

# ESP32 DHT22 Web Server Monitoring

This project provides a complete breakdown of monitoring 2 DHT22 Temperature and Humidity sensors through esp32 web server, including a full component list with estimated pricing, curated shop links with QR Codes ([links/prices are for the Philippine market](#)), and source code. I've included detailed circuit diagrams to ensure a smooth assembly process.

I am a student developer taking Computer Science at Bicol University working on many side projects involving **IoT, Game Development, and Android Applications**. You can find more of my hardware experiments and software projects on [my Github Profile](#).

If you notice any bugs in the code or improvements for the circuit design, I'd love to hear from you. Please reach out via email at [josefurei2019@gmail.com](mailto:josefurei2019@gmail.com).

## Table Of Contents

Component List, Prices, Shop Links & QR Code.....	2
Circuit Diagram.....	3
Board Manager & Libraries .....	4
Code.....	5-6
Deploying .....	7

## Build Components, Prices & Shop Links:

### Microcontroller:

For the Microcontroller, i've tested 2 products from different stores and they both worked as expected, however I used ENG LAB's ESP32 since it's cheaper and they come with a plastic case and a heatsink.

- [Makerlab's ESP32](#) ..... ₱385
- [ENG LAB.ph's ESP32](#) ..... ₱289 Type C, ₱279 Micro USB

Includes:

- Plastic Case
- Heatsink

Makerlab's ESP32



ENG LAB's ESP32



### DHT22 Temperature and Humidity Sensor

- [Makerlab's DHT22 Temperature & Humidity Sensor](#)

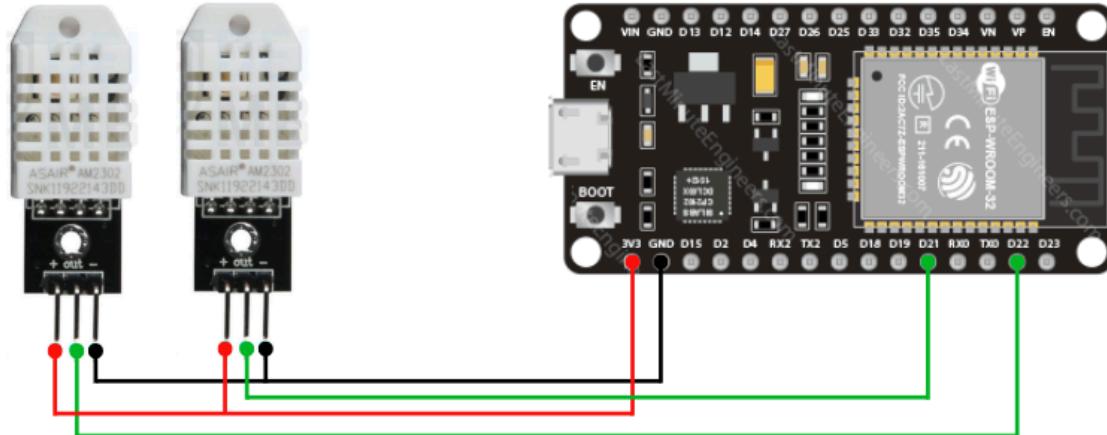


**TOTAL PRICE: ₱729**

### NOTE:

Prices may have changed at the time (February 2026) after this document was written.

# Circuit Diagram



## Pin Connections

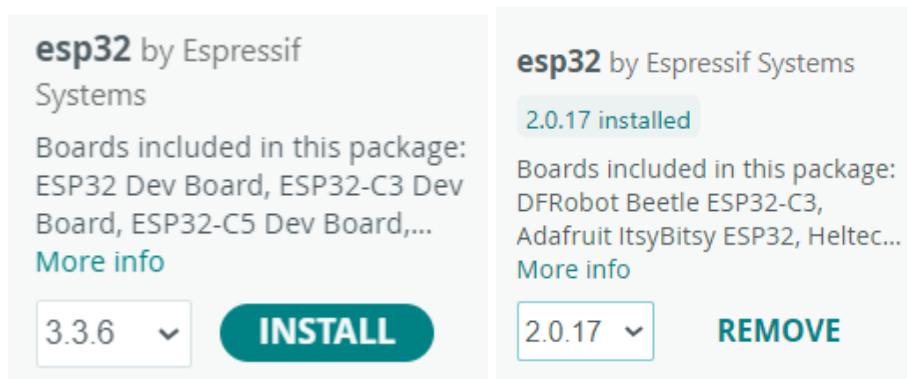
DHT22 (1)	ESP32
GND (-)	GND (-)
3v3 (+)	3v3 (+)
Signal	GPIO 21
DHT22 (2)	ESP32
GND (-)	GND (-)
3v3	3v3 (+)
Signal	GPIO 22

