

# Swarm Adaptive Network



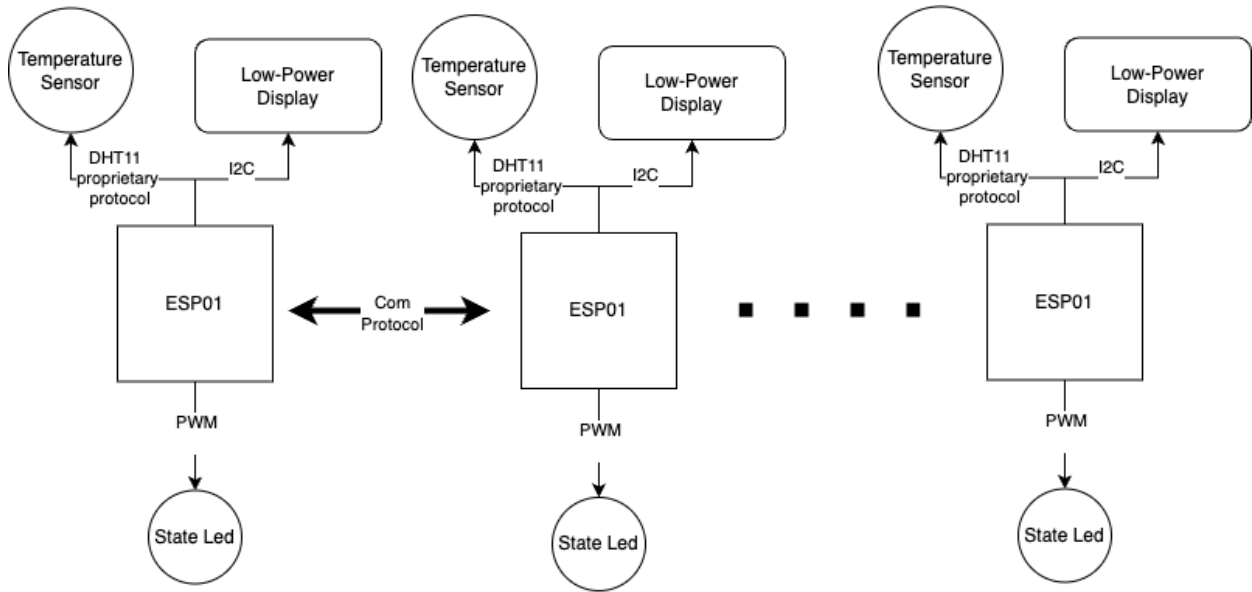
## Prezentarea proiectului

**Ce face sistemul?** Swarm Adaptive Network facilitează crearea unei rețele de plăcuțe care comunică autonom, funcționând fără conexiune WiFi pe întreaga zonă. Sistemul este capabil să suporte și să se recupereze după diverse anomalii.

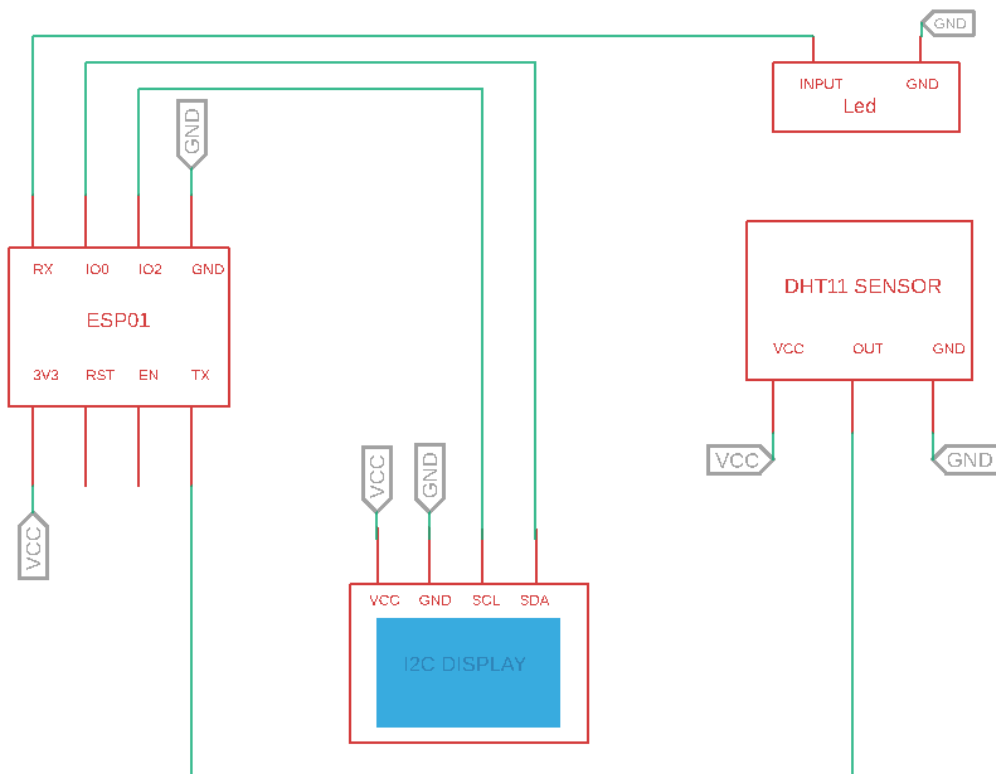
**Scopul sistemului?** Dezvoltarea unui ecosistem robust, optimizat pentru consum redus de energie și utilizarea cu microcontrolere.

