

## HOME AUTOMATION USING NODE MCU

In this tutorial, we will build basic home automation project using a Node Mcu.

### Components Required:

1. 1\*Node Mcu
2. 2\*Led

### About Node Mcu:

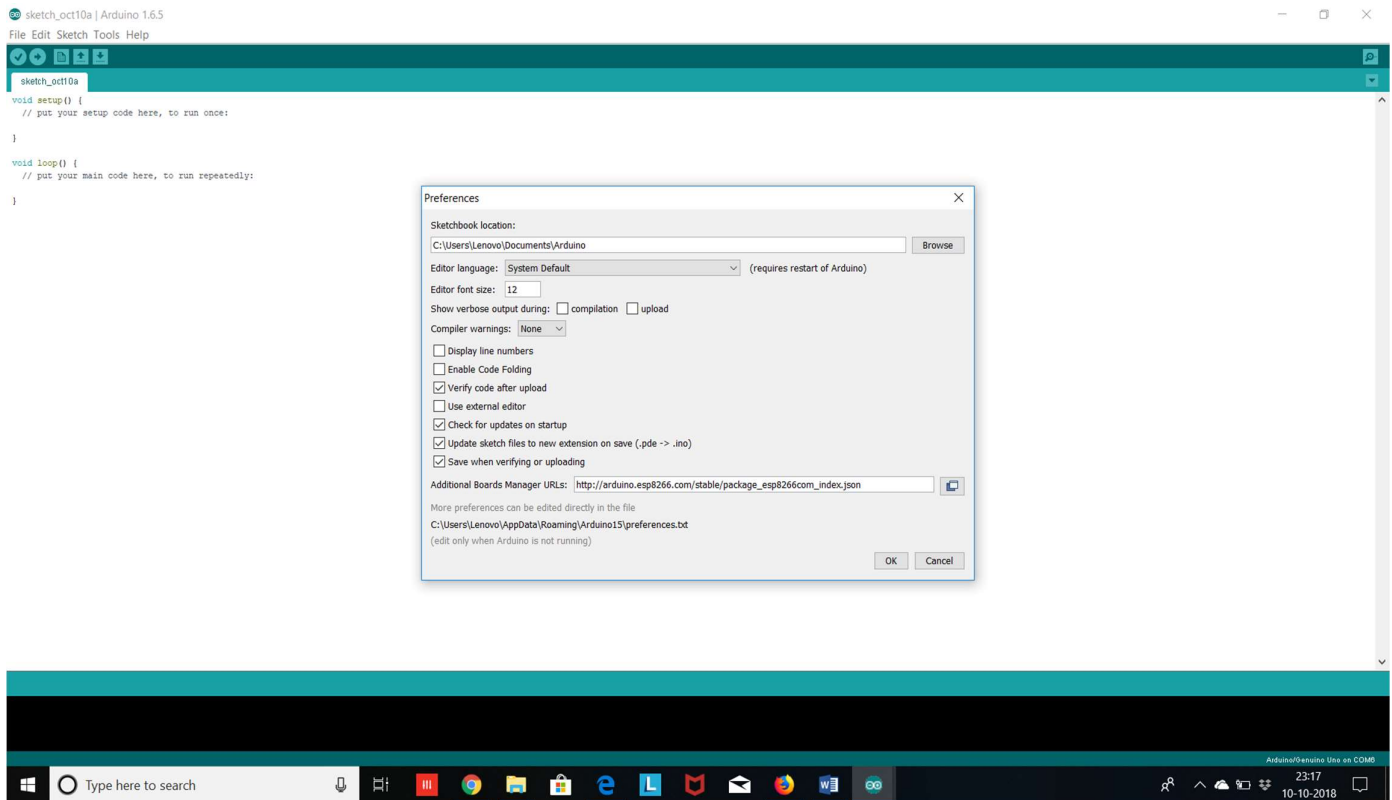
Node Mcu is an open source IOT platform that works on the ESP8266 microcontroller. It provides easy connectivity to internet and also comes in a compact size. Node Mcu works on Lua Scripting language but in this tutorial we will be using the Arduino IDE to code for our Node Mcu.



