

Smart Weather Station with ESP32 and Sensors

Διανεμημένη Παραγωγή

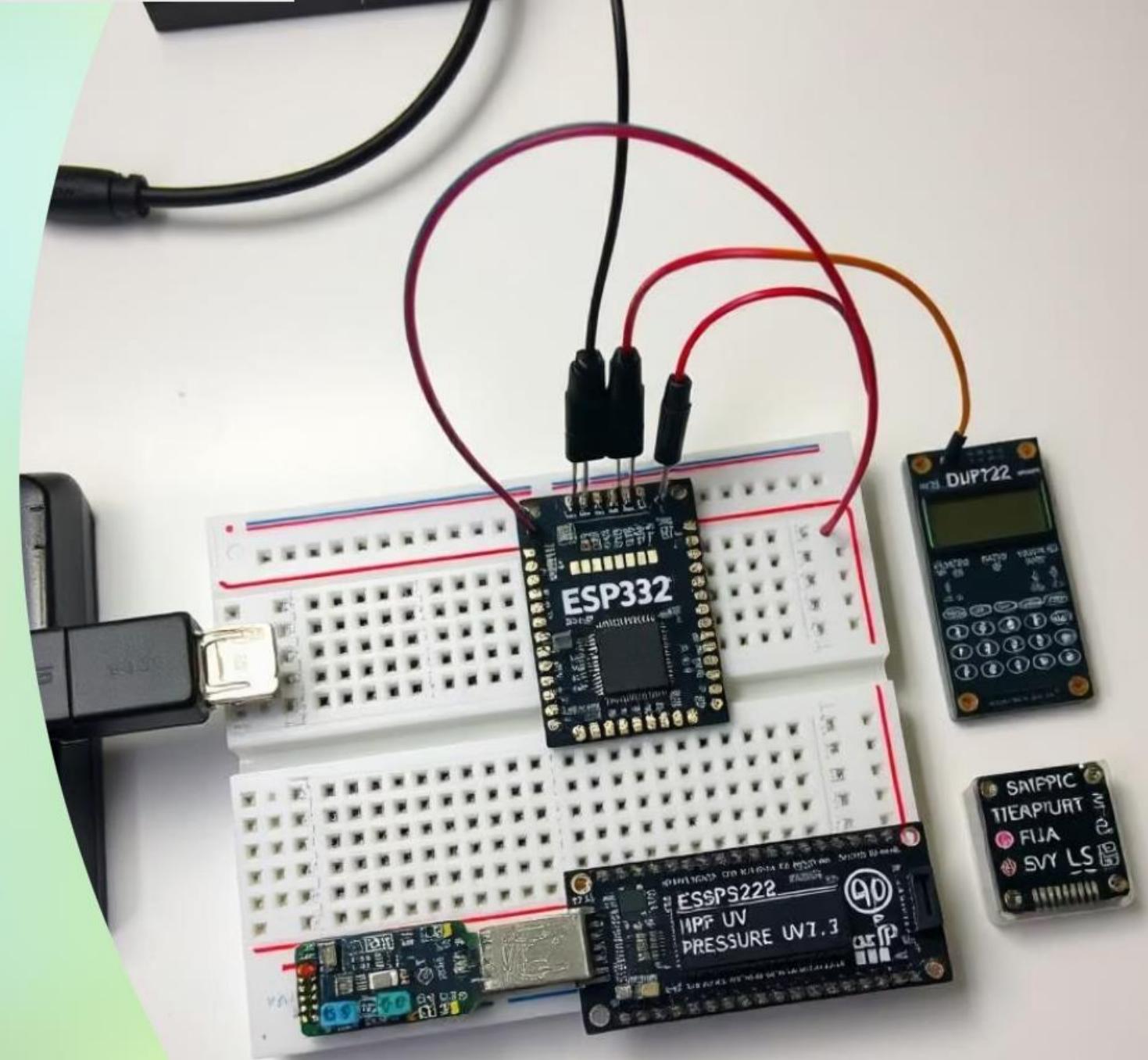
Εαρινό Εξάμηνο 2024-2025

Μαμουγιώργη Μαρία 10533

Ξακουστού Αιμιλία 10324

mmamougi@ece.auth.gr

xakoustou@ece.auth.gr



Υλικά & Εργαλεία



ESP32

- Κεντρική μονάδα επεξεργασίας με ενσωματωμένο Wi-Fi.
- Λειτουργεί ως web server για ασύρματη αποστολή δεδομένων.



Sensors

- DHT22:** Μετρά θερμοκρασία και υγρασία.
- UV Sensor:** Ανιχνεύει επίπεδα υπεριώδους ακτινοβολίας.
- BMP280:** Μετρά ατμοσφαιρική πίεση με ακρίβεια.



ARDUINO IDE

Ολοκληρωμένο περιβάλλον ανάπτυξης για προγραμματισμό και upload κώδικα.



