

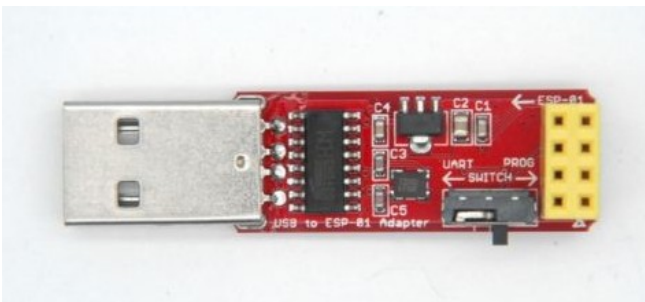
ESP-01 / ESP8266 Guide

The ESP-01 is a small board version of the Espressif Systems ESP8266, which has a full WiFi stack and two GPIO pins. The ESP-01 also features 1MB of onboard flash and can be programmed using the Arduino IDE software.

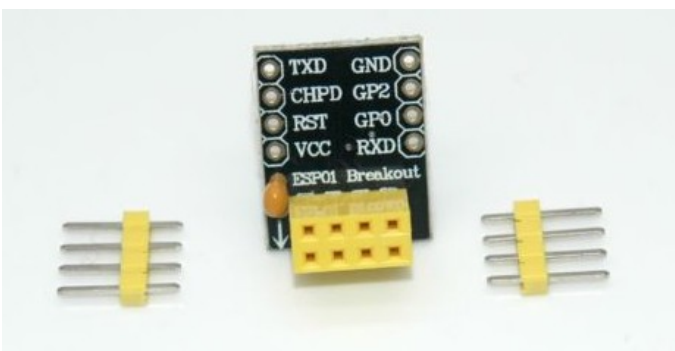


The original blue PCB version came with 512k, but all the versions for sale now with black PCBs include 1MB flash storage and are sometimes referred to as ESP-01S, and sell for around £2 on eBay.

To follow this guide you'll also need a few accessories: A USB programmer with a programming and uart switch. Around £4.50 on eBay. (Don't be tempted to buy the cheaper USB programmer without the switch as it didn't work for me on Windows 7 with the Arduino software).



