

ETH - Enviromental Telemetry Hub

Introducere

Proiectul ETH (Environmental Telemetry Hub) reprezinta un sistem IoT (Internet of Things) pentru monitorizarea conditiilor de mediu dintr-o incapere.

Sistemul colecteaza date in timp real despre temperatura, umiditatea aerului si nivelul de lumina ambientala, pe care le trimite atat la un dashboard online precum si pe un ecran local, declansand alarme vizuale si sonore cand parametrii ies din valorile optime.

Scopul principal este mentinerea unui mediu de lucru sau de locuit sanatos si confortabil, sau mentinerea unui mediu pretentios (plante, echipamente electronice, muzee) in parametri optimi, informand utilizatorul cand trebuie sa ia masuri (ex. sa aeriseasca o camera, sa porneasca caldura sau sa ajusteze lumina ambientala).

Descriere generala



Proiectul functioneaza pe baza unui microcontroller ESP32 care actioneaza ca unitate centrala, care colecteaza date de la senzori si actioneaza diversi actuatori.

Schema bloc a sistemului:

- **Modul Central:** Microcontroller ESP32.
- **Module de Intrare (Input):**
 - **DHT11:** Modul digital care transmite catre ESP32 valorile masurate pentru temperatura mediului ambiant si umiditatea relativa.
 - **Fotorezistor (LDR):** Folosit intr-un divizor de tensiune, ofera o tensiune variabila citita de modulul ADC al ESP32-ului, care determina intensitatea luminoasa.
- **Module de Iesire (Output):**
 - **Infrastructura Online:** Datele colectate sunt trimise catre un VPS aflat online, care permite vizualizarea istoricului acestora.
 - **Broker MQTT** - Mosquitto - Colecteaza datele trimise de sistem.
 - **Baza de date** - InfluxDB v2 - Stocheaza datele colectate
 - **Dashboard** - Grafana - Permite afisarea datelor colectate
 - **Display OLED SSD1306:** Ecran ce primeste comenzi prin protocol I2C pentru a afisa parametrii exacti masurati si mici elemente de interfata.
 - **LED RGB:** Indicator de stare comandat prin semnale PWM. Semnaleaza starea generala prin diverse culori si pulsatii.
 - **Buzzer:** Un modul audio actionat de un pin digital pentru a emite semnale acustice de avertizare

in caz de parametrii critici.

Elemente de originalitate si complexitate

Desi monitorizarea ambientală este un concept comun, acest proiect se diferentiaza prin rigoarea implementarii tehnice:

- **Procesare avansata a semnalelor analogice:** Pentru citirea luminozitatii, s-a implementat o filtrare a zgomotului prin medierea a 64 de esantioane intr-o fereastră de 20ms, urmata de un clipping la intervalul de liniaritate al ADC controller-ului de pe ESP32.
- **Arhitectura software bazata pe RTOS:** Trecerea de la o bucla de executie secventiala la task-uri FreeRTOS asigura ca citirea senzorilor cu timing strict (DHT11) nu este blocata de latentele retelei (MQTT).
- **Sistem hibrid de alertare:** Sistemul functioneaza independent; ofera alarme locale (Buzzer/RGB) in timp real, concomitent cu persistenta datelor in infrastructura de Cloud containerizata (InfluxDB + Grafana).

Hardware Design



Qty	Price (RON)	Item	Link
1	60.35	ESP32 WROOM 32D	Link
1	2.13	Modul Led RGB	Link
1	2.99	Buzzer Pasiv	Link
1	18.98	SSD1306	Link
1	6.99	DHT11	Link
1	1.49	Fotorezistor (tip 5528)	Link
1	0.1	Rezistor 10k	Link
1	13.99	Breadboard 830	Link
1	9.99	Set Fire Dupont tata-tata 10cm	Link
1	6.99	Set Fire Dupont tata-mama 20cm	Link
	124.00		

Componenta	Pin Componenta	Net Label	Pin ESP32	Funcție
Ecran OLED	SDA	I2C_SDA	IO21	Date I2C
Ecran OLED	SCL	I2C_SCL	IO22	Ceas I2C
Senzor DHT11	DATA	DHT_DATA	IO4	Date digitale Temp/Umiditate
LED RGB	R (Rosu)	LED_R	IO25	Control PWM Rosu
LED RGB	G (Verde)	LED_G	IO26	Control PWM Verde
LED RGB	B (Albastru)	LED_B	IO27	Control PWM Albastru
Senzor LDR	Midpoint	LDR	IO34	Intrare Analogica (ADC)
Buzzer	S (Semnal)	BUZZER	IO33	iesire PWM

Software Design

Firmware-ul va fi dezvoltat in limbajul C, folosind framework-ul oficial ESP-IDF prin PlatformIO, permitand utilizarea conceptelor low-level invatate la laboratoare. Pentru implementarea multitasking-ului (ecran, senzori, networking) este folosit FreeRTOS.