Server Sent Events (SSE)

Τεχνολογία για δυναμική, πραγματικού χρόνου, μονής κατεύθυνσης ενημέρωση δεδομένων.



Static Website

Διασύνδεση χρήστη για την προβολή των μετρήσεων σε πραγματικό χρόνο. Ο χρήστης συνδέεται στο ESP32 μέσω Wi-Fi.



Arduino UNO

Χρήση της πλακέτας για έλεγχο των αισθητήρων και δοκιμή του αντίστοιχου κώδικα.



Extra Components

- Breadboard
- Καλώδια και αντίσταση 10KΩ
- Powerbank
- Πολύμετρο

Λειτουργία του συστήματος



Αρχικοποίηση ESP32

Συνδέεται στο Wi-Fi και εκκινεί web server.



Ανάγνωση Δεδομένων από Αισθητήρες

Ο ESP32 αποστέλλει δεδομένα μέσω SSE σε κάθε client που έχει συνδεθεί.



Αποστολή Δεδομένων μέσω Server-Sent Events (SSE)

Το ESP32 στέλνει τις τιμές αυτόμata στη σελίδα του client σε πραγματικό χρόνο.



Προβολή στη Στατική Ιστοσελίδα

Ο χρήστης βλέπει ενημερωμένες μετρήσεις χωρίς refresh, μέσω browser (τοπικό IP).

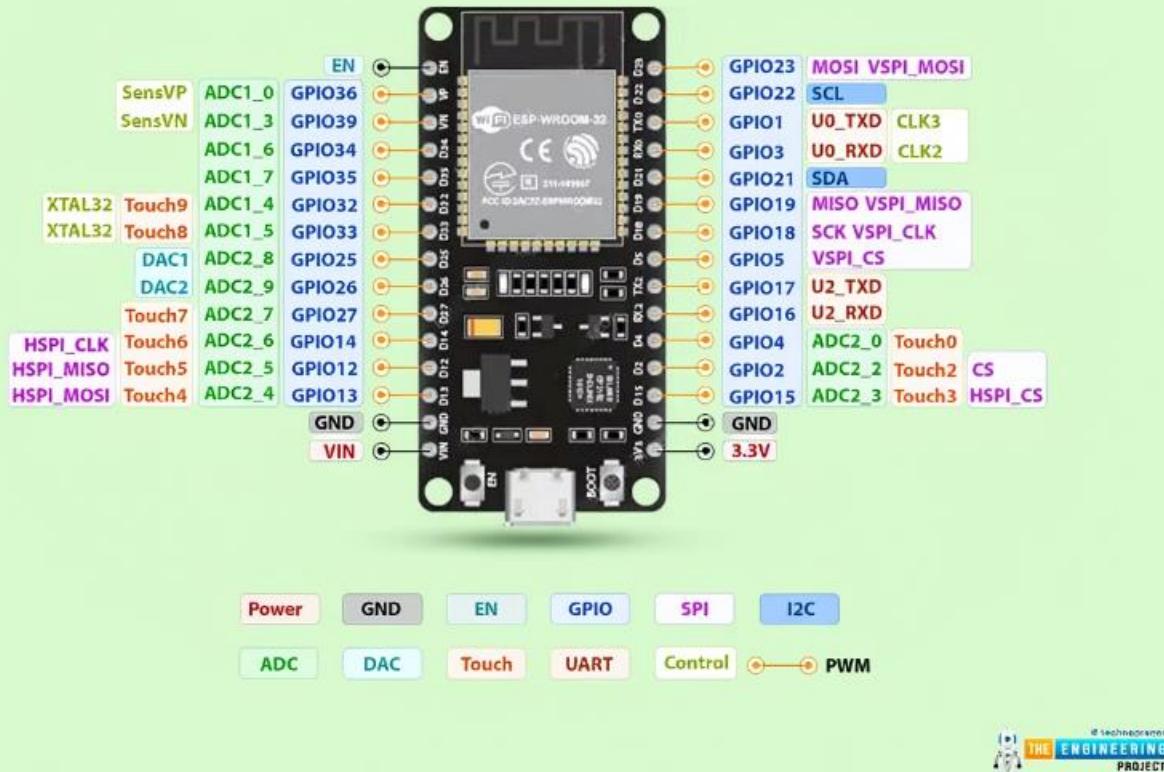


Άμεση Απόκριση

Δεν απαιτείται polling – εξοικονόμηση πόρων & άμεση απόκριση.