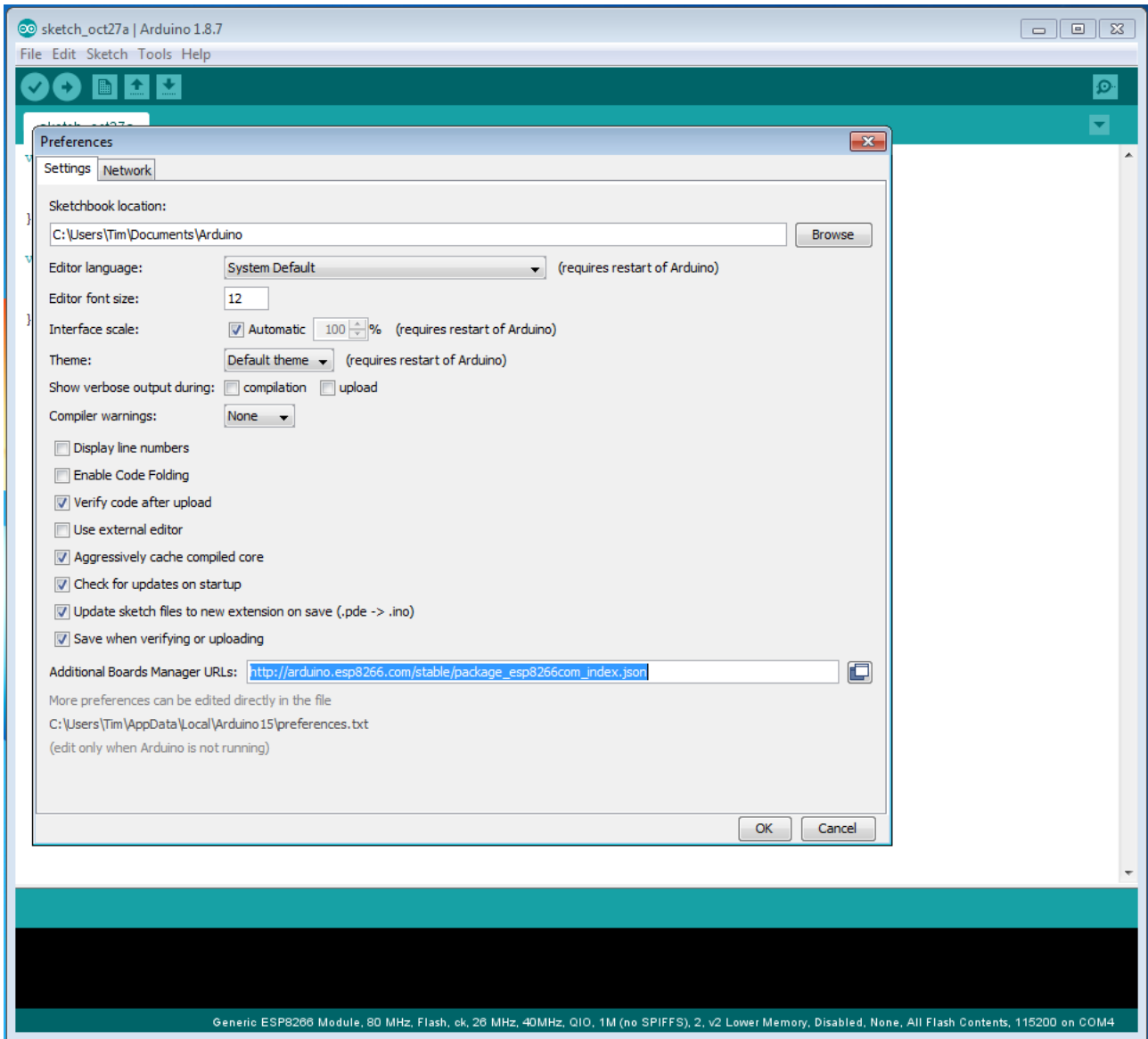
also a £2.25 breadboard breakout PCB



You'll also need a mini breadboard, an LED, a 220 ohm resistor and some breadboard cables.

First we need to install the Arduino IDE software from <https://www.arduino.cc/en/Main/Software> Currently at version 1.8.7 and available for Windows, Mac and Linux.

Once installed run the Arduino software and go to File → Preferences → Settings and in the Additional Boards Manager box paste in:
http://arduino.esp8266.com/stable/package_esp8266com_index.json



Then go to Tools → Board → Boards Manager → type esp into the search box, select the esp8266 by ESP8266 Community box and then click Install.

Now when you go to Tools → Boards you'll be able to select Generic ESP8266 module.

We're now ready to program the board with our first Sketch.

The first project allows us to turn and LED on or off using our mobile phones web browser to operate the On and Off buttons.

