontrollers (e.g., NodeMCU), relay modules for switching electrical loads, power supply units, and optional sensors such as temperature or motion detectors. The next phase focuses on software development, where firmware is programmed using platforms like Arduino IDE to manage device operations, sensor input, and network communication. Cloud integration is then implemented using services like Blynk, Firebase, or MQTT to enable remote access, data storage, and real-time updates. A user-friendly mobile application interface is developed or configured to allow easy interaction with the system. Once all components are integrated, the entire setup is rigorously tested for reliability, connectivity, and safety.

1.6 Organisation of report

The report has been organised as follows,

Chapter 2- Describes Literature Survey

Chapter 3- Describes Existing Methodology

Chapter 4- Describes Proposed Methodology

Chapter 5- Describes Result & Discussion

Chapter 6- Ends with Conclusion and Future Work

CHAPTER 2

LITERATURE SURVEY

2.1. IoT Based Smart Home Automation System

Authors: Piyush Maheshwari, Manan Dureja, and Yash Khanduja (2018)

Abstract: This paper presents a smart home automation system that integrates an IoT-based switchboard using the NodeMCU microcontroller. The system uses the Blynk application for smartphone-based control of electrical appliances

2.2. Smart Switch Board for Home Automation

Authors: A. Praveen, S. Deepika, and M. Mahalakshmi (2017):

Abstract: The authors propose a low-cost smart switchboard system based on ESP8266 and relay modules. The system is operated through a local web server hosted on the device, allowing users to control appliances over Wi-Fi.

2.3. IoT Based Smart Switch Using Wi-Fi

Authors: M. R. Rajeswari and K. J. Sangeetha (2019)

Abstract: This paper describes a Wi-Fi enabled smart switch system that can be managed via a mobile application. The system allows real-time control of devices and includes a simple user interface.

2.4. Home Automation System Using IoT and Raspberry Pi

Authors: N. Sriskanthan and F. Tan (2018)

Abstract: In this research, the authors use Raspberry Pi as a central controller to manage a smart switchboard setup. The system supports web-based and voice-controlled commands.

2.5. IoT Based Electrical Device Surveillance System

Authors: K. Sravanthi and D. Rakesh (2020)

Abstract: The paper focuses on monitoring and controlling home appliances through an IoT-enabled switchboard. It incorporates sensors for real-time status feedback and enables users to access device information remotely.

2.6. Android App Based Home Automation Using ESP8266

Authors: H. S. Patel and M. G. Kanabar (2021)

Abstract: This work presents a home automation solution using an ESP8266 microcontroller and an Android application. The smart switchboard can be controlled through Wi-Fi, and users can receive feedback on the status of connected devices.

2.7. Design and Implementation of IoT Based Smart Switchboard with Voice Control

Authors: R. Shankar and A. Anjali (2019)

Abstract: The authors implement voice-controlled functionality into an IoT switchboard using Google Assistant and the IFTTT platform.

2.8. Energy Efficient IoT Based Smart Switchboard System

Authors: T. S. Banupriya and S. P. Kavitha (2020)

Abstract: This paper proposes an energy-efficient switchboard system integrated with IoT to reduce electricity usage. It includes scheduling functions, load monitoring, and automatic switching based on usage patterns.

2.9.IoT Based Automation of Switchboards for Disabled People

Authors:R. Kumari and V. Mehta (2021)

Abstract: Focusing on accessibility, this paper presents a smart switchboard designed for disabled users. The system uses voice commands and gesture recognition for device control.

2.10. Home Automation using Arduino and IoT

Authors: K. Senthil Kumar and R. Nandhini (2017)

Abstract: This study illustrates how Arduino Uno can be employed in creating a low-cost IoT-based smart switchboard. The system is controlled remotely using a web interface and is capable of switching multiple devices.