ESP32

ESP32 DEV. BOARD PINOUT



Χαρακτηριστικά

- Διπύρηνος επεξεργαστής (Dual-core) με υψηλή ταχύτητα.
 - Υποστηρίζει ευρεία γκάμα περιφερειακών εξαρτημάτων. (ADC, DAC, SPI, I2C, UART)
 - 34 GPIO θύρες
 - Ενσωματωμένο WiFi και Bluetooth
 - Σύνδεση σε WiFi
 - Επικοινωνία μέσω Bluetooth
 - Δυνατότητα λειτουργίας ως web server
 - Αποστολή δεδομένων σε ένα cloud
 - Χαμηλή κατανάλωση ενέργειας
 - Υποστήριξη από Arduino IDE

Server Sent Events (SSE)

Μονοκατευθυντική Επικοινωνία

Πρωτόκολλο HTTP

Ανανέωση Δεδομένων

Εξουκονόμηση Πόρων & Αμεση Απόκριση

Πως λειτουργεί:

- Ο browser στέλνει αίτημα σύνδεσης στον ESP32 (/events endpoint).
- Ο ESP32 στέλνει συνεχώς νέα δεδομένα όταν υπάρχουν αλλαγές.
- Ο browser εμφανίζει τα δεδομένα άμεσα χωρίς ανανέωση σελίδας.

Endpoint "/events"

Εξυπηρέτηση δεδομένων αισθητήρων.

Μορφή event του κάθε αισθητήρα

Κάθε event έχει:

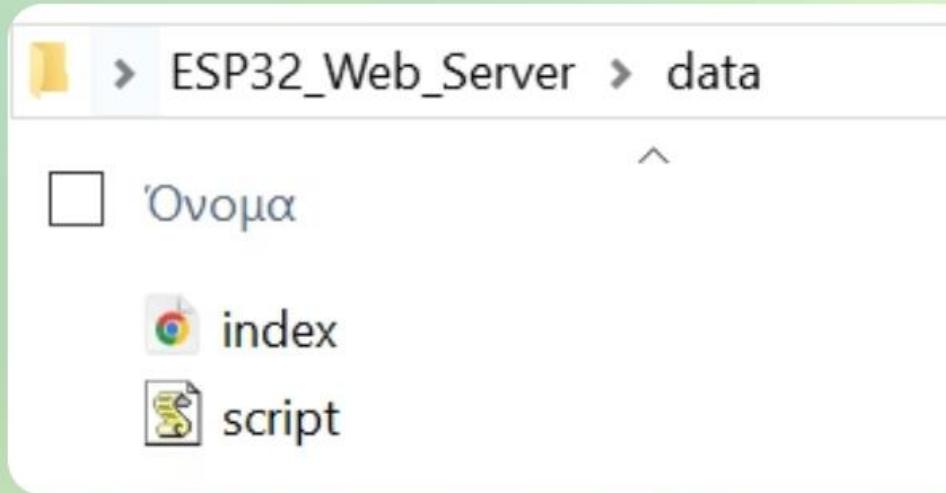
- `data`: την τιμή
- `event name`: π.χ. "UV"
- `ID`: μοναδικό timestamp (χρησιμοποιείται από τον browser)

Code

```
events.send("ping", NULL, millis());
events.send(String(temperature).c_str(), "temperature", millis());
events.send(String(humidity).c_str(), "humidity", millis());
events.send(String(pressure).c_str(), "pressure", millis());
events.send(String(UV).c_str(), "UV", millis());
```

LittleFS

- Το LittleFS.begin() κάνει mount το εσωτερικό file system του ESP32.
- Ο server εξυπηρετεί τα αρχεία της ιστοσελίδας (HTML/JS/CSS) από το LittleFS.
- Αν κάποιος ανοίξει την IP του ESP32 από browser, φορτώνεται αυτόματα το index.html.



Μορφή συστήματος αρχείων LittleFS

```
// Init LittleFS
if (!LittleFS.begin()) {
    Serial.println("LittleFS mount failed");
    return;
}

// Serve static files
server.serveStatic("/", LittleFS, "/").setDefaultFile("index.html");
```

Φόρτωση στατικής ιστοσελίδας - σύνδεση με το αρχείο html.

