# Node MCU

<https://www.teachmemicro.com/intro-nodemcu-arduino/>

### Step 1: Connect your NodeMCU to your computer

You need a USB micro B cable to connect the board. Once you plugged it in, a blue LED will start flashing. If your computer is not able to detect the NodeMCU board, you may need to download the driver on [this page](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers).

### Step 2: Open Arduino IDE

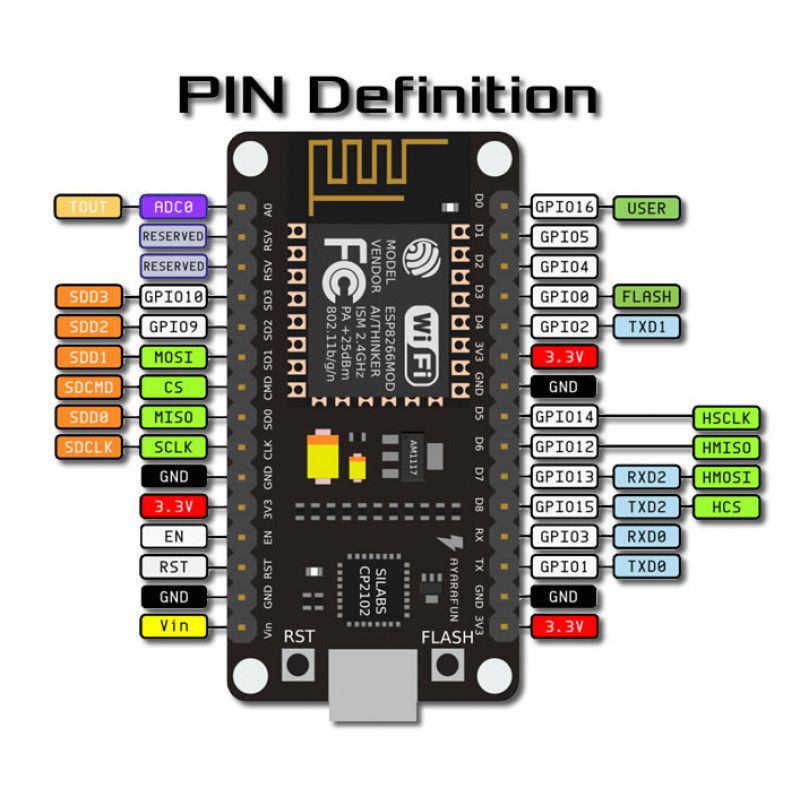
You need to have at least Arduino IDE version 1.6.4 to proceed with this.

Go to File > Preferences. In the "Additional Boards Manager URLs" field, type (or copy-paste) http://arduino.esp8266.com/stable/package\_esp8266com\_index.json. Don't forget to click OK!

Then go to  Tools > Board > Board Manager. Type "esp8266" in the search field. The entry "esp8266 by ESP8266 Community" should appear. Click that entry and look for the install button on the lower right.

# Wifi connection

|  |
| --- |
| #include <ESP8266WiFi.h> |
|  |  |
|  | const char\* ssid="Your SSID – network-properties"; |
|  | const char\* password = "Your Password"; |
|  |  |
|  | int ledPin = 13; |
|  |  |
|  | void setup() { |
|  |  |
|  | pinMode(ledPin,OUTPUT); |
|  | digitalWrite(ledPin,LOW); |
|  |  |
|  | Serial.begin(115200); |
|  | Serial.println(); |
|  | Serial.print("Wifi connecting to "); |
|  | Serial.println( ssid ); |
|  |  |
|  | WiFi.begin(ssid,password); |
|  |  |
|  | Serial.println(); |
|  | Serial.print("Connecting"); |
|  |  |
|  | while( WiFi.status() != WL\_CONNECTED ){ |
|  | delay(500); |
|  | Serial.print("."); |
|  | } |
|  |  |
|  | digitalWrite( ledPin , HIGH); |
|  | Serial.println(); |
|  |  |
|  | Serial.println("Wifi Connected Success!"); |
|  | Serial.print("NodeMCU IP Address : "); |
|  | Serial.println(WiFi.localIP() ); |
|  |  |
|  | } |
|  |  |
|  | void loop() { |
|  | // put your main code here, to run repeatedly: |
|  |  |
|  | } |



Led matrix

<https://www.instructables.com/id/Interface-LED-Dot-Matrix-8x8-With-NodeMCU/>

# connect to restapi

#include <ESP8266HTTPClient.h>

#include <ESP8266WiFi.h>

const char\* ssid="NETGEAR43";

const char\* password = "dynamiclake424";

int ledPin = 13;

void setup() {

pinMode(ledPin,OUTPUT);

digitalWrite(ledPin,LOW);

Serial.begin(115200);

Serial.println();

Serial.print("Wifi connecting to ");

Serial.println( ssid );

WiFi.begin(ssid,password);

Serial.println();

Serial.print("Connecting");

while( WiFi.status() != WL\_CONNECTED ){

delay(500);

Serial.print(".");

}

digitalWrite( ledPin , HIGH);

Serial.println();

Serial.println("Wifi Connected Success!");

Serial.print("NodeMCU IP Address : ");

Serial.println(WiFi.localIP() );