**Sursa de inspirație?** Inspirat de interacțiunea flawless între dispozitivele Apple, mi-am propus să creez un sistem similar aplicabil în diverse scenarii.

**Utilitate?** Demonstrarea utilității sistemului prin dezvoltarea unei aplicații practice: un sistem de urmărire și răcire pentru zone vaste fără acoperire WiFi completă, cum ar fi un câmp de panouri solare.

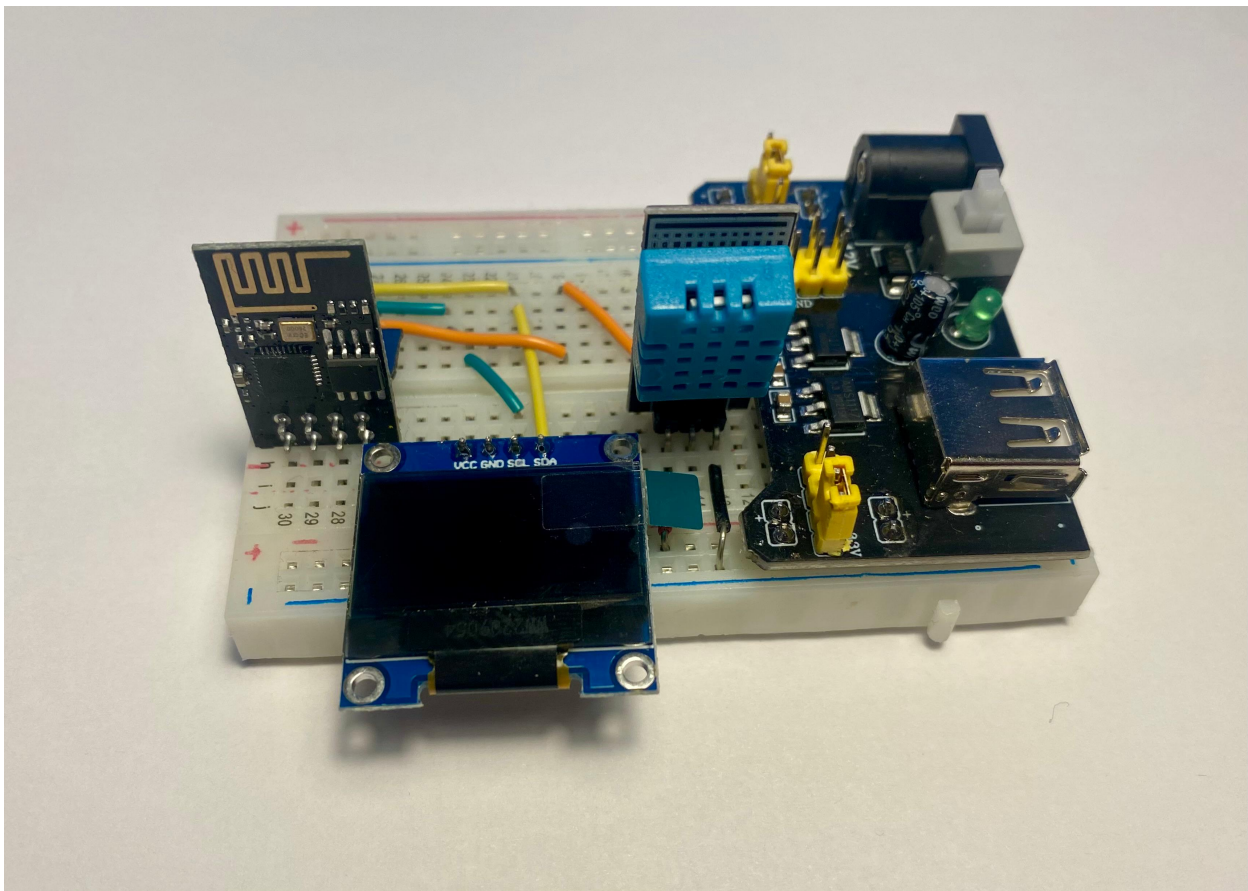


## Hardware Design



Tabelul cu piesele folosite se poate vedea mai jos.

Nume Piesa	Model	Protocol
Microcontroller	ESP01	-
Display	OKY4020	I2C
LED	-	PWM
Temperature Sensor	DHT11	Proprietary protocol



## Software Design

### **Motivati alegerea bibliotecilor folosite in cadrul proiectului \***

ESP8266HTTPClient - pentru a putea face request-uri HTTP catre serverul de baza \*  
ESP8266WiFi - pentru a putea conecta placa la un router WiFi \* ESP8266WebServer -  
pentru a putea crea un server, pentru setup-ul initial al placi \* U8g2lib - pentru a  
putea controla display-ul OLED, mai eficienta decat biblioteca Adafruit din punct de  
vedere al gestionarii memoriei \* ESP8266mDNS si ArduinOTA - pentru a putea face

update-uri Over The Air \* espnow - pentru a putea comunica intre placi fara a avea nevoie de un router WiFi \* Arduinojson - pentru a putea manipula JSON-uri

**Evidentiati elementul de noutate al proiectului** Comunicarea intre un Network de placi fara a avea nevoie de un router WiFi, folosind ESP-NOW. (Exemplu de utilizare: sistem de monitorizare a unui camp de panouri solare) Esp-now are o distanta de acoperire de 3 ori mai mare ca WIFI-ul. Utilizarea foarte simpla a unui sistem de 100 de dispozitive fara a fi nevoie de configurare manuala a fiecarei placi.

