# ESP8266-12E Quick Guide

October 10, 2016

1. Pin mapping of ESP8266-12E:
   - `http://amazingrobots.net/resources/nodemcu_pinout/`
2. Motor shield:
   - Pin mapping of motor shield: `http://amazingrobots.net/resources/motor_shield_diagram/`
   - Pin name and function descriptions for motor shield: `https://smartarduino.gitbooks.io/user-mannual-for-esp-12e-motor-shield/content/interface.html`
   - Note that in above descriptions, Pin D1 and D2 are for motor speed controls, and Pin D3 and D4 are for motor direction controls.
   - If you are instrested, go through the Arduino programming tutorial below first and come back to see an example of controlling the motor at section ”Node MCU Motor Shield” here: `https://hackaday.io/project/8856/logs`
3. ESP8266 Arduino programming tutorial:
   - Step 1: Install driver and blink LEDs
     - Go to url: `http://www.instructables.com/id/Programming-the-ESP8266-12E-using-Arduino-software/?ALLSTEPS`
     - For step 2, drivers for all OSes can be found here: `https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx`
     - Follow steps 3-13 exactly as written
       - Be aware, in step 6, the menu may be in Edit → Preferences instead of Tools
       - Note that in the DoubleBlink.ino code in last step, the author forgets to put line breaks before line ”digitalWrite(BUILTIN_LED2, HIGH);” and line ”delay(1000);”

  - Pin 2 (GPIO2) is equivalent to pin D4, and pin 16 (GPIO16) is the same as D0. So, one can also write

    const short int BUILTIN_LED1 = D4;<br>
    const short int BUILTIN_LED2 = D0;

    at the top instead.

- Step 2: Control the LED with a web server
  - Go to url:
    `http://www.instructables.com/id/Programming-a-HTTP-Server-on-ESP-8266-12E/?ALLSTEPS`
  - Follow all steps, feel free to change the SSID and PASSWORD to your faviourate ones in WiFi.softAP() function
  - Make sure to open the IDE Serial Monitor from the beginning to see all output infomation
  - You can connect to your ESP's AP from your cell phone too
  - For purpose of reference, below is what your codes should look like in the last step:

```
#include <ESP8266WiFi.h>

WiFiServer server(80); //Initialize the server on Port 80
const short int LED_PIN = 16;//GPIO16

void setup() {
    WiFi.mode(WIFI_AP); //Our ESP8266-12E is an AccessPoint
    WiFi.softAP("Hello_IoT", "12345678"); // Provide the (SSID, password); .
    server.begin(); // Start the HTTP Server

    //Looking under the hood
    Serial.begin(115200);
    IPAddress HTTPS_ServerIP= WiFi.softAPIP(); // Obtain the IP of the Server
    Serial.print("Server IP is: "); // Print the IP to the monitor window
    Serial.println(HTTPS_ServerIP);

    pinMode(LED_PIN, OUTPUT); //GPIO16 is an OUTPUT pin;
    digitalWrite(LED_PIN, LOW); //Initial state is ON
}

void loop() {
    WiFiClient client = server.available();
    if (!client) {
      return;
    }
    //Looking under the hood
    Serial.println("Somebody has connected :)");

    //Read what the browser has sent into a String class
    //and print the request to the monitor
    String request = client.readStringUntil(’\r’);
    //Looking under the hood
    Serial.println(request);

    // Handle the Request
    if (request.indexOf("/OFF") != -1){
      digitalWrite(LED_PIN, HIGH); }
    else if (request.indexOf("/ON") != -1){
      digitalWrite(LED_PIN, LOW);
    }

    // Prepare the HTML document to respond and add buttons:
    String s = "HTTP/1.1 200 OK\r\n";
    s += "Content-Type: text/html\r\n\r\n";
    s += "<!DOCTYPE HTML>\r\n<html>\r\n";
    s += "<br><input type=\"button\" name=\"b1\" value=\"LED On\"";
    s += " onclick=\"location.href=’/ON’\">";
    s += "<br><br><br>";
    s += "<br><input type=\"button\" name=\"b1\" value=\"LED Off\"";
    s += " onclick=\"location.href=’/OFF’\">";
    s += "</html>\n";

    //Serve the HTML document to the browser.
    client.flush(); //clear previous info in the stream
    client.print(s); // Send the response to the client
    delay(1);
    Serial.println("Client disonnected"); //Looking under the hood
}
```

- Step 3: Check out more documentations and examples!
  - Go to url: `https://github.com/esp8266/Arduino`
  - More related documentations can be found under ”doc” folder
  - Examples for each ESP8266 Arduino library can be found under ”libraries/LIBRARY_NAME/examples” folder.

    For intance, more examples about the ESP8266WiFi.h libaray that we just used in step2 and 3 can be found at ”libraries/ESP8266WiFi/examples”
  - Note that they may use different pin numbers for LED in the example code, so things may not work as expected straight out of the box.