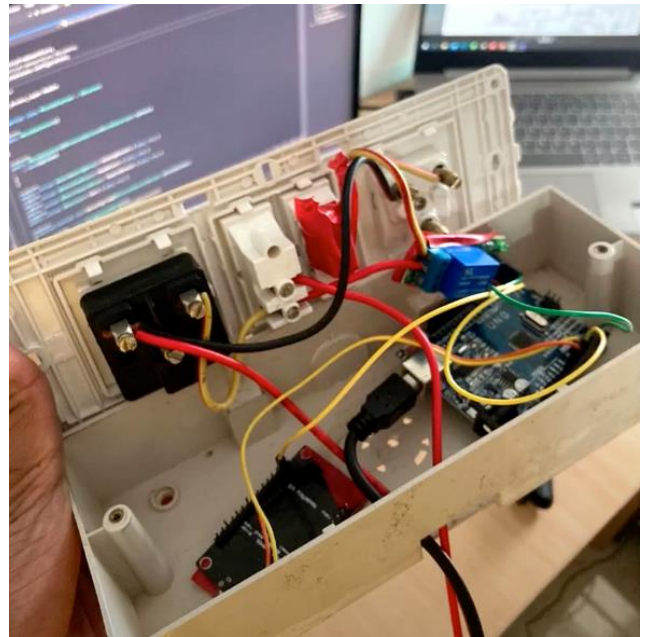


Home Automation using NodeMCU (ESP8266)

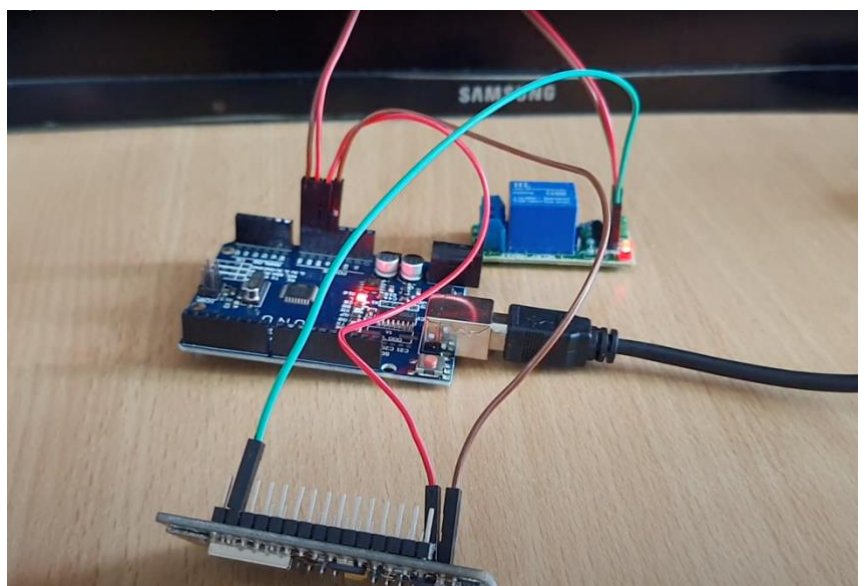
By- [Mayank Patel](#)

Home automation using NodeMCU (ESP8266) is a versatile and cost-effective way to control and monitor devices in a smart home. The NodeMCU is a development board equipped with an ESP8266 microcontroller, which supports Wi-Fi connectivity. This allows it to interact with IoT platforms and mobile applications for seamless integration and control.



In this setup, the NodeMCU is the system's brain, connecting to home appliances like lights, fans, and security systems through relays or other actuators. The relays act as switches that can turn devices on or off. The ESP8266's Wi-Fi capabilities enable it to receive commands via apps like Blynk, MQTT-based platforms, or custom-built dashboards. Additionally, it can send real-time feedback to users about the state of connected devices.

Sensors, such as motion detectors, temperature, or light sensors, can also be integrated, enhancing automation by triggering specific actions based on environmental changes. For instance, lights can automatically turn on when motion is detected or adjust brightness based on ambient light.

