Wireless Home Automation System Using Internet of Things

Archit Bhagowati	210710007008
Hillol Pratim Kalita	210710007016
Rishika Hazarika	210710007037
Shanan Thakuria	210710007045
Saquib Bin Halim	220750007005

May 14, 2025

1 Introduction

This project, titled *Wireless Home Automation System Using Internet of Things*, demonstrates a simple yet effective IoT-based system to control light bulbs wirelessly using the Blynk app. The system leverages an `ESP32S NodeMCU microcontroller`, a `dual-channel relay module`, and the `Blynk` platform to provide a user-friendly interface for home automation.The primary objective of this project is to design and implement a system that allows users to turn on or off two light bulbs remotely via a smartphone app.

2 System Design and Components

The wireless home automation system consists of both hardware and software components working together to achieve remote control of appliances. The core idea is to use an IoT-enabled microcontroller to interface between a user's smartphone and the appliances.

2.1 Hardware Components

- **ESP32S NodeMCU**: A microcontroller with built-in Wi-Fi and Bluetooth capabilities. It serves as the central control unit, connecting to the internet and communicating with the Blynk app. The **ESP32S operates at 3.3V** and is powered via a micro-USB port.

- **Dual-Channel Relay Module (Songle SRD-05VDC-SL-C)**: A 2-channel relay module rated for 10A at 250VAC/125VAC. It acts as a switch to control high-voltage devices (light bulbs) by receiving signals from the ESP32S. The module has two input pins (IN1, IN2) to control each channel, along with VCC and GND for power.

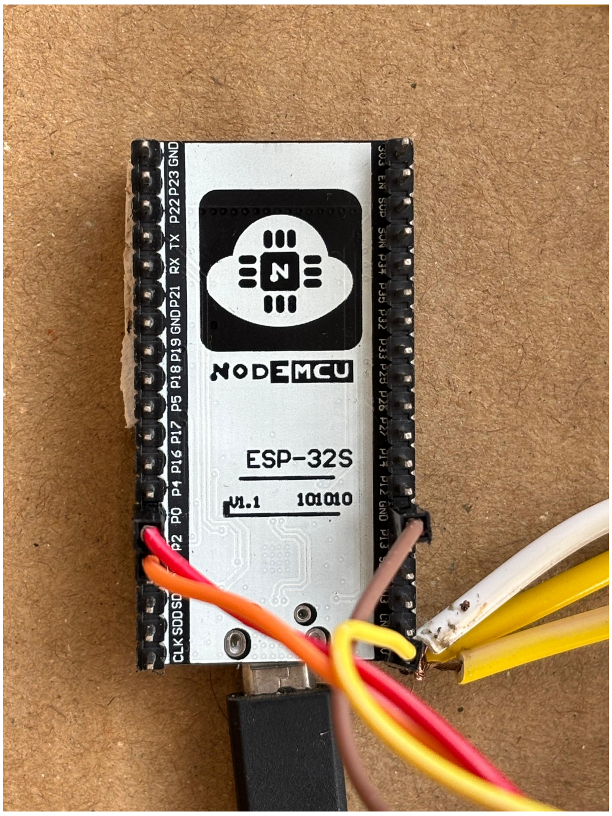
NODEMCU
ESP-32S
V1.1
101010
CLK SD SD P2 P0 P4 P16 P17 P5 P18 P19 GND P21 RX TX P22 P23 GND