Here's the Sketch we're going to flash to the board:

```
#include <ESP8266WiFi.h>

const char* ssid = "yourrouterID";//type your ssid
const char* password = "yourRoutersPassword";//type your password

int ledPin = 2; // GPIO2 of ESP8266
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

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

  WiFi.begin(ssid, password);

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

// 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("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
}
if (request.indexOf("/LED=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("Led pin is now: ");

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

client.println("<br><br>");
client.println("Click <a href=\"/LED=ON\">here</a> turn the LED
on pin 2 ON<br>");
client.println("Click <a href=\"/LED=OFF\">here</a> turn the LED
on pin 2 OFF<br>");
client.println("<input type=button style=width:100px;height:45px
value= ON onmousedown=location.href='/LED=ON\">");
client.println("<input type=button style=width:100px;height:45px

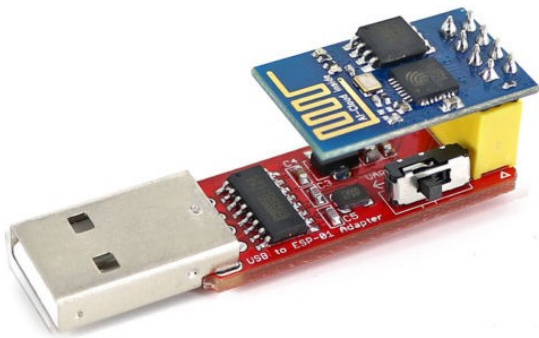
```

```
value= OFF onmousedown=location.href='/LED=OFF\''>");
  client.println("</html>");

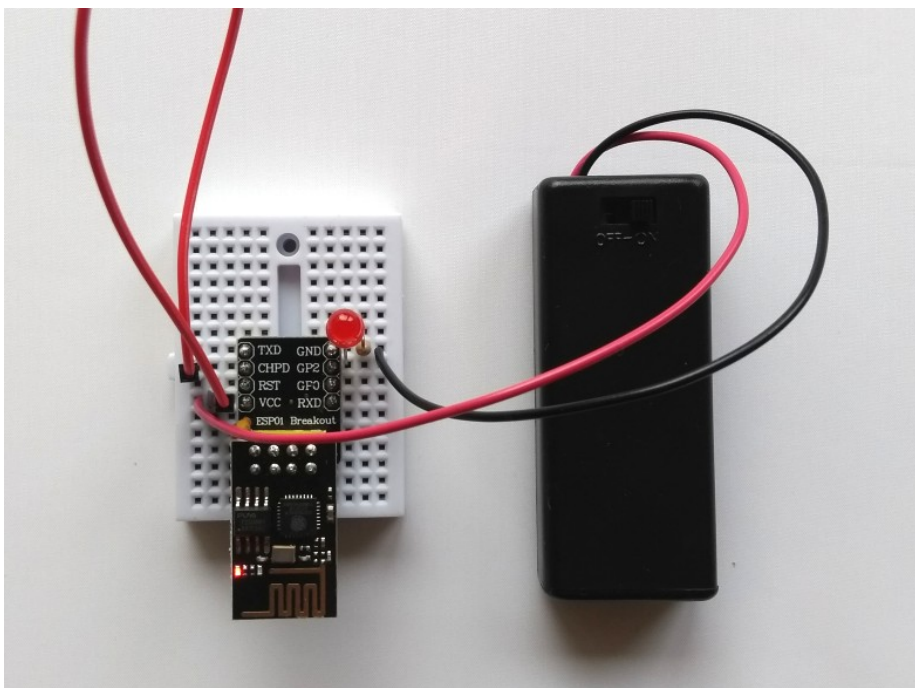
  delay(1);
  Serial.println("Client disconnected");
  Serial.println("");
}
```

The Sketch generates a web page with two buttons on it and serves it on port 80. You type the IP address that's been allocated by your router to the ESP-01 board into your mobile phones web browser and you can turn the LED that's connected to GPIO2 on and off remotely.

This is how the board sits on the programmer. The switch should be moved right so it's nearest to the yellow block for programming mode. If you have the black PCB ESP-01S 1M board change the setting Tools → Flash size to 1MB (128K SPIFFS) before programming. (SPIFFS is separate storage space for images you might use on a webpage)

