

Sistem de udare automata a plantelor

I. Introducere

Obiective:

Sistemul IoT pune la dispozitia utilizatorului in timp real informații despre nivelul de umiditate din sol, nivelul apei din rezervorul din care se va uda solul, precum si starea pompei. Acest proiect este destinat unui sistem de irigație automatizat pentru plante.

Sistemul notifica utilizatorul in momentul când nivelul de umiditate din sol scade sub un anumit nivel. Acesta poate oricând sa activeze pompa de apa prin intermediul unei aplicații web. Cloud-ul si modulul Wi-Fi ESP8266 vor comunica prin protocolul de comunicatie MQTT.

Serverul reprezentat de laptop preia datele de pe cloud trimise de ESP folosind tot MQTT si stochează și pune la dispoziție o platforma de vizualizare printr-o interfață web. Clientul final comunica cu serverul web prin protocolul HTTP.

Actuatori:

- **pompa de apa:**

Aceasta va fi activata de clientul final care acceseaza interfata web si decide pe baza informatiilor daca porneste sau opreste pompa

Senzori:

- **nivel de umiditate din sol**
 - Măsoară umiditatea solului pentru a determina dacă plantele necesită apă.

nivel apa din rezervor

- **nivelul apei din rezervor**
 - Măsoară nivelul apei din rezervor pentru a preveni epuizarea apei și pentru

a asigura o gestionare optimă a resurselor

Acesti senzori vor prelua datele necesare utilizatorului.

II. Arhitectura

Componente hardware:

- **Senzor de umiditate a solului:** Pentru a detecta nivelul de umiditate din sol; daca scade sub un anumit nivel, microcontroller-ul trimite o notificare utilizatorului, si el va decide dacă trebuie udate plantele

 Optimus Digital



https://www.optimusdigital.ro/en/humidity-sensors/73-ground-humidity-sensor-module.html?gad_source=1&gclid=CjwKCAiAmfq6BhAsEiwAX1jsZyv3Hw0ktxKrLxnzthy02WAqDhtn-xwH9s1G2VA71xkORrzJyLnQUxoChOUQAvD_BwE

- **Pompă de apă:** Pentru transportul apei din rezervorul de apa către sol



Optimus Digital



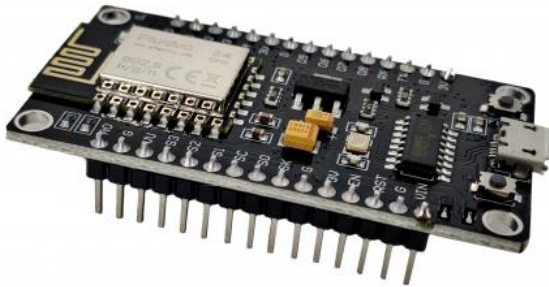
 Optimus Digital

-



-
- A blue printed circuit board (PCB) module featuring two large circular speakers with mesh grilles. A small ultrasonic sensor, labeled 'SR04', is mounted on the board. The board has several electronic components, including resistors and capacitors, and four pins extending from the bottom edge. Labels 'T' and 'R' are visible on the board.

- **Modul WiFi ESP8266:** Este responsabil de gestionarea mesajelor între dispozitivele IoT și serverul web. Publică datele pe anumite topicuri, iar serverul web este abonat la aceste topicuri pentru a obține datele și a le vizualiza pe aplicația web.



https://www.optimusdigital.ro/ro/placi-cu-wifi/266-placa-de-dezvoltare-wifi-cu-esp8266.html?search_query=esp8266&results=62

- **Alte componente:** Rezistențe, tranzistoare și fire de conectare: necesare pentru circuitele auxiliare

Software:

