**Abstract**

***Today India getting digital, many IoT platforms are being introduced. With the increase in the technology, the applications have also vastly increased. Many open-source IoT Devices like Node MCU,ESP8266, raspberry pi are into market available at affordable prices. Hence, these technologies can be incorporated to make many of the solutions for the new automation. These can be further be implemented to many large-scale systems. These systems may include many devices that can be controlled with many other IoT devices.***

**Introduction**.

In this project, we will be using NODE MCU as a main microcontroller. The microcontroller has the capability of Wi-Fi communication. Hence, it can be used to control many devices. The Concept in this project is to control the lighting systems or devices in home wirelessly or remotely using IoT. The Node MCU is connected to all the devices to control the system. The advantage being that the devices can be controlled from one place ideal for differently abled people. With this system, it can be turned on and off remotely.

**Block Diagram**

Device-1

Node MCU

Device-2

Device-3

Power Supply

Program Flow Chart

Turn on repective pins in ESP8266

HTML Code for web page

Connect to hotspot using password 12345678 and login to the HTML page using 192.168.4.1

Turn on Hotspot of ESP8266 and wait for connection

Initiate Microcontroller

Program

/\*

\* Sketch: ESP8266\_LED\_Control\_02A

\* Now with added CSS

\* Control an LED from a web browser

\* Intended to be run on an ESP8266

\*

\* connect to the ESP8266 AP then

\* use web broswer to go to 192.168.4.1

\*

\*/

#include <ESP8266WiFi.h>

const char WiFiPassword[] = "12345678";

const char AP\_NameChar[] ="Home Automation" ;

WiFiServer server(80);

String header = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";

String html\_1 = "<!DOCTYPE html><html><head><meta name='viewport' content='width=device-width, initial-scale=1.0'/><meta charset='utf-8'><style>body {font-size:140%;} #main {display: table; margin: auto; padding: 0 10px 0 10px; } h2,{text-align:center; } .button { padding:10px 10px 10px 10px; width:100%; background-color: #4CAF50; font-size: 120%;}</style><title>LED Control</title></head><body><div id='main'><h2>Home Automation</h2>";

String html\_2 = "<center><form><input formaction='LED1ON' class='button' type='submit' value='Light\_1 ON' ><input formaction='LED1OFF' class='button' type='submit' value='Light\_1 OFF' ></form></center><br>";

String html\_3 = "<center><form><input formaction='LED2ON' class='button' type='submit' value='Light\_2 ON' ><input formaction='LED2OFF' class='button' type='submit' value='Light\_2 OFF' ></form></center><br>";

String html\_4 = "<center><form><input formaction='LED3ON' class='button' type='submit' value='Light\_3 ON' ><input formaction='LED3OFF' class='button' type='submit' value='Light\_3 OFF' ></form></center><br>";

String html\_5 = "<center><form><input formaction='LED4ON' class='button' type='submit' value='Light\_4 ON' ><input formaction='LED4OFF' class='button' type='submit' value='Light\_4 OFF' ></form></center><br>";

String html\_6 = "</div></body></html>";

String request = "";

int LED1\_Pin = D5;

int LED2\_Pin = D3;

int LED3\_Pin = D0;

int LED4\_Pin = D1;

//int LED5\_Pin = D2;

void setup()

{

pinMode(LED1\_Pin, OUTPUT);

pinMode(LED2\_Pin, OUTPUT);

pinMode(LED3\_Pin, OUTPUT);

pinMode(LED4\_Pin, OUTPUT);

// pinMode(LED5\_Pin, OUTPUT);

boolean conn = WiFi.softAP(AP\_NameChar, WiFiPassword);

server.begin();

} // void setup()

void loop()