Dashboard & HTML/CSS/JavaScript



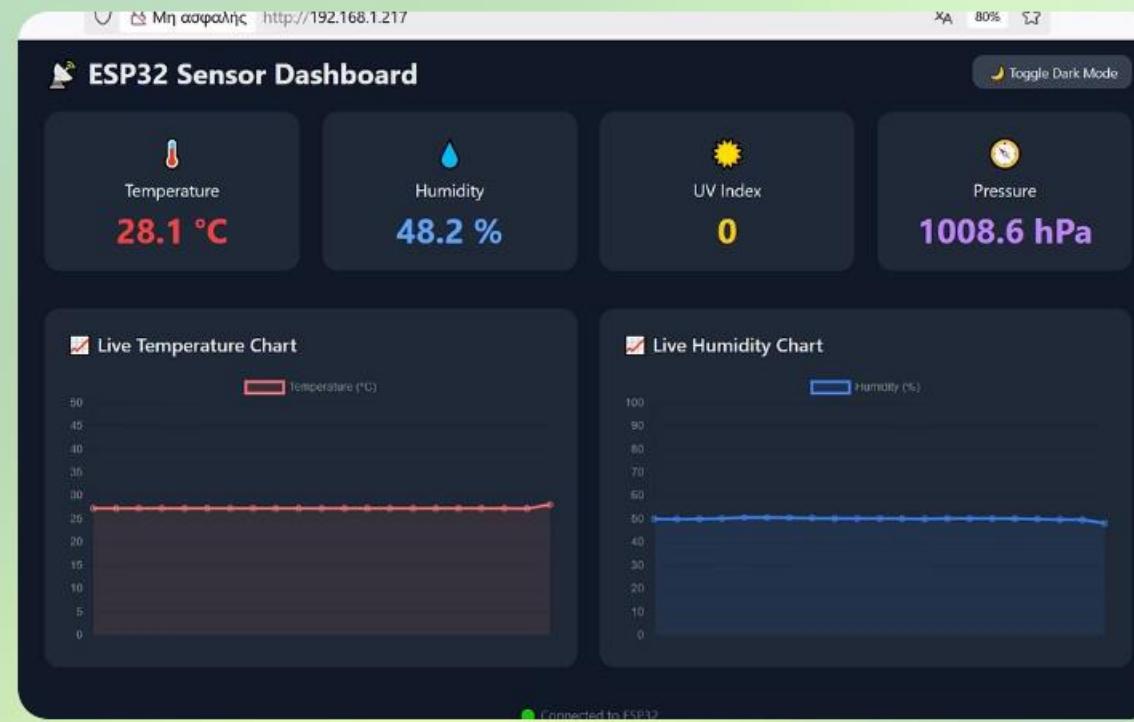
Χρήση EventSource για λήψη δεδομένων σε πραγματικό χρόνο



Υποστήριξη πολλαπλών events:
temperature, humidity, uv,
pressure



Dynamic ενημέρωση περιεχομένου
HTML και γραφημάτων Chart.js



Εικόνα του Dashboard με δεδομένα αισθητήρων και το αντίστοιχο JavaScript.

```
const source = new EventSource('/events');

source.addEventListener('temperature', function(e) {
  const temp = parseFloat(e.data);
  document.getElementById("temp").textContent = temp.toFixed(1)
  // Ενημέρωση γραφήματος
});
```

JavaScript EventSource

Αισθητήρας Θερμοκρασίας & Υγρασίας DHT22

Χαρακτηριστικά:

- Ψηφιακός αισθητήρας μέτρησης θερμοκρασίας και σχετικής υγρασίας
- Επικοινωνία με ESP32 μέσω ψηφιακού pin
 - Vin: 3.3V-6V
 - Ψηφιακή Έξοδος:** Αποστολή έτοιμων τιμών μέσω 1-wire πρωτοκόλλου
 - Εύρος Θερμοκρασίας:** -40°C έως +80°C
 - Ακρίβεια Θερμοκρασίας:** ±0.5°C
 - Εύρος Υγρασίας:** 0% – 100% RH
 - Ακρίβεια Υγρασίας:** ±2% – 5% RH

Τρόπος Λειτουργίας:

- Στέλνει διαδοχικά πακέτα δεδομένων με τιμές θερμοκρασίας (°C) και υγρασίας (%)
- Χρήση βιβλιοθήκης DHT.h
 - dht.readTemperature()
 - dht.readHumidity()

Κώδικας

```
#include <DHT.h> // Βιβλιοθήκη για DHT αισθητήρες

#define DHTPIN 4      // Ο αισθητήρας είναι συνδεδεμένος στο GPIO4
#define DHTTYPE DHT22 // Ορίζουμε τον τύπο του αισθητήρα

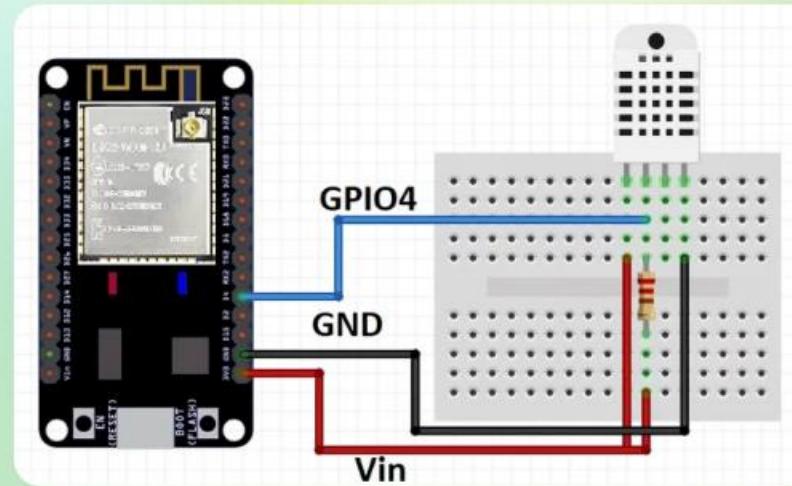
DHT dht(DHTPIN, DHTTYPE); // Δημιουργία αντικειμένου DHT

float temperature; // Μεταβλητή για θερμοκρασία
float humidity;    // Μεταβλητή για υγρασία

void getDHTReadings() {
    float temp = dht.readTemperature(); // Ανάγνωση θερμοκρασίας (°C)
    float hum = dht.readHumidity();     // Ανάγνωση υγρασίας (%)

    // Αν οι τιμές είναι έγκυρες (όχι NaN), τις αποθηκεύουμε
    if (!isnan(temp)) temperature = temp;
    if (!isnan(hum)) humidity = hum;
}
```