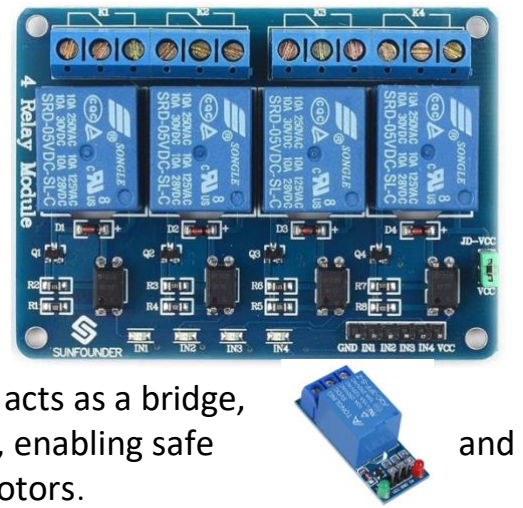


This system's affordability, reliability, and scalability make it ideal for creating customized home automation solutions, empowering users to enhance comfort, security, and energy efficiency in their homes.

Material Required

1. Relay Board

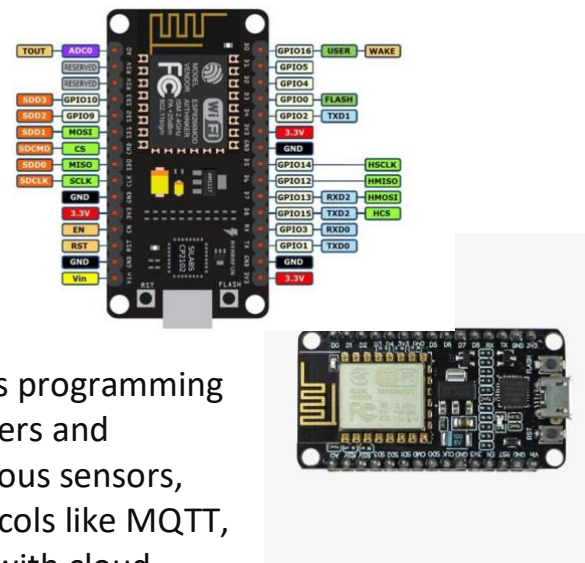
A relay board is an electronic module that uses relays to control high-voltage or high-current devices through low-voltage signals, such as those from microcontrollers like Arduino or NodeMCU. It acts as a bridge, isolating low-power circuits from high-power loads, enabling safe efficient control of appliances like lights, fans, or motors.



2. Node MCU

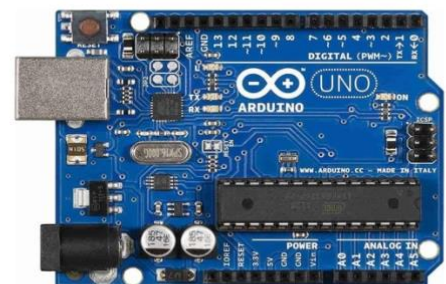
The NodeMCU ESP8266 is a low-cost, open-source IoT development board featuring the ESP8266 Wi-Fi module.

It combines a powerful 32-bit microcontroller with built-in Wi-Fi capabilities, making it ideal for IoT and smart home projects. The board supports programming in Lua or Arduino IDE, providing flexibility for beginners and experts. With its GPIO pins, it can interface with various sensors, relays, and actuators. The NodeMCU supports protocols like MQTT, HTTP, and WebSocket for seamless communication with cloud platforms and mobile apps. Its compact design, affordability, and versatility make it popular for building connected devices and prototyping innovative IoT solutions.