2.11. SUMMARY

The development of IoT-based switchboards has gained significant attention in recent years, as demonstrated by various research efforts. Many studies have focused on creating low-cost and user-friendly systems that allow remote control of electrical appliances. For instance, Maheshwari et al. (2018) utilized NodeMCU and the Blynk app for smartphone-based switching, while Praveen et al. (2017) proposed a web server–controlled system using ESP8266. Similarly, Rajeswari and Sangeetha (2019) developed a Wi-Fi-based smart switch that supports mobile app integration for easy appliance control. Raspberry Pi was employed by Sriskanthan and Tan (2018) for centralized control and voice-command execution, offering a more advanced approach. Other researchers, such as Sravanthi and Rakesh (2020), incorporated real-time monitoring and energy consumption tracking into their IoT switchboards. Patel and Kanabar (2021) focused on mobile app-based control using ESP8266, adding simplicity to the user interface. Voice control has also been a key feature in some systems, with Shankar and Anjali (2019) integrating Google Assistant and IFTTT for seamless voice operation. Banupriya and Kavitha (2020) highlighted energy efficiency through scheduling and automated switching. Inclusivity was addressed by Kumari and Mehta (2021), who developed a voice- and gesture-based control system tailored for disabled individuals. Lastly, Senthil Kumar and Nandhini (2017) demonstrated the use of Arduino for building a basic yet reliable IoT switchboard with safety features.

CHAPTER 3

EXISTING METHODOLOGY

3.1 INTRODUCTION

Traditional electrical switchboards operate manually, where users physically press switches to control appliances. These systems have the following characteristics:

1. **Manual Operation:**
   All appliances must be turned on or off by physically interacting with the switchboard.
2. **No Remote Control:**
   Users cannot control appliances from a distance or outside their homes/offices.
3. **Lack of Automation:**
   Traditional systems do not offer scheduling, timers, or automatic control features.
4. **No Monitoring Capabilities:**
   There is no way to monitor the status or energy usage of connected appliances.
5. **Higher Energy Wastage:**
   Appliances may remain on unnecessarily due to forgetfulness or lack of awareness, leading to increased energy consumption.
6. **No Smart Integration:**
   Traditional systems cannot be connected to smartphones, internet services, or smart assistants.

3.2. Working principle

The IoT-based Switchboard works on the principle of remote communication between a web application and embedded hardware using Wi-Fi technology. The system enables users to control and monitor electrical appliances over the internet in real time.

1. **User Input via Web Application:**
   The user interacts with a web-based interface to turn appliances ON or OFF, set schedules, or monitor their status.
2. **Data Transmission via Wi-Fi:**
   The commands from the web app are sent to the microcontroller (NodeMCU/ESP8266) via Wi-Fi using HTTP requests or MQTT protocol.
3. **Microcontroller Processing:**
   The NodeMCU receives the data, processes the commands, and activates or deactivates the corresponding relay module.
4. **Relay Control:**
   The relay module acts as an electronic switch that physically controls the connection of appliances to the power supply.
5. **Feedback and Monitoring:**
   The system sends status updates (ON/OFF) back to the web interface, allowing users to view real-time appliance status.
6. **Scheduling (Optional):**
   Timers set through the web application are stored in the microcontroller, and appliances are triggered based on the specified times.

This setup allows for automated, secure, and remote operation of electrical appliances, making the system efficient and user-friendly.

3.3. Limitation

1. **Internet Dependency:**
   The system requires a stable internet connection; without it, remote control and monitoring are not possible.

2. **Limited Range of Local Control:**

   If there's no internet access, users must manually operate switches, as local web hosting is not included.

3. **Basic Security Measures:**

   The current system may have only simple password protection, which might not be strong enough against advanced cyber threats.

4. **No Mobile App Integration:**

   The system uses a web application only; a dedicated mobile app could enhance usability and accessibility.

5. **Limited Appliance Control:**

   The number of appliances that can be controlled depends on the number of relay channels used in the setup.

6. **No Voice Assistant Support:**

   It does not support voice control through smart assistants like Alexa or Google Assistant in its current form.

7. **Energy Monitoring is Basic or Absent:**

   Detailed energy usage analytics and cost estimation features are not implemented.