{

// Check if a client has connected

WiFiClient client = server.available();

if (!client) { return; }

// Read the first line of the request

request = client.readStringUntil('\r');

if ( request.indexOf("LED1ON") > 0 ) { digitalWrite(LED1\_Pin, HIGH); }

else if ( request.indexOf("LED1OFF") > 0 ) { digitalWrite(LED1\_Pin, LOW); }

else if ( request.indexOf("LED2ON") > 0 ) { digitalWrite(LED2\_Pin, HIGH); }

else if ( request.indexOf("LED2OFF") > 0 ) { digitalWrite(LED2\_Pin, LOW); }

else if ( request.indexOf("LED3ON") > 0 ) { digitalWrite(LED3\_Pin, HIGH); }

else if ( request.indexOf("LED3OFF") > 0 ) { digitalWrite(LED3\_Pin, LOW); }

else if ( request.indexOf("LED4ON") > 0 ) { digitalWrite(LED4\_Pin, HIGH); }

else if ( request.indexOf("LED4OFF") > 0 ) { digitalWrite(LED4\_Pin, LOW); }

client.flush();

client.print( header );

client.print( html\_1 );

client.print( html\_2 );

client.print( html\_3 );

client.print( html\_4 );

client.print( html\_5 );

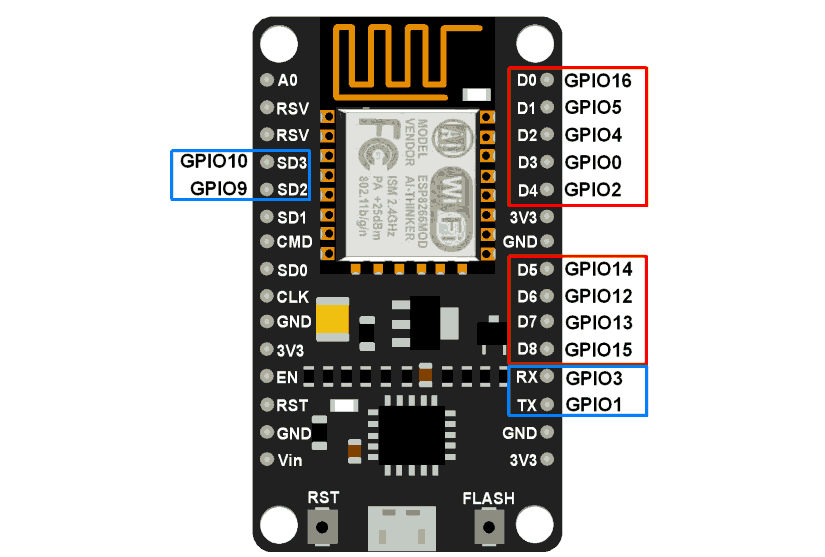
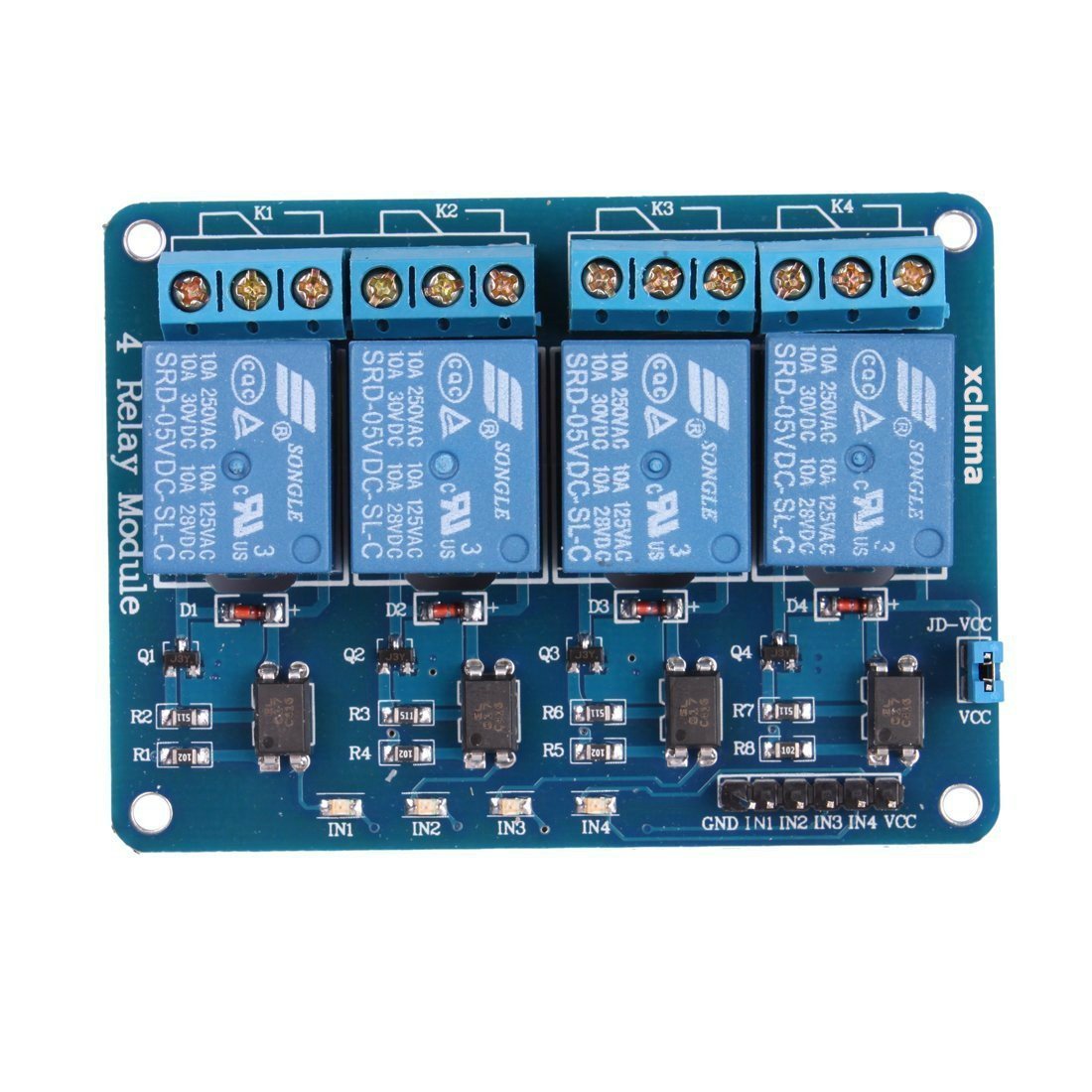
client.print( html\_6 );