## Power Note:

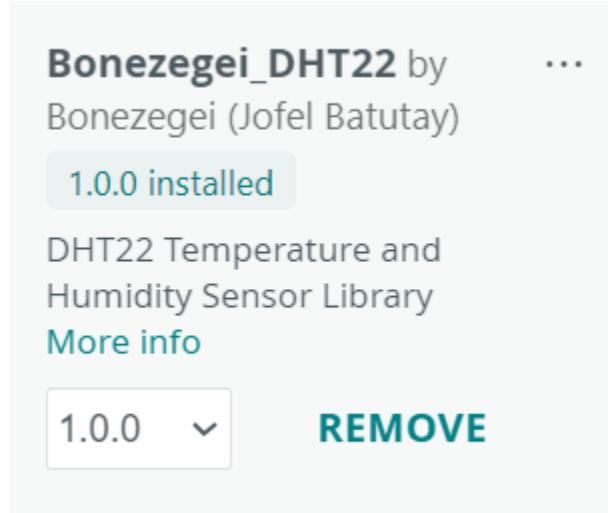
In this setup I am using a powerbank to power the ESP32 while the ESP32 is powering both DHT22's, if you are looking for a different setup, please be careful connecting anything to the Vin pin of ESP32 as it might get fried instantly if the voltage is too high or too low.

# Board Manager & Libraries (Arduino IDE)

1. Start Arduino IDE and go to File > Preferences at Additional boards manager URL paste this  
[https://raw.githubusercontent.com/espressif/arduino-esp32/master/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/master/package_esp32_index.json)
2. Install esp32 by Espressif Systems in Arduino IDE boards manager. The latest version (as of writing this document) does not work on my system, however the version 2.0.17 works fine.



3. Install Bonezegei\_DHT22 library at arduino IDE's Library manager. You can also try other libraries, I'm sure they all work just as fine if not even better.



## NOTE:

There might be version changes at the time (February 2026) of writing this document.

## Code

The source code for this project and many more tutorials are hosted on [my Github Profile](#) for version control and accessibility.

```
#include <Bonezegei_DHT22.h>
#include <WiFi.h>
#include <WebServer.h>

Bonezegei_DHT22 dht1(21);
Bonezegei_DHT22 dht2(22);

// WiFi Configuration
const char* ssid = "DHT22_Access_Point";
const char* password = "password123";

WebServer server(80);

// Global variables to store the latest data
float g_temp1, g_temp2;
int g_humi1, g_humi2;
unsigned long lastSensorRead = 0;

// This html will be served to the user as they access the IP
Address when connected to the Access Point
void handleRoot() {
    String html = R"=====(
        <!DOCTYPE html>
        <html>
        <head>
            <meta name='viewport' content='width=device-width,
initial-scale=1.0'>
            <meta http-equiv='refresh' content='5'>
            <style>
                body { font-family:Arial; text-align:center;
background:#1a1a1a; color:#eee; }
                .card { background:#2d2d2d; margin:20px auto;
padding:20px; width:80%; max-width:300px; border-radius:10px;
border: 1px solid #444; }
                .temp { color: #ffa500; font-size: 1.5rem; }
                .humi { color: #00bfff; font-size: 1.5rem; }
            </style>
        </head>
        <body>
            <h1>System Monitor</h1>
)=====;

    html += "<div class='card'><h2>Sensor #1</h2>";
    html += "<p class='temp'>" + String(g_temp1) + " °C</p>";
```

```

    html += "<p class='humi'>" + String(g_humi1) + "%</p></div>";

    html += "<div class='card'><h2>Sensor #2</h2>";
    html += "<p class='temp'>" + String(g_temp2) + " °C</p>";
    html += "<p class='humi'>" + String(g_humi2) + "%</p></div>";

    html += "</body></html>";
    server.send(200, "text/html", html);
}

void setup() {
    Serial.begin(115200);
    dht1.begin();
    dht2.begin();

    // Set up Access Point
    Serial.println("Configuring Access Point...");
    WiFi.softAP(ssid, password);

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

    // Define Server Routes
    server.on("/", handleRoot);

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

void loop() {
    server.handleClient();

    // Read sensors every 2 seconds independently of the web
    // server
    if (millis() - lastSensorRead > 2000) {
        if (dht1.getData()) {
            g_temp1 = dht1.getTemperature();
            g_humi1 = dht1.getHumidity();
        }
        if (dht2.getData()) {
            g_temp2 = dht2.getTemperature();
            g_humi2 = dht2.getHumidity();
        }
        lastSensorRead = millis();
    }
}

```

## Deploying

1. Connect the ESP32 to a powerbank or your laptop, open you phone's WiFi and look for "DHT22\_Access\_Point", connect to it using the password "password123" then go to any browser and type in 192.168.4.1 it should post the html with the sensor data.

You can change the access point SSID and password through the code at the WiFi configuration line 9 & 10.