8. **No Backup System:**

   In case of power failure or hardware malfunction, the system does not have a backup or alert mechanism.

CHAPTER 4

PROPOSED METHOD

4.1. INTRODUCTION

The proposed method aims to modernize traditional switchboards by integrating **IoT technology**, allowing users to control and monitor their home or office electrical appliances **remotely via a web application**.

Key Features of the Proposed System:

1. **Web-Based Control:**
   A responsive web application allows users to switch appliances ON/OFF, monitor their status, and set schedules from any location.
2. **IoT Integration Using NodeMCU (ESP8266):**
   The NodeMCU microcontroller with built-in Wi-Fi connects to the internet and acts as the central controller for appliance switching.
3. **Relay-Based Switching:**
   Electrical appliances are connected to relays, which are controlled by the NodeMCU based on commands received from the web app.
4. **Real-Time Monitoring:**
   The system provides instant feedback on appliance status, ensuring accurate and timely updates for the user.
5. **Scheduling Functionality:**
   Users can set timers to automate appliance operation, improving energy efficiency and convenience.
6. **Secure Access:**
   The system includes authentication features to ensure that only authorized users can control the appliances.

7. **Scalable Design:**

   Additional relays or modules can be added to control more appliances, making the system suitable for both small and large-scale use.

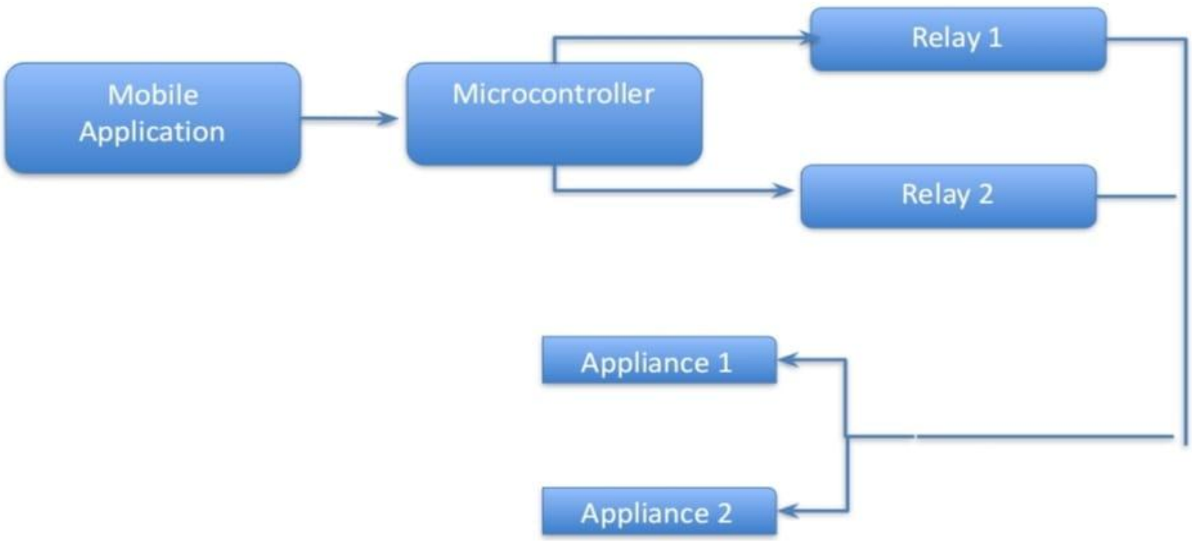

Figure 4.1 Block Diagram

4.2 Explanation

1. **Mobile Application:**
   - The user interacts with this app to send control commands (like ON/OFF) to appliances.
   - It serves as the front-end interface.
2. **Microcontroller:**
   - This is typically a NodeMCU (ESP8266) or similar IoT-enabled board.
   - It receives the commands from the mobile application via Wi-Fi.
   - It acts as the brain of the system, processing commands and forwarding them to relays.