**Justificati utilizarea functionalitatilor din laborator in cadrul proiectului.** Am folosit notiunile din laborator mai mult ca si o baza, proiectul fiind axat foarte mult pe partea de software.

**Explicati scheletul proiectului, interactiunea dintre functionalitati si modul in care a fost validat ca acestea functioneaza conform** Scheletul implementeaza un cod universal care se afla pe fiecare placa, placile pot intra in mai multe moduri pentru a asigura o extindere si o asigurare a transmiterii informatiei.

**Explicati cum, de ce si unde ati realizat optimizari** Am realizat optimizari in ceea ce priveste consumul de energie, am folosit ESP-NOW pentru a comunica intre placi, fara a avea nevoie de un router WiFi, astfel consumul de energie este mult mai mic. Am incercat ca nodurile finale sa intre in sleep mode cat mai des posibil, pentru a economisi cat mai multa energie.



GitHub cu codul sursa



DEMO!

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

O arhivă 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 este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).

## Jurnal

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

## Bibliografie/Resurse

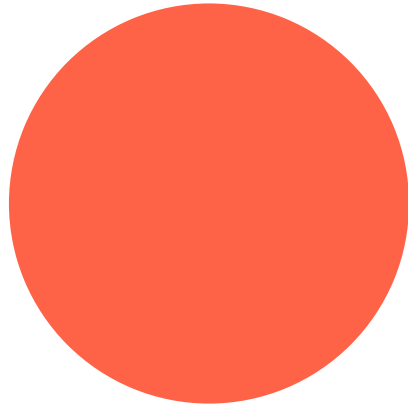
Listă cu documente, datasheet-uri, resurse Internet folosite, grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

Last update:  
2024/05/29 23:30

pm:prj2024:mdinica:andrei.anghelescu

<http://ocw.cs.pub.ro/courses/pm/prj2024/mdinica/andrei.anghelescu>



via GIPHY