1. Biblioteci Arduino pentru MQTT și Wi-Fi si pentru conectarea la Cloud:

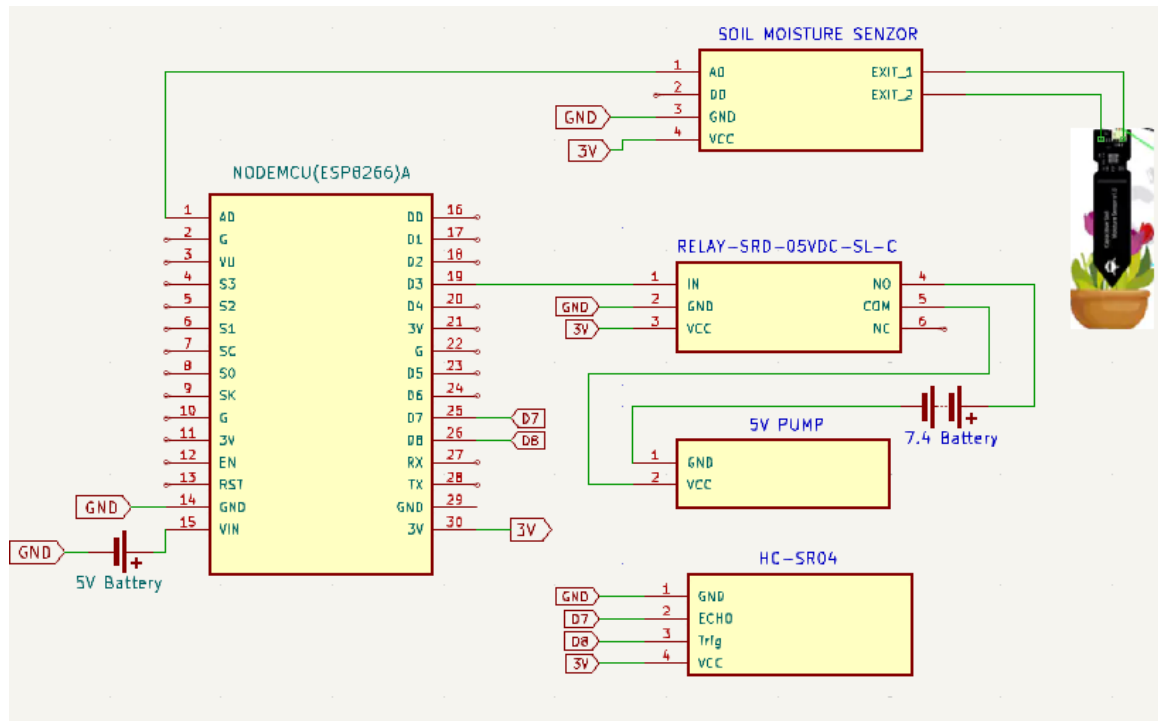
- ESP8266WiFi.h
- WiFiClientSecure.h
- PubSubClient.h
- ArduinoJson.h

