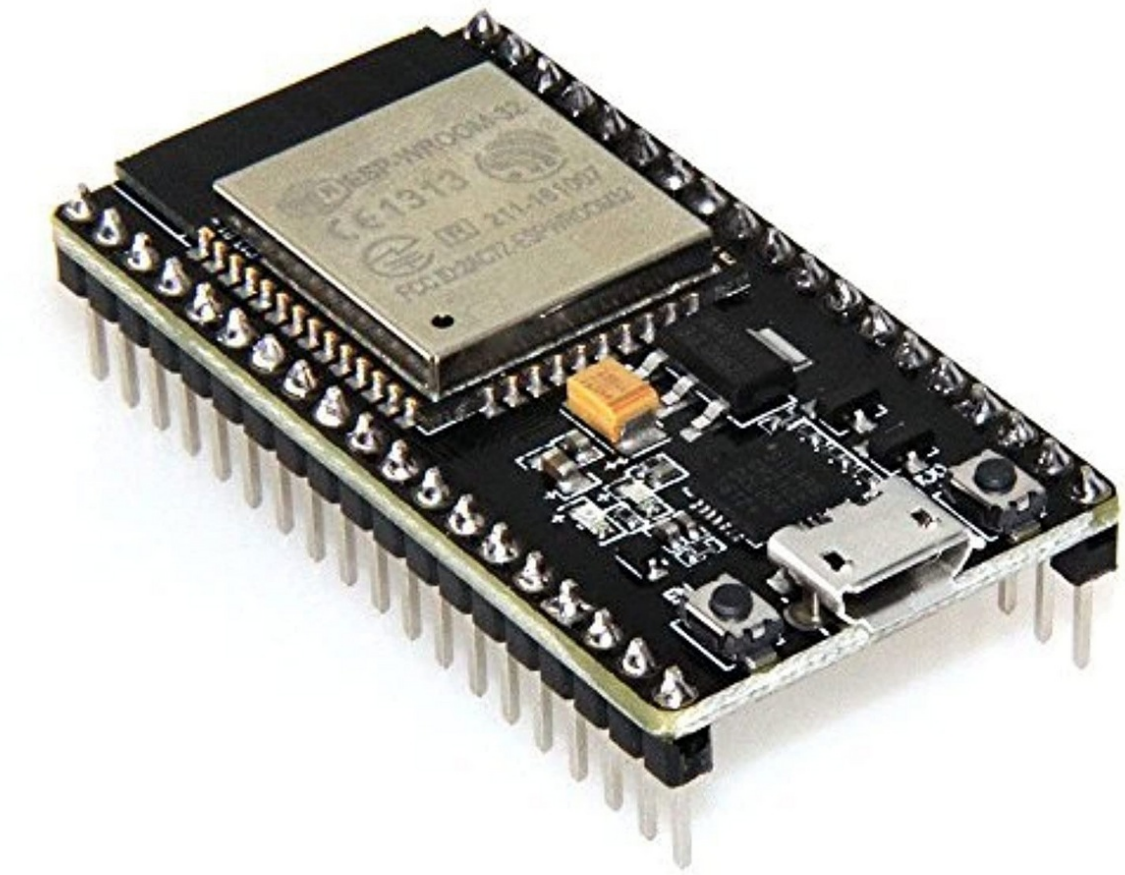
Bucknell UNIVERSITY

ESP32 HTTP Webserver

Written By: Liam Briening



INTRODUCTION

The ESP32 is a WiFi enabled micro-controller capable of serving HTTP webpages or hosting a REST API.