Συνδεσμολογία



→ Ο DHT22 χρησιμοποιεί μονής γραμμής επικοινωνία (**one-wire protocol**).

→ Μετά την αρχική παλμοσειρά από τον ESP32, η γραμμή δεδομένων γίνεται **floating** (αιωρούμενη) και χρειάζεται μια **σταθερή λογική κατάσταση** όταν δεν μεταδίδεται κάτι.

→ Η **pull-up** αντίσταση τραβάει τη γραμμή προς τα πάνω (λογικό HIGH), εξασφαλίζοντας ότι τα bits αναγνωρίζονται σωστά.

Αισθητήρας UV - SENS43

Χαρακτηριστικά:

- Αναλογικός αισθητήρας μέτρησης UV ακτινοβολίας
- Παρέχει αναλογική έξοδο τάσης ανάλογη με την ένταση UV
 - **Τάση λειτουργίας (Vin):** 2.7V – 5.5V (π.χ. 3.3V με ESP32)
 - **Εύρος φασματικής απόκρισης:** 240 nm – 370 nm (UVB & UVA)

Τρόπος Λειτουργίας:

- Το ESP32 διαβάζει την αναλογική τάση (ADC) και υπολογίζει τον UV δείκτη
- Ο αλγόριθμος κάνει προσαρμογή των τάσεων $3.3V \rightarrow 5V$ για ακρίβεια δηλαδή
 - **vout_5V = (vout_actual / 3.3V) * 5V**

Κώδικας

```
#define UVPIN 34 // Ο UV αισθητήρας είναι συνδεδεμένος στο GPIO34

int UV = 0; // UV Index

void getUVReading() {
    long sum = 0;

    // Συλλογή 1024 δειγμάτων με καθυστέρηση 2ms
    for (int i = 0; i < 1024; i++) {
        sum += analogRead(UVPIN); // Ανάγνωση από αναλογική είσοδο
        delay(2);
    }

    int avgSensorValue = sum >> 10; // Υπολογισμός μέσου όρου (sum / 1024)

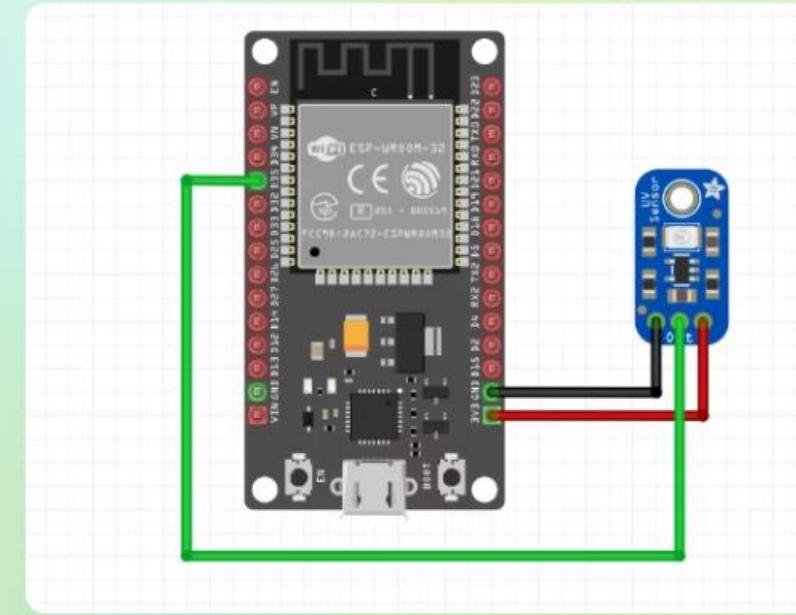
    // Υπολογισμός τάσης εξόδου σε millivolts (3.3V reference)
    float vout_actual = avgSensorValue * (3300.0 / 4095.0);

    // Αναγνώριση τιμής σε 5V, ώστε να αντιστοιχούν σωστά τα UV επίπεδα
    float vout_5V = (vout_actual / 3300.0) * 5000.0;
}
```

```
// Αντιστοίχιση σε UV Index σύμφωνα με επίσημες τιμές
if (vout_5V < 50) UV = 0;
else if (vout_5V < 227) UV = 1;
else if (vout_5V < 318) UV = 2;
else if (vout_5V < 408) UV = 3;
else if (vout_5V < 503) UV = 4;
else if (vout_5V < 606) UV = 5;
else if (vout_5V < 696) UV = 6;
else if (vout_5V < 795) UV = 7;
else if (vout_5V < 881) UV = 8;
else if (vout_5V < 976) UV = 9;
else if (vout_5V < 1079) UV = 10;
else UV = 11;

// Εκτύπωση για έλεγχο
Serial.printf("UV Voltage (3.3V): %.2f mV\n", vout_actual);
Serial.printf("Estimated Voltage (5V): %.2f mV\n", vout_5V);
```