Figure 1: ESP32 - WiFi module

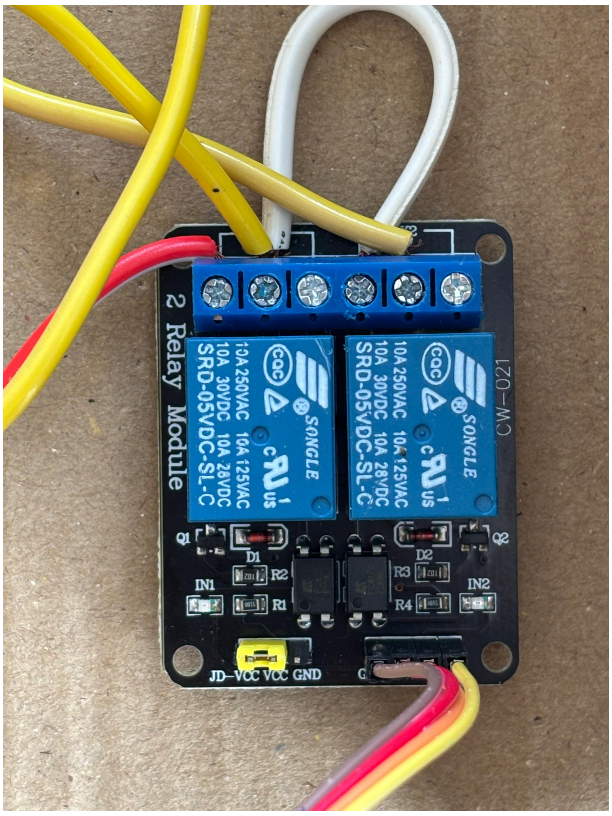

Figure 2: Dual-Channel Relay module

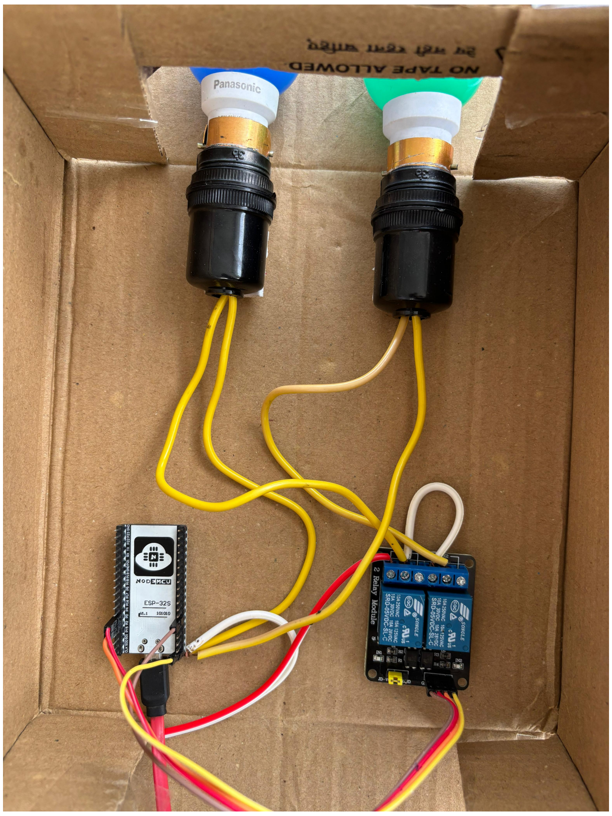

Figure 3: Project setup

- **Light Bulbs and Holders**: Two Panasonic light bulbs mounted in bulb holders serve as the appliances to be controlled. They are connected to the relay module's output terminals (COM and NO).
- **Wires**: Jumper wires (red, yellow, white) are used for connections between the ESP32S, relay module, and bulb holders.
- **Power Supply**: The ESP32S is powered via a USB cable, while the relay module operates at 5V, sourced from the ESP32S or an external power supply.

2.2 Software Components

- **Blynk App**: A mobile application used to create a user interface for controlling the light bulbs. It communicates with the ESP32S over the internet using Wi-Fi.
- **Arduino IDE**: Used to program the ESP32S with the necessary code to interface with Blynk and control the relay module.

2.3 System Architecture

The ESP32S connects to a Wi-Fi network and establishes communication with the Blynk server. The Blynk app, installed on a smartphone, sends commands (e.g., turn on/off) to the ESP32S via the Blynk server. The ESP32S then toggles the relay module's channels by sending high or low signals to the IN1 and IN2 pins, which in turn control the light bulbs.

3 Implementation

The implementation of the wireless home automation system involves both hardware assembly and software configuration.

3.1 Hardware Setup

The hardware setup begins with connecting the ESP32S NodeMCU to the dual-channel relay module. The connections are as follows:

- ESP32S GND to Relay GND.
- ESP32S 3.3V to Relay VCC (or an external 5V source if required).
- ESP32S GPIO pins (e.g., D5 and D4) to Relay IN1 and IN2, respectively.

Next, the light bulbs are connected to the relay module. Each bulb holder is wired to one channel of the relay:

- Bulb 1: Connected to Relay Channel 1 (COM1 and NO1).
- Bulb 2: Connected to Relay Channel 2 (COM2 and NO2).

The entire setup is placed in a cardboard box for demonstration, with the bulb holders mounted on top. The ESP32S is powered via a USB cable connected to a computer or power adapter.

3.2 Software Configuration

The software setup involves programming the ESP32S using the Arduino IDE and configuring the Blynk app.

1. **Blynk App Setup**: A new project is created in the Blynk app, and an authentication token is generated. Two virtual buttons are added to the project interface, one for each light bulb, mapped to virtual pins (e.g., V1 and V2).

2. **ESP32S Programming**: The ESP32S is programmed using the Arduino IDE. The Blynk library is installed, and the following code is uploaded to the ESP32S:

```
#define BLYNK_TEMPLATE_ID "TMPL3OxfpPLoy"
#define BLYNK_TEMPLATE_NAME "IoT"
#define BLYNK_AUTH_TOKEN "ee7OlOzhkkQlxRaDPZpyJ-y-vLFBINiX"
#define RELAY1 26
#define RELAY2 27

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char ssid[] = "Moto";
char pass[] = "halim123";
```

```
void setup() {
  Serial.begin(115200);
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  digitalWrite(RELAY1, HIGH); // OFF initially (active LOW)
  digitalWrite(RELAY2, HIGH);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

BLYNK_WRITE(V0) {
  int value = param.asInt();
  digitalWrite(RELAY1, value ? LOW : HIGH); // Active LOW
}

BLYNK_WRITE(V1) {
  int value = param.asInt();
  digitalWrite(RELAY2, value ? LOW : HIGH);
}

void loop() {
  Blynk.run();
}
```

This code connects the ESP32S to the Wi-Fi network, interfaces with the Blynk server, and controls the relay pins based on commands from the app.

3.3 Testing

Once the hardware and software are set up, the system is tested by launching the Blynk app and toggling the virtual buttons. When a button is pressed, the corresponding relay channel activates, turning the light bulb on or off. The system successfully allows wireless control of the light bulbs from the smartphone.

4 Conclusion

The Wireless Home Automation System Using Internet of Things was successfully implemented using an ESP32S NodeMCU, a dual-channel relay module, and the Blynk app. The system demonstrates the potential of IoT in creating affordable and efficient home automation solutions. Users can control two light bulbs remotely with ease, and the setup is scalable for additional devices.

However, the current prototype has limitations, such as being housed in a temporary cardboard box and lacking safety features for high-voltage handling. Overall, this project serves as a practical introduction to IoT-based home automation and its applications.