2. Broker MQTT: Amazon IOT Core: Cloud-ul in care se stocheaza datele

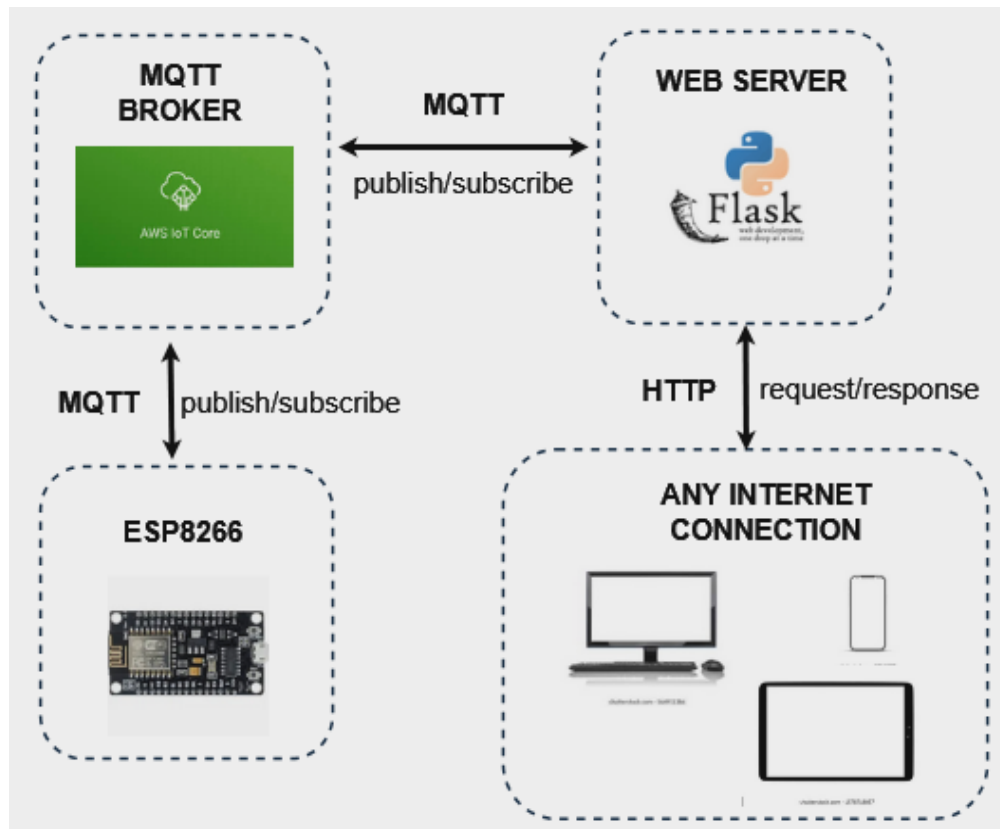
3. **Server Web (Python + Flask):** Crearea unui server simplu care să primească și să afișeze datele senzorilor; Serverul web se conectează la brokerul MQTT pentru a primi datele senzorilor și pentru a trimite comenzi către dispozitivele IoT (pentru a porni/opri pompa).

4. Grafana si InfluxDB folosind Docker: pentru stocarea si vizualizarea datelor

Schema elettrica:



Topologie



Protocoale de comunicare:

- MQTT (Message Queuing Telemetry Transport) este protocolul utilizat pentru a transmite date între microcontrolerul IoT și serverul web. Este un protocol ușor, eficient, perfect pentru dispozitive IoT care necesită o comunicare rapidă și în timp real.
- HTTP: Protocolul folosit pentru a comunica între utilizator și serverul web. Comenzile de control sunt trimise prin cereri HTTP (pornește/oprește pompa).

III. Implementare

Configurarea hardware-ului

Am folosit schema electrica pentru a realiza conexiunile între componentele hardware. Conexiunile de baza sunt:

- **Senzor de umiditate a solului:** Pin de semnal conectat la un pin analogic al ESP.

- **HC-SR04:** Trig și Echo conectați la pini digitali ai ESP.
- **Releu pentru pompă:** Intrare de comandă conectată la un pin digital configurat ca OUTPUT.
- **Wi-Fi Module:** Incorporat în ESP32/ESP8266 pentru conectivitate.

Configurarea software-ului

a. Cod pentru ESP32/ESP8266

Pentru implementarea dispozitivului IoT, am utilizat Arduino IDE împreună cu un set de biblioteci folosite pentru conexiunea la internet, criptarea datelor și comunicarea eficientă cu brokerul prin protocolul de comunicare MQTT.

Biblioteci utilizate

1. **WiFiClientSecure:**
 - Permite conexiuni criptate pentru securizarea comunicării cu brokerul MQTT.
 - Este esențială pentru transmiterea în siguranță a datelor între dispozitiv și server
2. **PubSubClient:**
 - Folosită pentru comunicarea MQTT între dispozitiv și brokerul Amazon IoT Core.
 - Gestionarea publicării și subscrierii la topicurile definite pentru transmiterea și primirea datelor.
3. **NTPClient:**
 - Responsabilă pentru sincronizarea ceasului intern al dispozitivului cu un server NTP, asigurând corectitudinea temporală a mesajelor.
4. **ArduinoJson:**
 - Utilizată pentru formatarea mesajelor în format JSON, necesar pentru schimbul de date structurate prin MQTT.

b. Flask & Python

Pentru gestionarea interfeței utilizatorului și a comunicării cu dispozitivele IoT, am utilizat **Flask**, un framework Python.

Structura Aplicației Web

1. **HTML și CSS:**
 - Interfața utilizator este realizată folosind HTML și CSS.
 - Include componente de afișare a datelor (umiditate sol, nivel apă) și butoane pentru pornirea/oprirea pompei.