Συνδεσμολογία



UV Index	0	UV INDEX 1	UV INDEX 2	UV INDEX 3	UV INDEX 4	UV INDEX 5
Vout(mV)	<50	227	318	408	503	606
UV Index	UV INDEX 6	UV INDEX 7	UV INDEX 8	UV INDEX 9	UV INDEX 10	UV INDEX 11+
Vout(mV)	696	795	881	976	1079	1170+

Αισθητήρας Πίεσης - BMP280

Χαρακτηριστικά:

- Ψηφιακός αισθητήρας πίεσης και θερμοκρασίας
- Χρησιμοποιεί I2C πρωτόκολλο επικοινωνίας
- Ατμοσφαιρική Πίεση:**
 - Εύρος: 300 – 1100 hPa (ή mbar)
 - Ακρίβεια: ± 1 hPa
 - Ανάλυση: 0.18 Pa (για υψομετρική ακρίβεια ~ 1.5 μέτρο)
 - Vin: 3.3V–5V

Τρόπος Λειτουργίας:

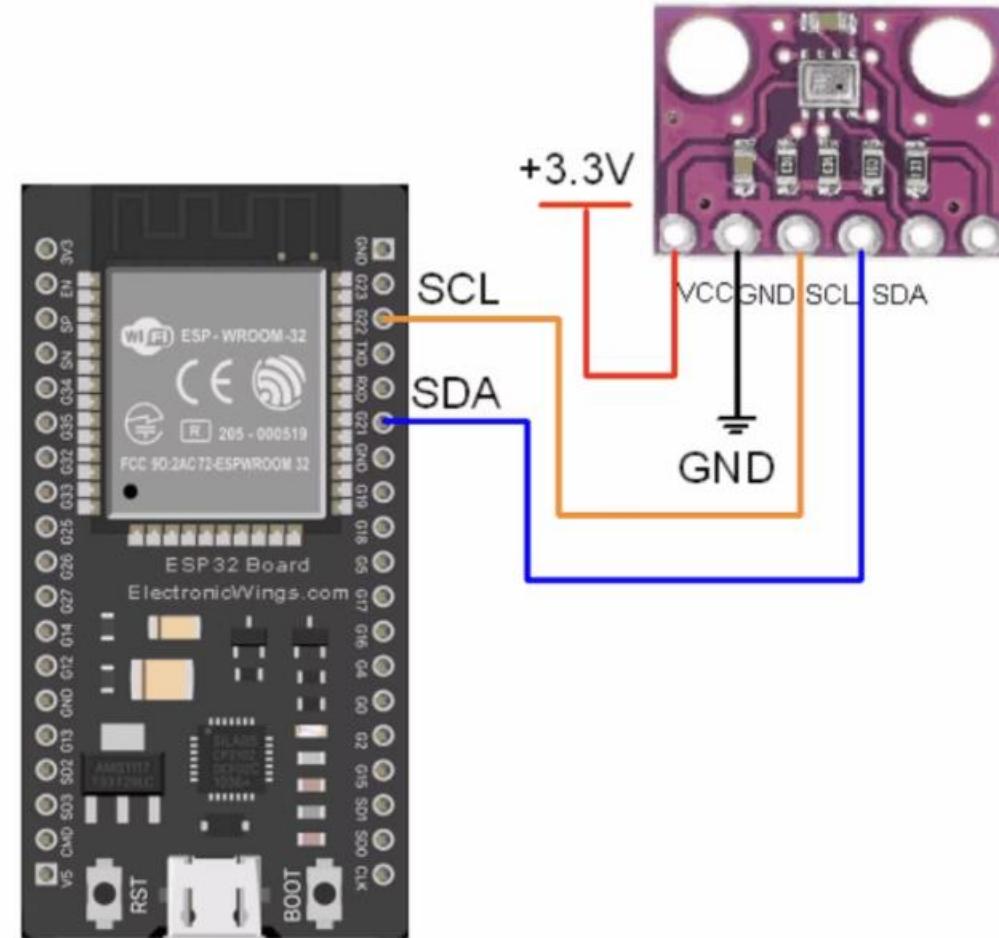
- Μετράει την πίεση του αέρα σε Pascal και μετατρέπεται σε hPa
- Ενσωμάτωση μέσω της βιβλιοθήκης: Adafruit_BMP280.h
 - `bmp.readPressure()`