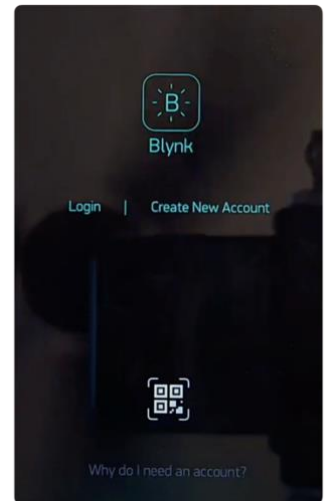
3. Arduino Uno

(Arduino is only used for power. other alternatives Can also be used.)



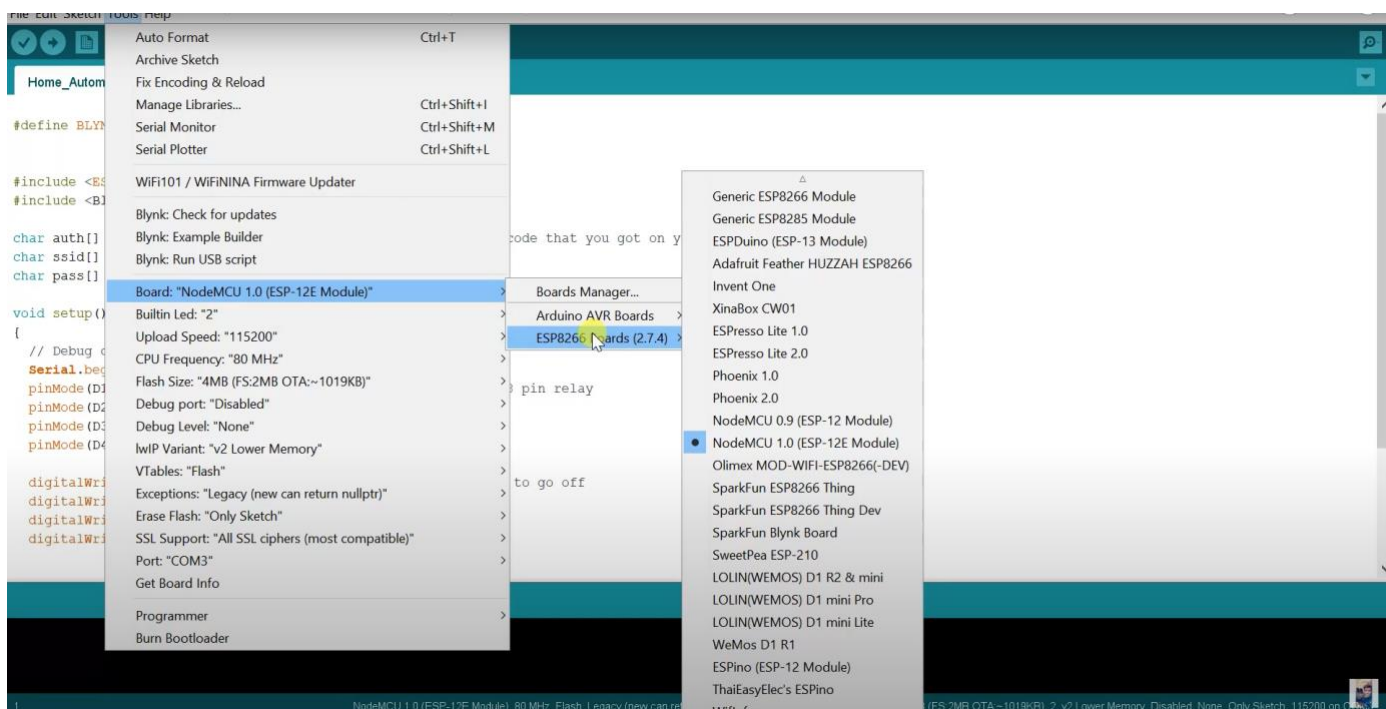
4. Blynk

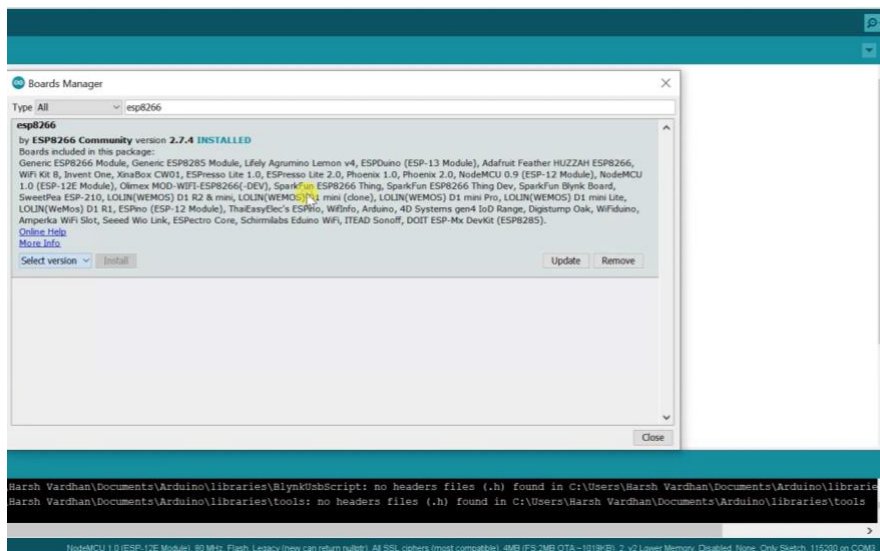
Blynk is a user-friendly IoT platform that allows developers to build and control smart devices through a mobile app. It supports a wide range of hardware, including Arduino, NodeMCU, ESP8266, and Raspberry Pi, enabling seamless integration with sensors and actuators. The platform features a drag-and-drop app builder, allowing users to create custom dashboards for real-time monitoring and control of connected devices. Blynk supports cloud, local server options, and communication protocols like Wi-Fi, Bluetooth, and GSM. Its features include virtual pins, automation, and notifications, making it ideal for IoT projects such as home automation, smart gardening, or industrial monitoring systems.



5. Arduino IDE

Arduino IDE is an open-source, cross-platform software application used to write, compile, and upload code to Arduino boards and compatible microcontrollers like NodeMCU or ESP32. It provides a simple interface with a code editor, a serial monitor for debugging, and built-in libraries for various hardware components. The IDE supports C and C++ programming, making it beginner-friendly while offering advanced capabilities for experienced developers. With a vast community and extensive library support, Arduino IDE simplifies creating projects like robotics, home automation, and IoT devices. Its versatility and ease of use make it a cornerstone of DIY electronics and prototyping.





CODE Uses

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

```
char auth[] = "2f8482b704cf41fbadb23d937ce10777"; // the auth code that you got on your gmail
char ssid[] = "Nalla_Manus"; // username or ssid of your WI-Fi
char pass[] = "ninimaab"; // password of your Wi-Fi
```

```
void setup()
{
  // Debug console
  Serial.begin(9600);
  pinMode(D1,OUTPUT); //extend these to D8 if you are using a 8 pin relay
  pinMode(D2,OUTPUT);
  pinMode(D3,OUTPUT);
  pinMode(D4,OUTPUT);
```

```
digitalWrite(D1,HIGH); // Make it low if you want everything to go off
digitalWrite(D2,HIGH); // in case of a power cut
digitalWrite(D3,HIGH);
digitalWrite(D4,HIGH);
Blynk.begin(auth, ssid, pass);
}
```