}

void loop() {

if(WiFi.status()== WL\_CONNECTED){ //Check WiFi connection status

HTTPClient http; //Declare object of class HTTPClient

http.begin("http://192.168.1.21:8000/esp\_check/"); //Specify request destination- ip address of pc- ipconfig

http.addHeader("Content-Type", "text/plain"); //Specify content-type header

int httpCode = http.POST("Message from ESP8266"); //Send the request

String payload = http.getString(); //Get the response payload

Serial.println(httpCode); //Print HTTP return code

Serial.println(payload); //Print request response payload

http.end(); //Close connectiona

}else{

Serial.println("Error in WiFi connection");

}

delay(30000); //Send a request every 30 seconds

}

# Node mcu as a server

Web [documentation](https://randomnerdtutorials.com/esp32-esp8266-input-data-html-form/)

/\*\*\*\*\*\*\*\*\*

  Rui Santos

  Complete project details at https://RandomNerdTutorials.com/esp32-esp8266-input-data-html-form/

  Permission is hereby granted, free of charge, to any person obtaining a copy

  of this software and associated documentation files.

  The above copyright notice and this permission notice shall be included in all

  copies or substantial portions of the Software.

\*\*\*\*\*\*\*\*\*/

#include <Arduino.h>

#ifdef ESP32

  #include <WiFi.h>

  #include <AsyncTCP.h>

#else

  #include <ESP8266WiFi.h>

  #include <ESPAsyncTCP.h>

#endif

#include <ESPAsyncWebServer.h>

AsyncWebServer server(80);

// REPLACE WITH YOUR NETWORK CREDENTIALS

const char\* ssid = "DESKTOP-Abhijith M";

const char\* password = "initial#024";

const char\* PARAM\_INPUT\_1 = "input1";

const char\* PARAM\_INPUT\_2 = "input2";

const char\* PARAM\_INPUT\_3 = "input3";

// HTML web page to handle 3 input fields (input1, input2, input3)

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML><html><head>

  <title>ESP Input Form</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  </head><body>

  <form action="/get">

    input1: <input type="text" name="input1">

    <input type="submit" value="Submit">

  </form><br>

  <form action="/get">

    input2: <input type="text" name="input2">

    <input type="submit" value="Submit">

  </form><br>

  <form action="/get">

    input3: <input type="text" name="input3">

    <input type="submit" value="Submit">

  </form>

</body></html>)rawliteral";

void notFound(AsyncWebServerRequest \*request) {

  request->send(404, "text/plain", "Not found");

}

void setup() {

  Serial.begin(115200);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  if (WiFi.waitForConnectResult() != WL\_CONNECTED) {

    Serial.println("WiFi Failed!");

    return;

  }

  Serial.println();

  Serial.print("IP Address: ");

  Serial.println(WiFi.localIP());

  // Send web page with input fields to client

  server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

    request->send\_P(200, "text/html", index\_html);

  });

  // Send a GET request to <ESP\_IP>/get?input1=<inputMessage>

  server.on("/get", HTTP\_POST, [] (AsyncWebServerRequest \*request) {

    String inputMessage;

    String inputParam;

    int params = request->params();

    Serial.printf("%d params sent in\n", params);

    for (int i = 0; i < params; i++)

    {

        AsyncWebParameter \*p = request->getParam(i);

        if (p->isFile())

        {

            Serial.printf("\_FILE[%s]: %s, size: %u", p->name().c\_str(), p->value().c\_str(), p->size());

        }

        else if (p->isPost())

        {

//            Serial.printf("\_POST[%s]: %s", p->name().c\_str(), p->value().c\_str());

            if ( p->name().equals("input1")){

//              inputMessage = p->value().c\_str();

              Serial.printf("input1 is  %s\n", p->value().c\_str());

            }

            if ( p->name().equals("input2")){

//              inputMessage = p->value().c\_str();

              Serial.printf("input2 is %s", p->value().c\_str());

            }

        }

        else

        {

            Serial.printf("\_GET[%s]: %s", p->name().c\_str(), p->value().c\_str());

        }

    }

    /\*

    // GET input1 value on <ESP\_IP>/get?input1=<inputMessage>

    if (request->hasParam(PARAM\_INPUT\_1)) {

      inputMessage = request->getParam(PARAM\_INPUT\_1)->value();

      inputParam = PARAM\_INPUT\_1;

    }

    // GET input2 value on <ESP\_IP>/get?input2=<inputMessage>

    else if (request->hasParam(PARAM\_INPUT\_2)) {

      inputMessage = request->getParam(PARAM\_INPUT\_2)->value();

      inputParam = PARAM\_INPUT\_2;

    }

    // GET input3 value on <ESP\_IP>/get?input3=<inputMessage>

    else if (request->hasParam(PARAM\_INPUT\_3)) {

      inputMessage = request->getParam(PARAM\_INPUT\_3)->value();

      inputParam = PARAM\_INPUT\_3;

    }

    else {

      inputMessage = "No message sent";

      inputParam = "none";

    }

    Serial.println(inputMessage);\*/

    request->send(200, "text/html", "HTTP GET request sent to your ESP on input field ("

                                     + inputParam + ") with value: " + inputMessage +

                                     "<br><a href=\"/\">Return to Home Page</a>");

  });

  server.onNotFound(notFound);

  server.begin();

}

void loop() {

}