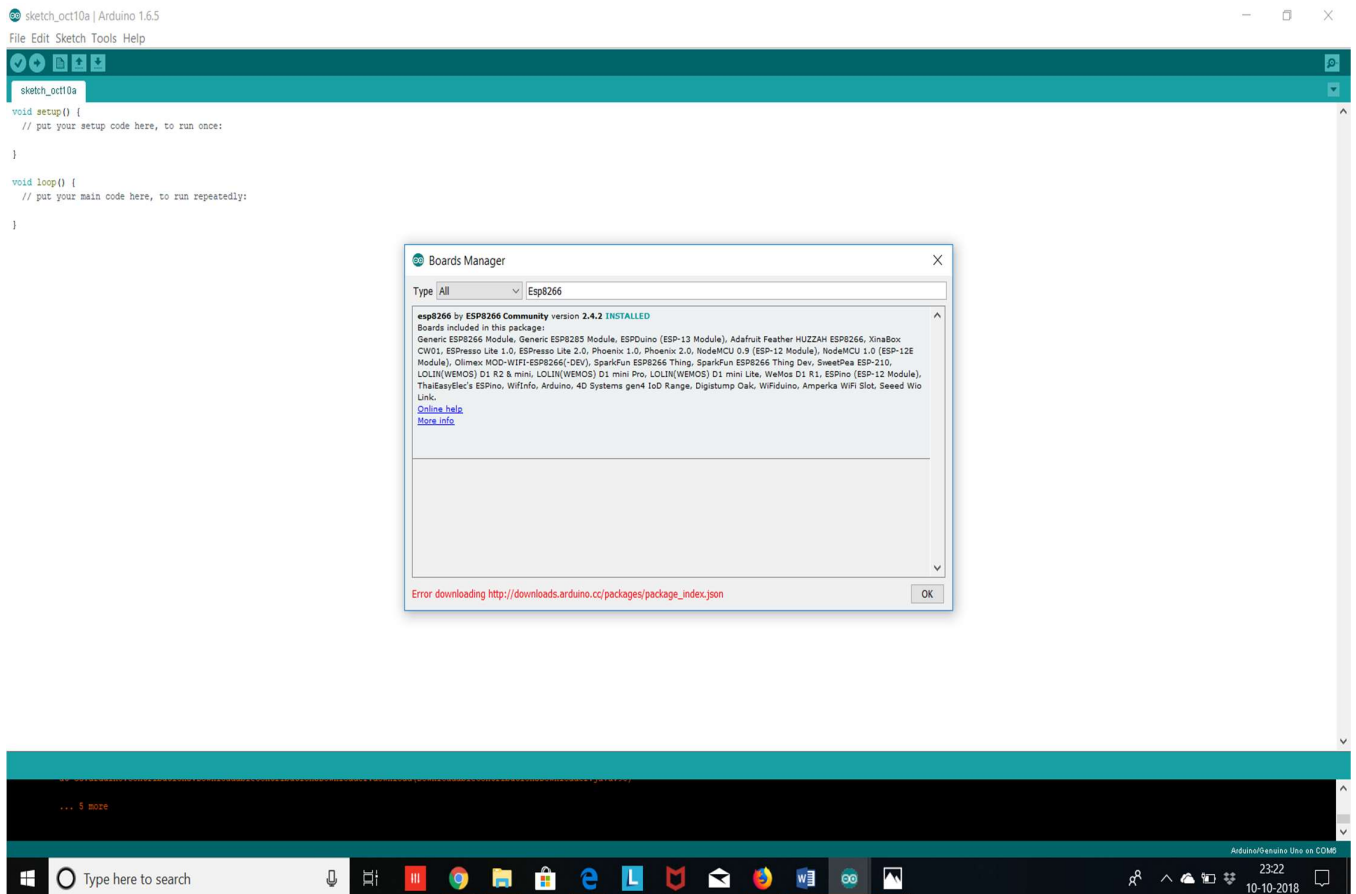
### Setting up Node Mcu in Arduino IDE:

1. Open Arduino IDE .
2. Go to Files->Preferences and copy the given url in additional board manager option.

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



3. Now go to Tools->Board Manager and search for ESP8266 and install the given board.



4. Now go to Tools->Board->Node mcu 1.0(12-E)

You are ready for writing the code and uploading to your node mcu.