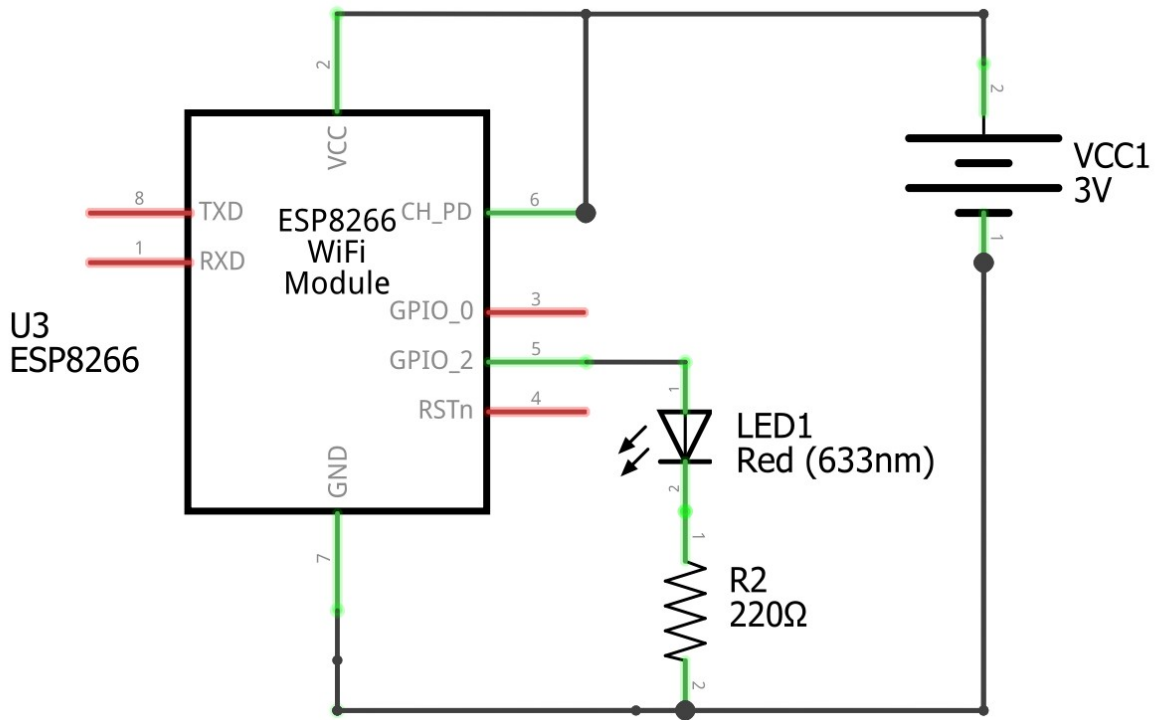


It will take a few minutes to compile and upload the file to the board. When you see “Done Uploading and 100%” in the bottom of the window, it's finished. Remove the ESP-01 from the programmer and insert it into the breadboard adapter.