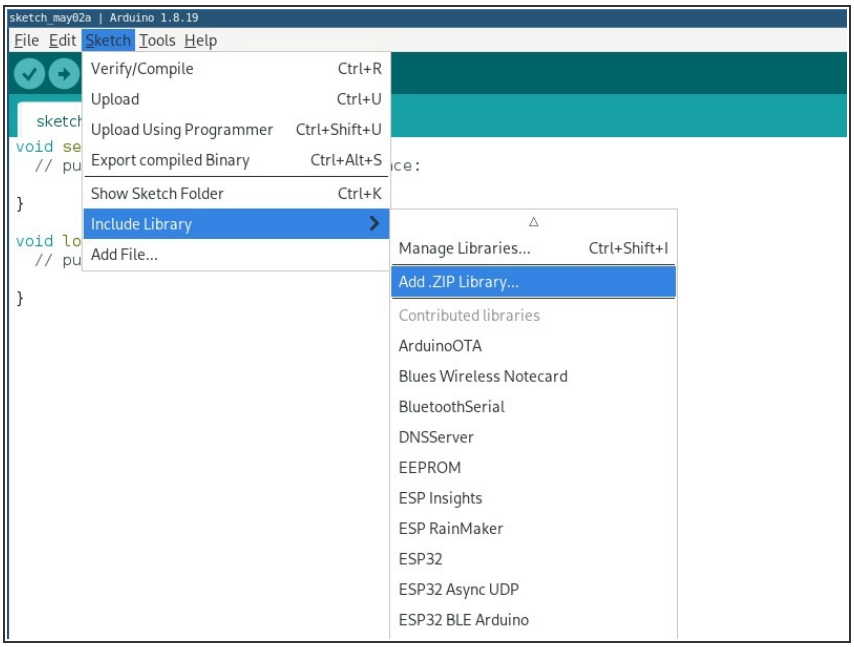
© 2023

Step 1 — Set up Arduino IDE



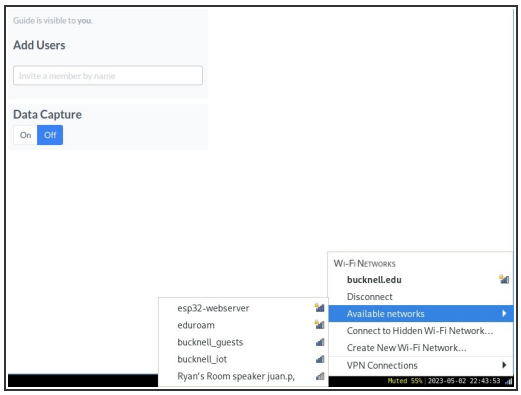
- Follow the steps here to install the Arduino IDE and configure it for the ESP-32
- Even if you already have the Arduino IDE installed you need to make sure the ESP-32 is added in the boards manager

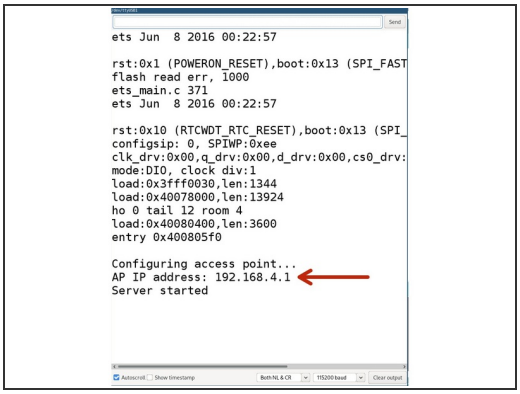
Step 2 — Install the ESP32 Async Web Server



- There is a synchronous web server also available for the ESP32 which is very similar in usage. I would recommend using the Asynchronous webserver because it allows multiple connections simmultaneously.
- Download the latest version of the ESP32 Async Web Server library from here.
- Install the library from the menu. Sketch > Include Library > Add .ZIP library
- The Web Server library also depends on the AsyncTCP library. Download it here and install it the same way.

Step 3 — Download and Run Template







- Download the template from Github here.
- Upload the code to the ESP32 and connect over the Serial Monitor to get the IP address.
- Connect to the ESP32's WiFi network and go to the IP address. You should be greeted with a welcome message.

Step 4 — Modify the Template



```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>

// Set these to your desired credentials.
const char *ssid = "esp32-webserver";
const char *password = "password";

AsyncWebServer server(80);

void handle_root(AsyncWebServerRequest *request) {
  request->send(200, "text/plain", "Hello from ESP32!");
}

void not_found(AsyncWebServerRequest *request) {
  request->send(404, "text/plain", "Not found");
}

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println("Configuring access point...");

  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);

  server.on("/", handle_root);
  server.onNotFound(not_found);

  server.begin();
  Serial.println("Server started");
}

void loop() {}
```



```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>

// Set these to your desired credentials.
const char *ssid = "esp32-webserver";
const char *password = "password";

AsyncWebServer server(80);

void handle_root(AsyncWebServerRequest *request) {
  request->send(200, "text/plain", "Hello from ESP32!");
}

void not_found(AsyncWebServerRequest *request) {
  request->send(404, "text/plain", "Not found");
}

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println("Configuring access point...");

  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);

  server.on("/", handle_root);
  server.onNotFound(not_found);

  server.begin();
  Serial.println("Server started");
}

void loop() {}
```

- The library use callback functions to handle different URLs. In the callback function plain text, json, or an HTML page can be sent as a response.
- These callback functions are bound to a URL and called each time something navigates to the URL.
- To read more about the library and see more examples you can look at the GitHub page for the library here.