delay(5);

// The client will actually be disconnected when the function returns and 'client' object is detroyed

} // void loop()

Circuit Diagram



Output

Input 230V

Components.

The main components of the projects include

1. **NODE MCU**

NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

Hardware typically used is a circuit board functioning as a [dual in-line package](https://en.wikipedia.org/wiki/Dual_in-line_package) (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on [breadboards](https://en.wikipedia.org/wiki/Breadboard). The design was initially was based on the ESP-12 module of the [ESP8266](https://en.wikipedia.org/wiki/ESP8266), which is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica) Xtensa LX106 core, widely used in IoT applications

* The ESP8266 chip requires 3.3V power supply voltage. It should not be powered with 5 volts like other arduino boards.
* NodeMCU ESP-12E dev board can be connected to 5Vusing **micro USB** connector or **Vin**pin available on board.
* The I/O pins of ESP8266 communicate or input/output max 3.3V only. i.e. the pins are **NOT**5V tolerant input.

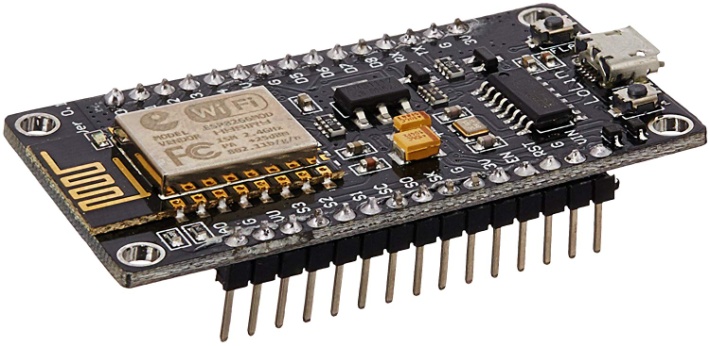
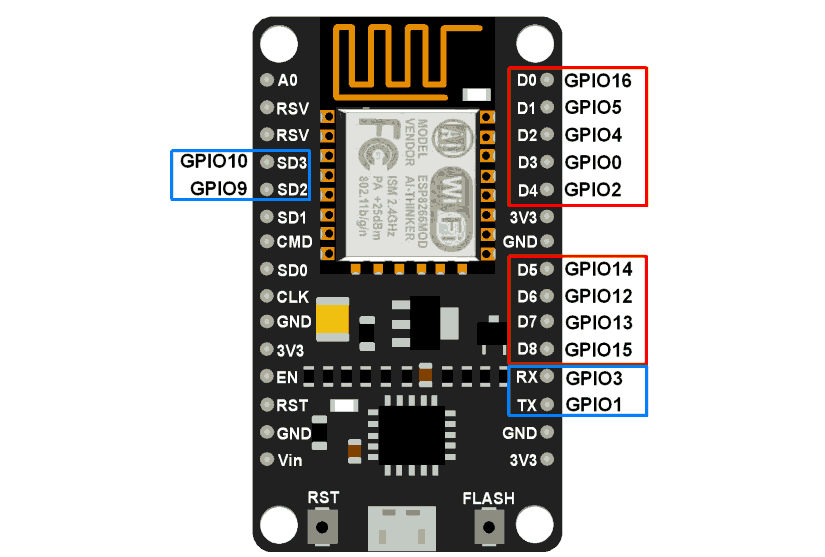


Figure 1: NODE MCU ESP8266

Pinouts

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behavior can be controlled at the run time.

NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care is that NodeMCU Dev kit pins are numbered differently than internal GPIO notations of ESP8266 as shown in below figure and table. For example, the D0 pin on the NodeMCU Dev kit is mapped to the internal GPIO pin 16 of ESP8266.

| Pin Names on NodeMCU Development Kit | ESP8266 Internal GPIO Pin number |
| --- | --- |
| D0 | GPIO16 |
| D1 | GPIO5 |
| D2 | GPIO4 |
| D3 | GPIO0 |
| D4 | GPIO2 |
| D5 | GPIO14 |
| D6 | GPIO12 |
| D7 | GPIO13 |
| D8 | GPIO15 |
| D9/RX | GPIO3 |
| D10/TX | GPIO1 |
| D11/SD2 | GPIO9 |
| D12/SD3 | GPIO10 |

Below table gives NodeMCU Dev Kit IO pins and ESP8266 internal GPIO pins mappin

Features

* Finally, programable WiFi module.
* Arduino-like (software defined) hardware IO.
* Can be programmed with the simple and powerful Lua programming language or Arduino IDE.
* USB-TTL included, plug & play.
* 10 GPIOs D0-D10, PWM functionality, IIC and SPI communication, 1-Wire and ADC A0 etc. all in one board.
* Wifi networking (can be used as access point and/or station, host a web server), connect to internet to fetch or upload data.
* Event-driven API for network applications.
* PCB antenna.

1. **Power Supply.**
2. Power Supply used is of 5V.
3. The power supply supplies power to the NODE MCU and relay module.
4. The power supply is of 1A SMPS type.

**Working**

1. The node MCU is programmed to operate via Wi-fi.
2. The pins of Node MCU are connected to controller.
3. The Controller here is nothing but the relay or transistor.
4. The node MCU can be accessed using a smart phone.
5. The options can be seen on the smart phone.
6. Connect to Hotspot names Home automation. Password is 12345678
7. Open any browser and type Ip address -192.1687.4.1
8. The HTML page is visible.
9. Using the options on web page the devices can be controlled.

**Advantages**.

1. Since It is wireless, it can be controlled from anywhere from the network.
2. Other devices up to 8 numbers can be added.
3. Single point control for all the devices.
4. Can be accessed by anyone who has the authorization.

**Disadvantages**.

1. Since it is IoT, Network availability is important.
2. After a certain range, other equipment to increase the range have to be added.
3. Only authorized personal will be allowed to operate.