DHT11

Senzorul DHT11 este un senzor ieftin, consacrat in industrie si de incredere. Acesta foloseste un protocol proprietar single wire pentru transmisia datelor, motiv pentru care am folosit o biblioteca (dht.c) pentru citirea valorilor.

Pentru calibrare am comparat rezultatele obtinute de acesta cu cele ale unui termometru digital, si am constatat ca se incadreaza in parametrii normali de eroare (umiditate $\pm 5\%$, temperatura $\pm 2^\circ\text{C}$)

Senzor GL5528

Senzorul fotorezistiv este conectat la o sursa de 3.3V, avand o rezistenta fixa de 10k Ohmi conectata la masa (GND). Acest circuit formeaza un divizor de tensiune, iar nivelul de tensiune este masurat de convertorul analog-digital (ADC) al placii ESP32.

1. Calculul Rezistentei LDR
Rezistenta curenta a senzorului (R_{ldr}) se calculeaza din valoare bruta ADC astfel:

$$R_{ldr} = R_{fixed} * ((4095 / ADC_{Raw}) - 1)$$

2. Calculul Iluminarii (Lux)

Relatia dintre rezistenta LDR si nivelul de lumina este log-liniara. Formula standard pentru conversie este:

$$Lux = 10 * (R_{10} / R_{ldr}) ^ (1 / Gamma)$$

Deoarece datasheet-ul ofera un interval destul de larg pentru valoare R_{10} (8k - 20k), am decis sa calibrez valoarea prin comparatie cu un senzor de referinta (al telefonului). Astfel am plasat cei doi senzori cat mai apropiati si paraleli, si am colectat date.

In urma calibrarii am obtinut ca senzorul meu are o valoare R_{10} de 17k Ohm, iar Gamma este conform specificatiei de 0.7.

Deoarece ADC-ul intern al controller-ului ESP32 se comporta neliniar la capete, am decis sa fac clip la valorile acestuia intr-un interval care sa excluda capetele (<100mV, >3100mV).

Pentru anulara zgomotului implementat luarea mediei a 64 de esantioane luate la interval de 312 microsecunde (20ms in total), intr-o sectiune critica de cod.

Buzzer pasiv

Un buzzer pasiv este folosit pentru a semnala situatii de urgenta (temperatura, umiditate iesite din parametrii de siguranta) folosind un semnal PWM.

Led RGB

Un LED RGB este folosit pentru a semnala situatia curenta (parametrii optimi sau nu).

Ecran OLED 0.96 SSD1306

Ecranul OLED de 0.96 inch echipat cu controller-ul SSD1306 este un display ce permite afisarea culorilor albastru si galben pe un fundal negru, cu o rezolutie tipica de 128x64 pixeli.

Comunicarea este realizata folosind biblioteca SSD1306 prin protocolul I2C.

Ecranul este folosit pentru afisarea valorilor in timp real.

Modul Wi-Fi

Modulul Wi-Fi prezent pe placa de dezvoltare ESP32 este folosit pentru a trimite datele inregistrate prin protocolul mqtt catre un broker Mosquitto, pentru a fi stocate si ulterior afisate de dashboard.

Pentru implementarea protocolului MQTT este folosita biblioteca esp-mqtt.

Dashboard

Arhitectura server-side este bazata pe un singur docker-compose folosind containere preexistente pentru a asigura o configuratie simpla, intuitiva si modulara.

Laboratoare Folosite

* Laboratorul 0 - GPIO + Laboratul 3 - Timere. PWM

1. Gestionarea led-ului RGB si a buzzer-ului pasiv folosind LEDC

* Laboratorul 1 - UART

1. UART este protocolul folosit pentru incarcarea codului pe placa, precum si pentru monitorizarea output-ului de debug emis de aceasta.

* Laboratorul 2 - Intreruperi

1. Am introdus un ISR pentru prelucrarea garantata a butonului de intrerupere a alarmei

* Laboratorul 4 - ADC

1. Valoarea rezistentei LDR este dedusa din tensiunea masurata de controller-ul ADC

* Laboratorul 6 - I2C

1. Ecranul OLED SSD1306 este controlat prin protocolul I2C

Rezultate Obtinute

[Link Video Demo](#)

Am reusit sa duc la capat proiectul, obtinand ceea ce mi-am propus la inceput: un sistem IoT care monitorizeaza un mediu si alerteaza cand sunt depasite constrangerile acestuia.

Concluzii

Pentru imbunatatirea proiectului exista urmatoarele posibilitati:

1. Folosire unor senzori cu precizie si acuratete mai mare
2. Folosirea unui PCB custom pentru minimizarea proiectului
3. Optimizarea pentru Power Consumption, permitand functionarea pe durate lungi fara alimentare externa

Bibliografie/Resurse

[Datasheet LDR GL5528](#)

[Datasheet DHT11](#)

[GitHub](#)

[Dashboard](#)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/andrei.batasev/daniel.malanca>



Last update: **2026/05/24 11:44**