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

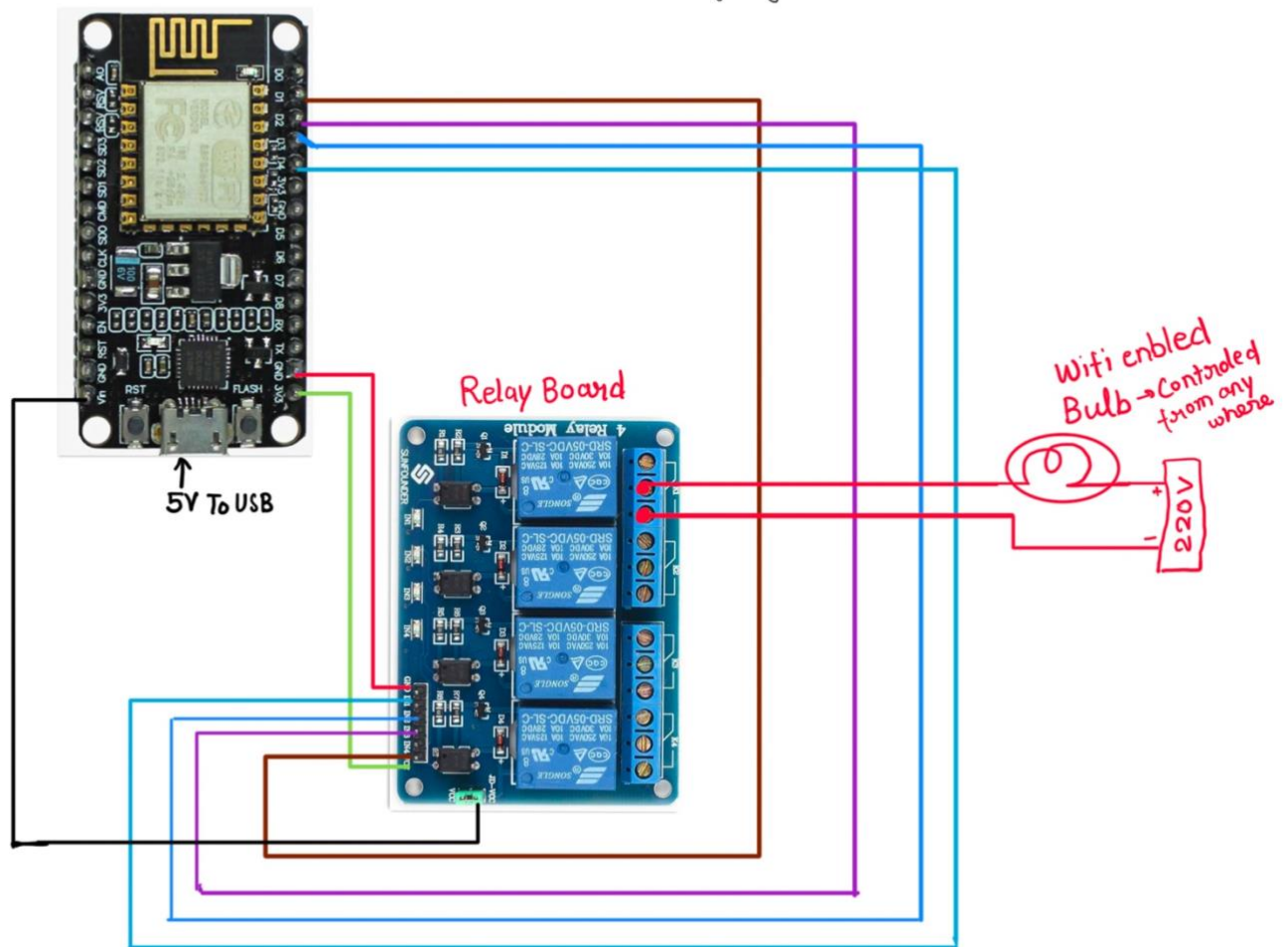

CIRCUIT DIAGRAM No.1

NOTE: Use the circuit diagram to see if your node MCU can supply the necessary 5V to the relay board. This Setup Does not use Arduino Uno.

Home Automation Circuit diagram

Node MCU (ESP8266)

by -Mayank Patel (darksprut.mpe@gmail.com)



CIRCUIT DIAGRAM No.2

NOTE:- If you DO NOT plan on making a case for the circuit. This uses Arduino as Node MCU didn't supply enough power to trigger relay ON/OFF.