3. **Relay 1 and Relay 2:**
   - These are electronic switches controlled by the microcontroller.
   - Each relay is connected to a different appliance.
   - When the microcontroller sends a signal, the corresponding relay opens or closes, turning the appliance ON or OFF.
4. **Appliance 1 and Appliance 2:**
   - These are the electrical devices (like lights, fans, etc.) that are controlled by the relays.
   - Each appliance is connected to one of the relays, allowing it to be independently controlled.

Flow of Operation:

- The user sends a command from the mobile app.
- The microcontroller receives the command over the internet (via Wi-Fi).
- It activates the appropriate relay (Relay 1 or Relay 2).
- The relay then controls the connected appliance (Appliance 1 or Appliance 2) by completing or breaking the power circuit.

4.3 Advantages

Remote Access:

Users can control electrical appliances from anywhere using an internet-connected device.

Energy Efficiency:

Helps save electricity by allowing users to turn off unused appliances remotely or on a schedule.

User Convenience:

No need for manual operation; appliances can be managed with a simple web interface.

Scheduling Functionality:

Devices can be turned on/off at specific times automatically, reducing human effort.

Real-time Monitoring:

Users can view the current status of each appliance instantly on the web application.

Improved Safety:

Reduces the risk of leaving appliances on accidentally and allows for remote shutdown.

Scalability:

The system can be easily expanded to control more devices or cover larger areas like offices or buildings.

Cost-effective:

Utilizes affordable components like NodeMCU and relays, making it budget-friendly for personal or small business use.

Enhanced Security:

Basic authentication protects the system from unauthorized access.

CHAPTER 5

RESULT AND DISCUSSION

5.1. Tool Description:

1. **NodeMCU (ESP8266):**

   A low-cost microcontroller with built-in Wi-Fi used to control the electrical devices and connect them to the internet. It acts as the core controller of the system, receiving commands from the web application and switching appliances accordingly.

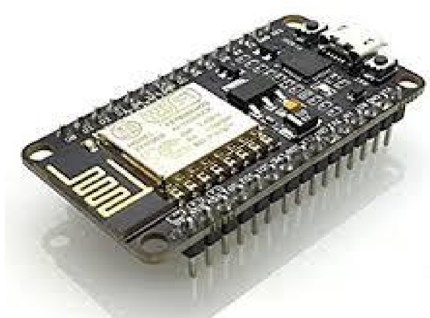

Fig 5.1 ESP8266

2. **Relay Module:**

   A relay module is used to control the high-voltage electrical appliances. It acts as a switch, allowing the NodeMCU to turn devices on or off safely.

3. **Web Application:**

   A user-friendly interface developed to allow users to control and monitor connected devices. It provides features such as on/off control, scheduling, and real-time status updates.

4. **Wi-Fi Router:**

   Provides internet connectivity for the NodeMCU to communicate with the web server and the web application.

5. **Power Supply:**

   Powers the NodeMCU, relay module, and other components in the circuit. Typically includes a 5V adapter or USB connection**.**

6. **Jumper Wires and Breadboard/PCB:**

   Used for making connections between the components during prototyping and testing.

- Stable Connectivity: NodeMCU maintained a stable Wi-Fi connection and responded quickly to user commands.
- Secure Access: The system included basic authentication to restrict unauthorized control.