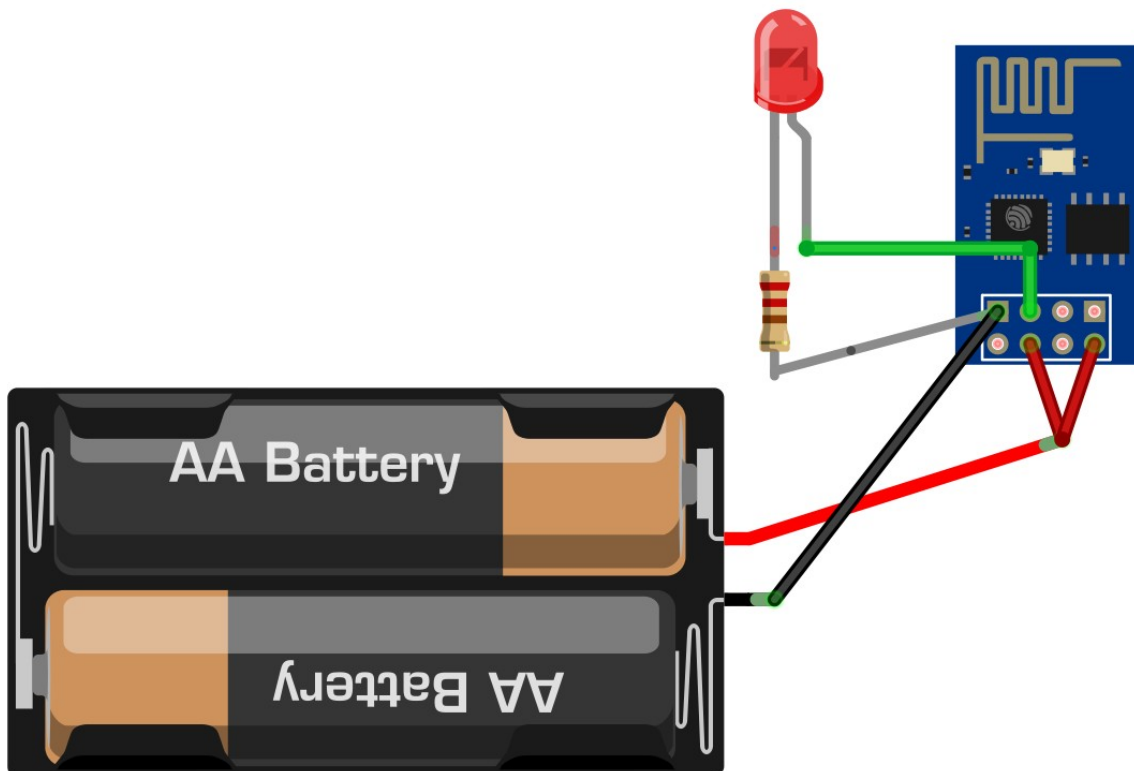


We are powering the circuit from 2x AAA batteries, but you could use 2x AA instead.