Κώδικας

```
//Init BMP280
void initBMP() {
    // Using ESP32's default I2C pins for BMP280 SDA and SCL
    if (!bmp.begin(0x76)) { // use 0x76 or 0x77 depending on your module
        Serial.println(F("Could not find BMP280 sensor at 0x76!"));
        while (1); // Halt if sensor not found
    }

void getBMPReading(){
    float pres = bmp.readPressure();
    if (!isnan(pres)) {
        pressure = pres / 100.0; // convert Pa to hPa
    }
}
```

Συνδεσμολογία



```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi ..");
while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
}
Serial.println(WiFi.localIP());
```



```
// For HTML template
String processor(const String& var){
    getSensorReadings();

    //Serial.println(var);
    if(var == "TEMPERATURE"){
        return String(temperature);
    }
    else if(var == "HUMIDITY"){
        return String(humidity);
    }
    else if(var == "PRESSURE"){
        return String(pressure);
    }
    else if (var == "UV") {
        return String(UV);
    }
    return String();
```

Κώδικας Arduino (ESP32)

Web Server

Δημιουργία με ESPAsyncWebServer.

Handlers

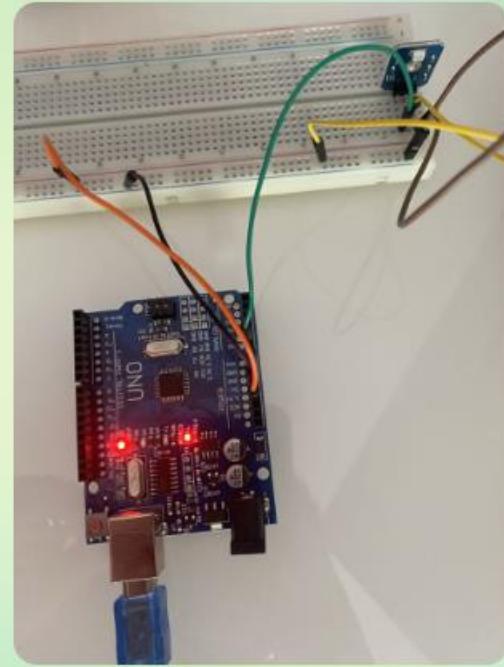
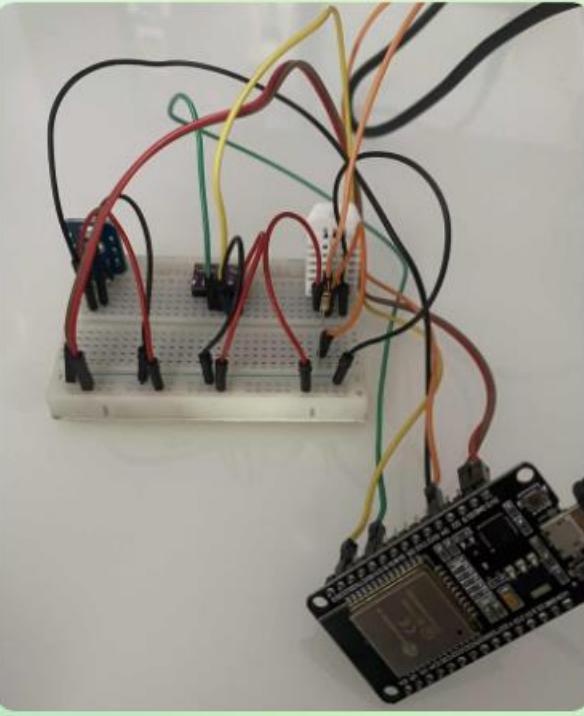
Ορισμός για "/" (dashboard) και "/events" (δεδομένα αισθητήρων)

Ανάγνωση Αισθητήρων

Κάθε 5 δευτερόλεπτα.

Αποστολή Δεδομένων

Μέσω SSE σε clients.



Συνολικά

1

Δευτερόλεπτο Χρόνος Απόκρισης

5

Δευτερόλεπτα Ανανέωσης Δεδομένων

3

Αισθητήρες

Βίντεο Επίδειξης

Google Docs

Smart Weather Station Video - Δημιουργήθηκε με το Clipchamp 1749331660860.m...