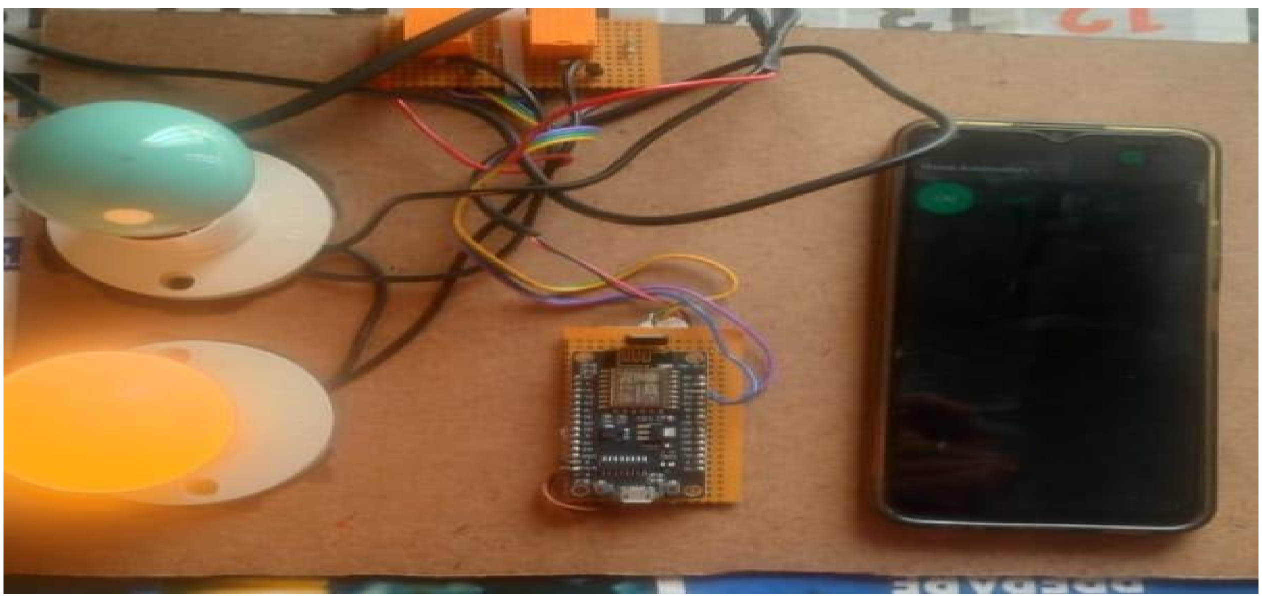

Fig 5.2 Hardware Model

CHAPTER 6

CONCLUSION AND FUTURE WORK

The IoT-based Switchboard project demonstrates a practical and efficient approach to modernizing traditional electrical systems. By integrating Internet of Things (IoT) technology, users can remotely control and monitor their home or office appliances through a user-friendly web application. This system improves convenience, saves energy, and adds a layer of automation to everyday tasks. Features like scheduling, real-time status updates, and secure access enhance the overall functionality and user experience. The project proves to be a reliable and scalable solution for smart home and office automation.

Future Work :

Voice Assistant Integration: The system can be enhanced by integrating voice assistants such as Amazon Alexa or Google Assistant, allowing users to control appliances through simple voice commands.

Mobile Application Development: Creating a dedicated mobile application can improve user experience by providing better accessibility and control from smartphones.

Advanced Energy Analytics: Implementing data analytics tools can help track power usage patterns and offer suggestions for reducing energy consumption, promoting smarter energy management.

System Scalability: The project can be scaled up to manage larger spaces such as office buildings, apartments, or institutions, contributing to smart infrastructure development.

AI-Based Automation: In the future, artificial intelligence can be used to learn user behavior and automate appliance control based on usage habits, time of day, or occupancy.

REFERENCES

1. Sharma, R., & Mishra, S. (2023). IoT-Based Home Automation Using NodeMCU and Blynk Application**.** International Journal of Emerging Technologies and Innovative Research (JETIR), 10(2), 456–462.
2. Patil, A., & Kulkarni, M. (2022). Design and Implementation of Smart Switchboard using IoT and ESP8266. International Journal of Engineering Research & Technology (IJERT), 11(4), 85–89.
3. Khan, M., & Rahman, S. (2021). Home Automation System Using IoT and Cloud-Based Services. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 7(1), 23–30.
4. Kumar, R. et al. (2023). IoT-Based Appliance Control Using NodeMCU and Google Firebase. International Journal of Innovative Science and Research Technology (IJISRT), 8(3), 512–518.
5. Sahu, D., & Pandey, A. (2022). Smart Switchboard for Home Automation Using ESP8266. Proceedings of the 2022 International Conference on Smart Technologies (ICST).
6. NodeMCU Official Documentation (2024). Available at: https://nodemcu.readthedocs.io/
7. Espressif Systems (2023). ESP8266 Datasheet and Technical Reference Manual.Available at:https://www.espressif.com/en/products/socs/esp8266