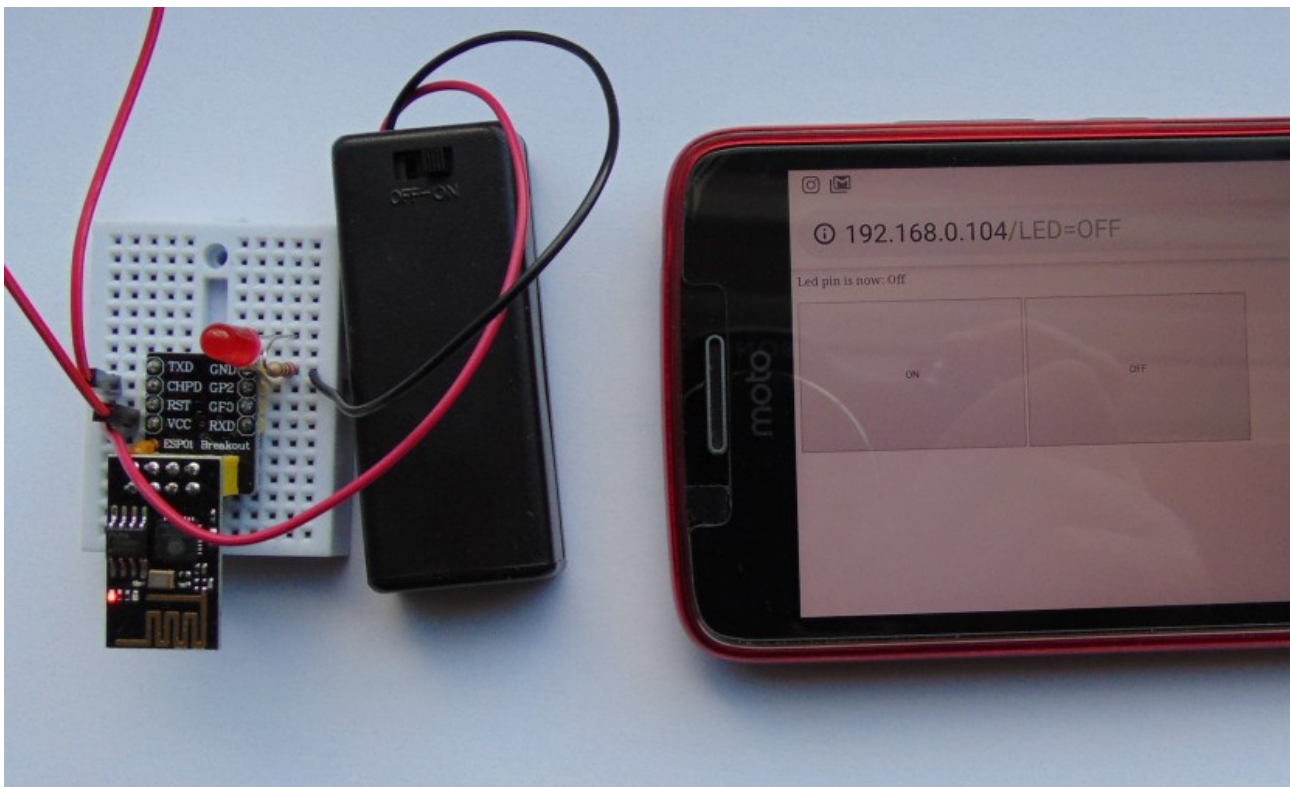
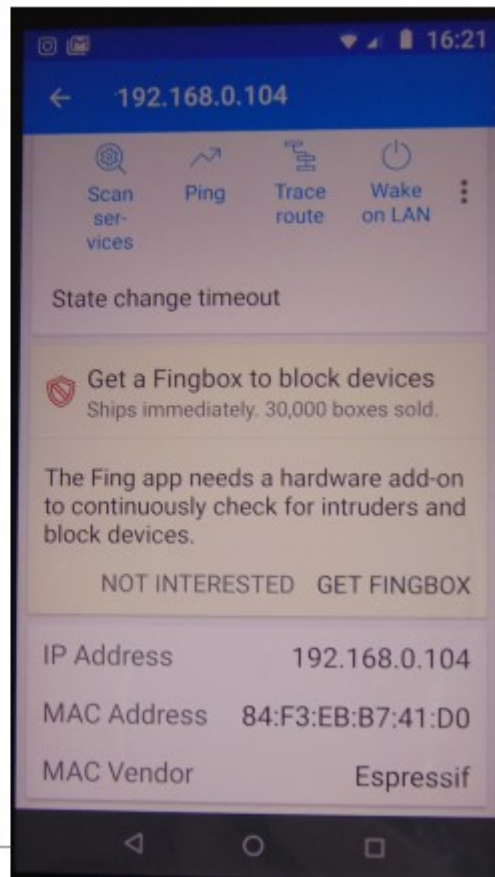
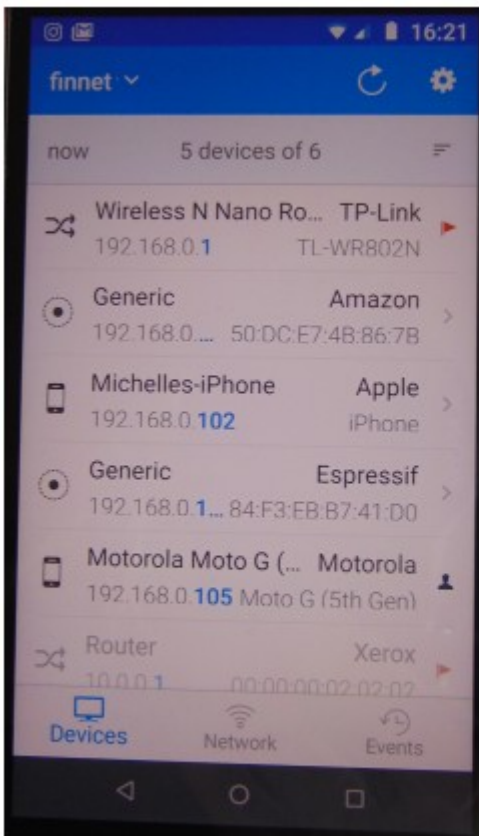
The photo doesn't show the wiring really clearly, so here's a circuit diagram:

