

ETH - Enviromental Telemetry Hub

Introducere

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

- **Ce face:** 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.
- **Care este scopul lui:** 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 aprinda lumina).
- **Care a fost ideea de la care am pornit:** Am observat ca este usor sa ignori calitatea aerului atunci cand petreci mult timp lucrind in interior. Am dorit un dispozitiv autonom care sa imi atraga atentia "fizic" (prin sunet si lumina) cand conditiile devin nepotrivite, fara sa trebuiasca sa deschid aplicatii externe pe telefon sau PC.
- **De ce este util:** Este extrem de util atat pentru sanatatea personala cat si pentru protejarea echipamentelor sensibile la fluctuatii de temperatura/umiditate sau chiar a plantelor care au nevoi particulare.

Descriere generală

Proiectul functioneaza pe baza unui microcontroler ESP32 care actioneaza ca unitate centrala. Acesta preia constant datele de la senzorii de mediu, le trimite la dashboard, si actioneaza actuatorii (rgb, buzzer & ecran).

Datele brute prelucrate de ESP32 sunt trimise printr-o retea Wi-Fi folosind protocolul MQTT catre o instanta de server aflata in cloud. Pe acest server sunt instalate aplicatii precum un broker Mosquitto (pentru receptia datelor), o baza de date (ex. InfluxDB) si o platforma de vizualizare (Grafana) care genereaza dashboard-ul online accesibil de pe orice dispozitiv.

Schema bloc a sistemului:

- **Modul Central de Procesare:** Microcontroler ESP32.
- **Module de Intrare (Input):**
 - **DHT11:** Modul digital care transmite catre ESP32 valorile masurate pentru temperatura mediului ambiant si umiditatea relativa.
 - **Fotorezistor (LDR):** Formeaza un divizor de tensiune, oferind o valoare analogica la pinii ADC ai ESP32-ului, direct proportionala cu intensitatea luminoasa.
- **Module de Iesire (Output / Feedback):**
 - **Infrastructura Online:** Datele colectate sunt trimise catre un VPS aflat online, care permite

vizualizarea istoricului acestora.

- **Display OLED SSD1306:** Ecran ce primește comenzi prin protocol I2C pentru a afișa parametrii exacti măsurati și mici elemente de interfață.
- **LED RGB:** Indicator de stare comandat prin semnale PWM. Semnalează starea generală (ex. Verde pentru normal, Rosu pentru alertă).
- **Buzzer:** Un modul audio acționat de un pin digital pentru a emite semnale acustice de avertizare în caz de parametri critici.

Hardware Design



Lista de piese componente:

- 1 x Placă de dezvoltare ESP32 WROOM 32-D
- 1 x Senzor de temperatură și umiditate DHT11
- 1 x Display OLED SSD1306 (I2C)
- 1 x Fotorezistor (LDR)
- 1 x LED RGB (cu pini dedicați pentru R, G, B)
- 1 x Buzzer (activ/pasiv)
- Rezistențe (ex: 10k ohmi pentru divizorul de tensiune LDR, rezistențe de limitare a curentului pentru LED-ul RGB)
- Fire de conexiune (Jumper wires M-M, M-F)
- 1 x Breadboard pentru prototipare

Software Design


Mediul de dezvoltare: Aplicația (firmware-ul) va fi scrisă folosind mediul framework-ul Rust Embassy.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[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: **2026/05/05 07:21**