APPENDICES - (CODING)

APPLICATION:

```
#define BLYNK_TEMPLATE_ID "TMPL3hIoYXhCn"
#define BLYNK_TEMPLATE_NAME "home automation"
#define BLYNK_AUTH_TOKEN "prXQuuNzxsBCtQedRrZD_r-8XMpZGxvI"
//#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
// define the GPIO connected with Relays and switches
#define RelayPin1 5  //D1
#define RelayPin2 4  //D2
#define RelayPin3 14 //D5
#define RelayPin4 12 //D6
#define SwitchPin1 10  //SD3
#define SwitchPin2 0   //D3
#define SwitchPin3 13  //D7
#define SwitchPin4 3   //RX
#define wifiLed   16   //D0
#define VPIN_BUTTON_1    V1
#define VPIN_BUTTON_2    V2
#define VPIN_BUTTON_3    V3
#define VPIN_BUTTON_4    V4
int toggleState_1 = 1; //Define integer to remember the toggle state for relay 1
int toggleState_2 = 1; //Define integer to remember the toggle state for relay 2
int toggleState_3 = 1; //Define integer to remember the toggle state for relay 3
int toggleState_4 = 1; //Define integer to remember the toggle state for relay 4
int wifiFlag = 0;
#define AUTH "prXQuuNzxsBCtQedRrZD_r-8XMpZGxvI"                    // You
should get Auth Token in the Blynk App.
#define WIFI_SSID "iot"            //Enter Wifi Name
#define WIFI_PASS "123456789"        //Enter wifi Password
BlynkTimer timer;
void relayOnOff(int relay){
 switch(relay){
      case 1:
             if(toggleState_1 == 1){
              digitalWrite(RelayPin1, LOW); // turn on relay 1
              toggleState_1 = 0;
              Serial.println("Device1 ON");
              }
```

```
        else{
         digitalWrite(RelayPin1, HIGH); // turn off relay 1
         toggleState_1 = 1;
         Serial.println("Device1 OFF");
         }
        delay(100);
    break;
    case 2:
        if(toggleState_2 == 1){
         digitalWrite(RelayPin2, LOW); // turn on relay 2
         toggleState_2 = 0;
         Serial.println("Device2 ON");
         }
        else{
         digitalWrite(RelayPin2, HIGH); // turn off relay 2
         toggleState_2 = 1;
         Serial.println("Device2 OFF");
         }
        delay(100);
    break;
    case 3:
        if(toggleState_3 == 1){
         digitalWrite(RelayPin3, LOW); // turn on relay 3
         toggleState_3 = 0;
         Serial.println("Device3 ON");
         }
        else{
         digitalWrite(RelayPin3, HIGH); // turn off relay 3
         toggleState_3 = 1;
         Serial.println("Device3 OFF");
         }
        delay(100);
    break;
    case 4:
        if(toggleState_4 == 1){
         digitalWrite(RelayPin4, LOW); // turn on relay 4
         toggleState_4 = 0;
         Serial.println("Device4 ON");
         }
        else{
         digitalWrite(RelayPin4, HIGH); // turn off relay 4
         toggleState_4 = 1;
```

```
            Serial.println("Device4 OFF");
            }
           delay(100);
      break;
      default : break;
      }
  }
void with_internet(){
   //Manual Switch Control
   if (digitalRead(SwitchPin1) == LOW){
     delay(200);
     relayOnOff(1);
     Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);    // Update Button
Widget
   }
   else if (digitalRead(SwitchPin2) == LOW){
     delay(200);
     relayOnOff(2);
     Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);    // Update Button
Widget
   }
   else if (digitalRead(SwitchPin3) == LOW){
     delay(200);