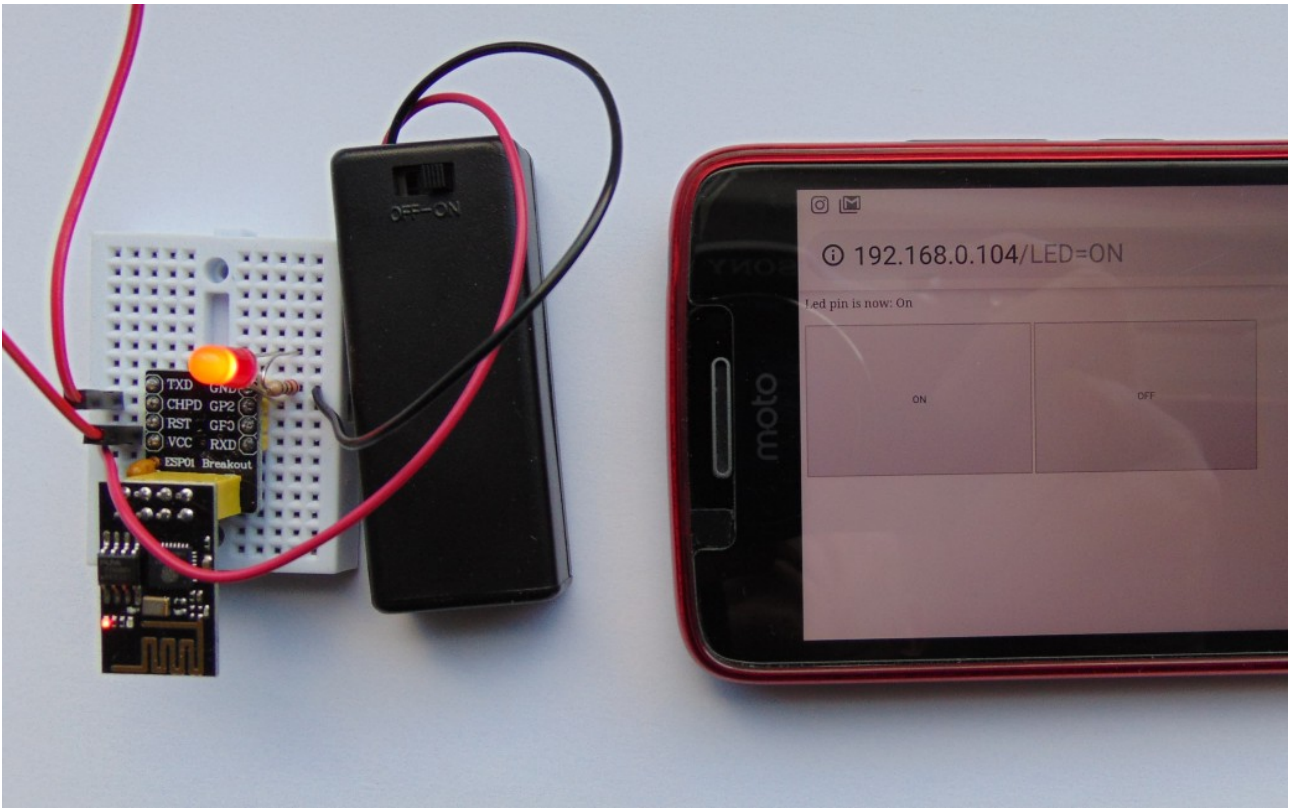


and here's how it connects if you're not using the breadboard adapter:

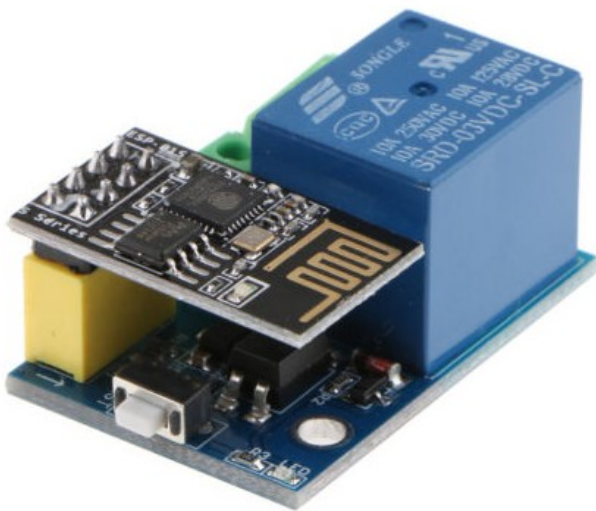


You can now go to another PC or mobile on your network and bring up the web page being generated over WiFi by the ESP-01. To find the IP address being used by the ESP-01 I used the Fing app on my Android phone, or you could log into your router and look at the list of attached devices.