Μελλοντικά Βήματα & Επεκτάσεις

Επέκταση Αισθητήρων

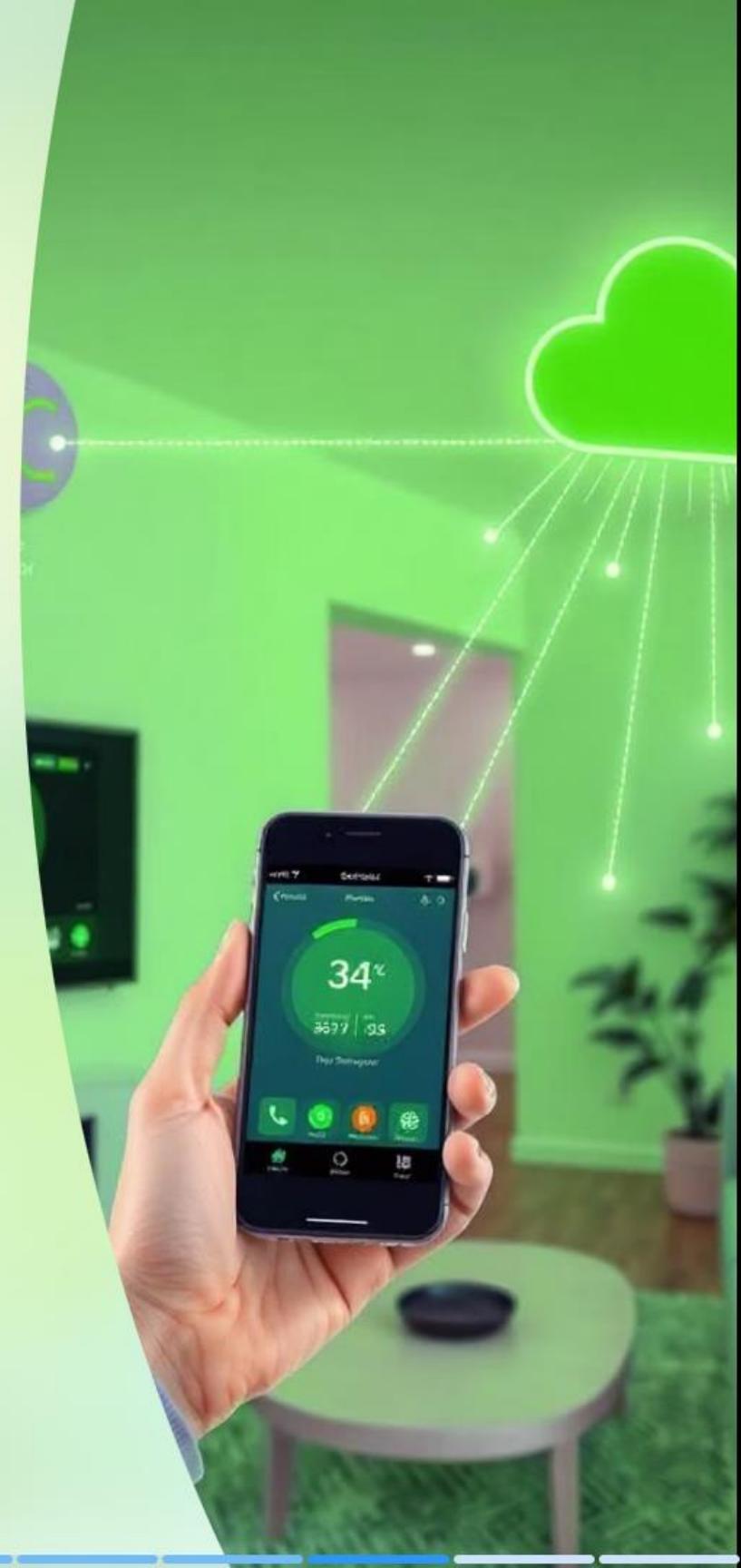
- Εμπλουτισμός δεδομένων.
- Αυτόματη ενεργοποίηση συσκευών με βάση αισθητήρες (π.χ. ανεμιστήρας)

Ενσωμάτωση Cloud

- Αποθήκευση δεδομένων σε πλατφόρμες cloud.
- Ενισχυμένη ανάλυση.
- Αρχείο.

Mobile App

- Εύκολη πρόσβαση στα δεδομένα.
- Ειδοποίησεις σε πραγματικό χρόνο.



Βιβλιογραφία

- <https://randomnerdtutorials.com/esp32-web-server-sent-events-sse/>
- <https://www.instructables.com/ESP32-HTTP-Web-Server-HTML-CSS-Simple-Counter-As-t/>
- <https://www.sunrom.com/p/ultraviolet-uv-light-radiation-sensor>
- https://abra-electronics.com/sensors/sensors-light-imaging-en/sens-43uv-analog-ultra-violet-light-sensor-module-for-5v-microcontrollers.html?utm_source=chatgpt.com
- https://www.waveshare.com/wiki/UV_Sensor
- <https://wokwi.com/esp32>
- <https://docs.cirkitdesigner.com/component/15a31805-192d-4969-9d97-ebc0787e8274/uv-sensor>
- <https://www.electronics-lab.com/project/arduino-uv-meter-using-uv30a-ultraviolet-sensor/>

Ευχαριστούμε Πολύ!

Ερωτήσεις;