     relayOnOff(3);
     Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);    // Update Button
Widget
   }
   else if (digitalRead(SwitchPin4) == LOW){
     delay(200);
     relayOnOff(4);
     Blynk.virtualWrite(VPIN_BUTTON_4, toggleState_4);    // Update Button
Widget
   }
}
void without_internet(){
   //Manual Switch Control
   if (digitalRead(SwitchPin1) == LOW){
     delay(200);
     relayOnOff(1);
   }
   else if (digitalRead(SwitchPin2) == LOW){
     delay(200);
```

```
      relayOnOff(2);
    }
    else if (digitalRead(SwitchPin3) == LOW){
      delay(200);
      relayOnOff(3);
    }
    else if (digitalRead(SwitchPin4) == LOW){
      delay(200);
      relayOnOff(4);
    }
}
BLYNK_CONNECTED() {
  // Request the latest state from the server
  Blynk.syncVirtual(VPIN_BUTTON_1);
  Blynk.syncVirtual(VPIN_BUTTON_2);
  Blynk.syncVirtual(VPIN_BUTTON_3);
  Blynk.syncVirtual(VPIN_BUTTON_4);
}


// When App button is pushed - switch the state
BLYNK_WRITE(VPIN_BUTTON_1) {
  toggleState_1 = param.asInt();
  digitalWrite(RelayPin1, toggleState_1);
}
BLYNK_WRITE(VPIN_BUTTON_2) {
  toggleState_2 = param.asInt();
  digitalWrite(RelayPin2, toggleState_2);
}
BLYNK_WRITE(VPIN_BUTTON_3) {
  toggleState_3 = param.asInt();
  digitalWrite(RelayPin3, toggleState_3);
}
BLYNK_WRITE(VPIN_BUTTON_4) {
  toggleState_4 = param.asInt();
  digitalWrite(RelayPin4, toggleState_4);
}
void checkBlynkStatus() { // called every 3 seconds by SimpleTimer
  bool isconnected = Blynk.connected();
  if (isconnected == false) {
    wifiFlag = 1;
    digitalWrite(wifiLed, HIGH); //Turn off WiFi LED
```

```
  }
  if (isconnected == true) {
    wifiFlag = 0;
    digitalWrite(wifiLed, LOW); //Turn on WiFi LED
  }
}
void setup()
{
  Serial.begin(9600);
  pinMode(RelayPin1, OUTPUT);
  pinMode(RelayPin2, OUTPUT);
  pinMode(RelayPin3, OUTPUT);
  pinMode(RelayPin4, OUTPUT);
  pinMode(wifiLed, OUTPUT);
  pinMode(SwitchPin1, INPUT_PULLUP);
  pinMode(SwitchPin2, INPUT_PULLUP);
  pinMode(SwitchPin3, INPUT_PULLUP);
  pinMode(SwitchPin4, INPUT_PULLUP);
  //During Starting all Relays should TURN OFF
  digitalWrite(RelayPin1, toggleState_1);
  digitalWrite(RelayPin2, toggleState_2);
  digitalWrite(RelayPin3, toggleState_3);
  digitalWrite(RelayPin4, toggleState_4);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  timer.setInterval(3000L, checkBlynkStatus); // check if Blynk server is connected
every 3 seconds
  Blynk.config(AUTH);
}
void loop()
{
  if (WiFi.status() != WL_CONNECTED)
  {
    Serial.println("WiFi Not Connected");
  }
  else
  {
    Serial.println("WiFi Connected");
    Blynk.run();
  }
 timer.run(); // Initiates SimpleTimer
  if (wifiFlag == 0)
    with_internet();
```

```
  else
    without_internet();
}
```

WIFI

```
#include <ESP8266WiFi.h>

// Replace with your WiFi credentials
const char* ssid = "iot";           // Your Wi-Fi SSID
const char* password = "123456789";  // Your Wi-Fi Password

void setup() {
  Serial.begin(9600); // Start the Serial communication
  delay(100);

  // Begin WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Connected
  Serial.println();
  Serial.println("WiFi connected successfully!");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP()); // Print local IP address
}

void loop() {
  // Nothing to do here for now
}
```