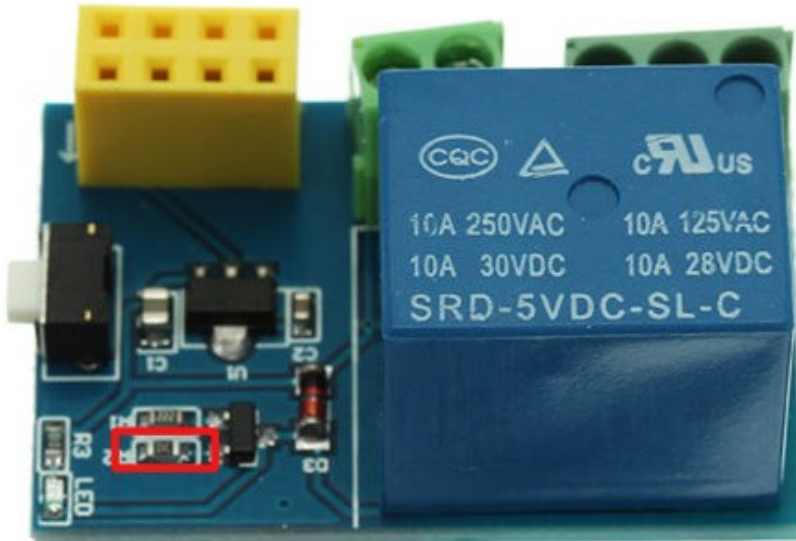
While learning how to turn an LED on and off is a useful exercise, it would be more useful if we could turn any electrical appliance on and off using the same software. Fortunately there's a relay board available for the ESP-01 that could be used to control any appliance.



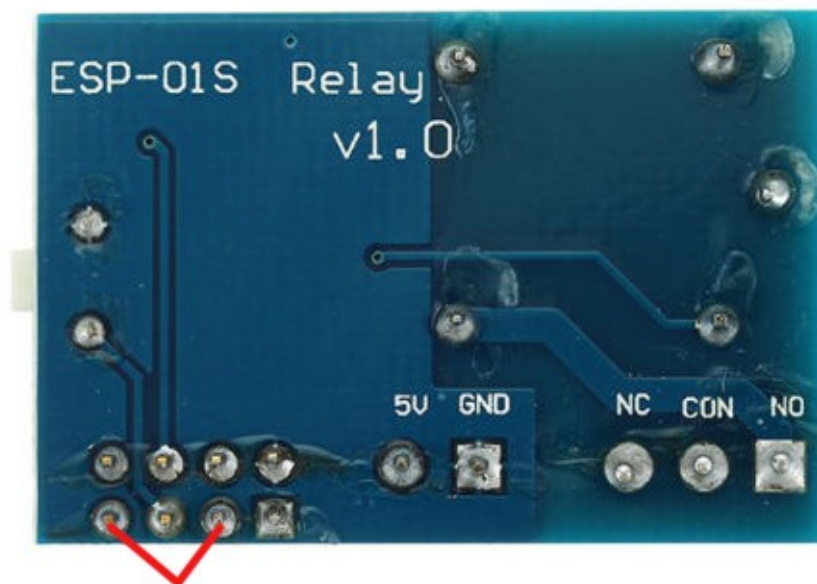
The relay boards operate on GPIO0 rather than GPIO2, so you'd need to edit the original Sketch and reprogram the board with the new Sketch. Then just attach 5 volts input to the relay board, which also powers the ESP-01 at 3 volts, and attach the positive rail of your device to the relay contacts CON and NO.

There are several versions of the relay board available, and the one in the photo above works fine, but there's also another relay board that is really cheap and is sometimes bundled with an ESP-01 that doesn't work without modification.

If your relay board looks like this one and it doesn't appear to work correctly without modification, you need to desolder the surface mount 10k resistor shown inside the red box.



And you also need to solder a wire link between these contacts on the back of the board



Once you've made those two changes, the board should work fine on GPIO0, if it didn't before.