### Code:

node\_mcu\_take\_2 | Arduino 1.6.5

File Edit Sketch Tools Help

```
node_mcu_take_2
#include <ESP8266WiFi.h>

const char* ssid = "nachi"; //Enter your Wifi name
const char* password = "1234567890"; //Enter your Wifi Password
int ledPin = D0;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(ledPin, OUTPUT);

  digitalWrite(ledPin, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
}
```

node\_mcu\_take\_2 | Arduino 1.6.5

File Edit Sketch Tools Help

```
void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Match the request

  int value = LOW;
  if (request.indexOf("/MOTOR=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
  if (request.indexOf("/MOTOR=OFF") != -1) {
    digitalWrite(ledPin, LOW);
    value = LOW;
  }

  // Set ledPin according to the request
  //digitalWrite(ledPin, value);

  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); // do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  client.print("Motor is now: ");

  if(value == HIGH) {
    client.print("On");
  } else {
    client.print("Off");
  }
}
```

```

if(value == HIGH) {
  client.print("On");
} else {
  client.print("Off");
}
client.println("<br><br>");
client.println("<a href=\"/MOTOR=ON\"><button>Turn On </button></a>");
client.println("<a href=\"/MOTOR=OFF\"><button>Turn Off </button></a><br />");
client.println("</html>");

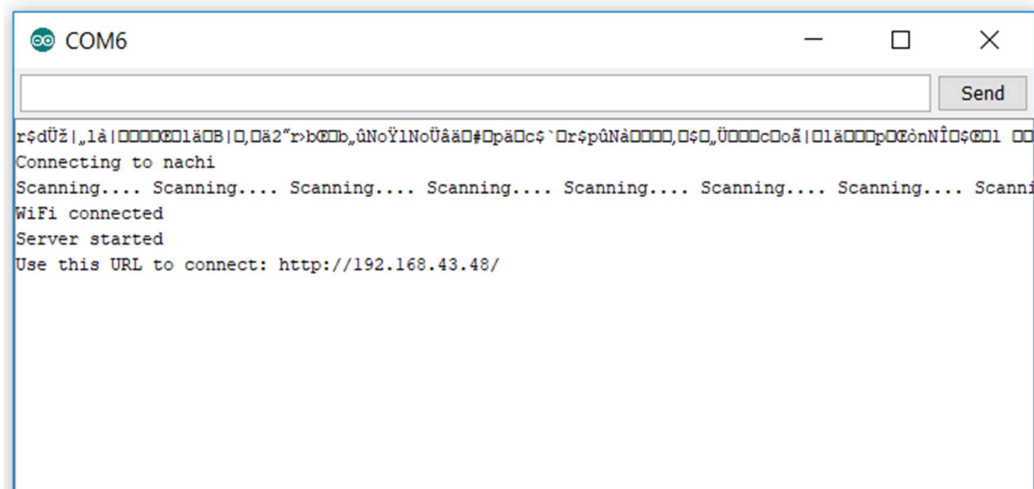
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}

```

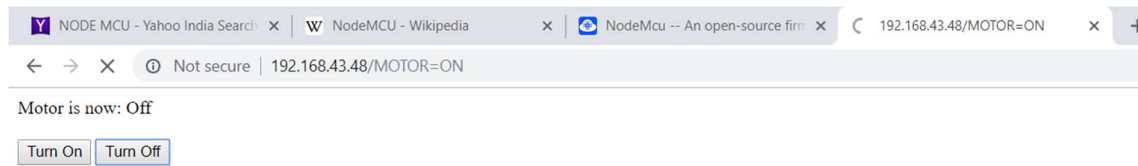
5.Compile and upload the code .

6.Connect positive terminal of the led onto the D0 pin and the negative terminal to the GND pin.

7.Once the code is uploaded , run the serial monitor and copy the ip address of the node mcu onto your web browser.



8. A webpage will open up that will enable you to control the led connected at D0.



You can now replace the led by any output device to achieve any home automated task using your web browser !