- Datele senzoriale sunt preluate și afișate în timp real utilizând iframe-uri Grafana integrate.

2. Rute Definite:

- /: Ruta principală unde este afișată interfața utilizator.
- /api/control: Ruta prin care serverul primește cereri pentru controlul pompei.
- /api/data: Ruta apelată periodic pentru actualizarea datelor afișate în interfață.

3. Integrarea cu Grafana:

- Dashboard-urile Grafana sunt integrate în pagină prin iframe-uri, oferind o vizualizare detaliată a datelor colectate.

c. InfluxDB si Grafana folosind Docker

InfluxDB:

Este configurată pentru a primi datele primite de server.

Structura bazei de date este definită pentru a include:

- Timestamp.
- Valori senzor (umiditate, nivel apă).
- Starea pompei

Se trimite în baza de date sub forma:

```
"tags": {
"station": station,
"key": key
},
"fields": {
"value": float(value)
},
"timestamp": last_checked
```

Grafana:

- Dashboard-urile au fost configurate pentru a afișa grafic evoluția în timp a datelor colectate.
- Configurarea automată a fost realizată prin utilizarea folder-ului provisioning

Configurare Docker:

- Fișierul docker-compose.yml definește serviciile pentru InfluxDB și Grafana.
- Volumele sunt folosite pentru persistența datelor și a configurației Grafana.

Brokerul MQTT

Am folosit **Amazon IOT Core**

Crearea și Gestionarea Certificatelor

În AWS IoT Core, pentru fiecare dispozitiv (Thing) s-a creat un set de certificate:

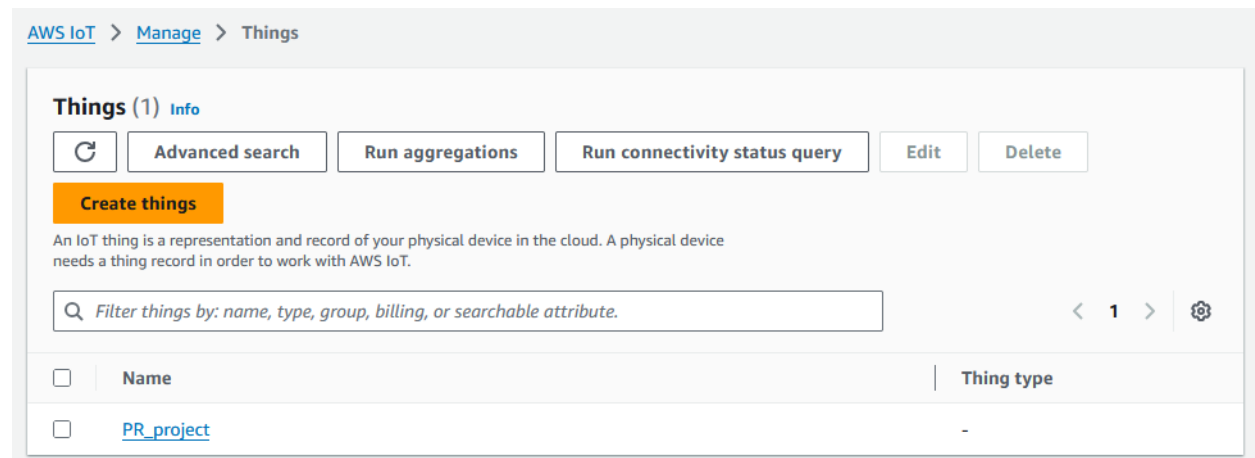
- **Certificatul CA.**
- **Certificatul clientului.**
- **Cheia privată.**

CertIFICATELE sunt descărcate și utilizate local pe dispozitivul ESP32/ESP8266.

Exemplu de configurare în AWS IoT Core:

- Se creează un nou „Thing” (dispozitiv).
- Se generează un certificat pentru „Thing”.
- Certificatul este descărcat și utilizat în codul ESP32/ESP8266.
- Politicile de securitate sunt configurate pentru a controla accesul la topicurile MQTT.

Configurarea acestuia s-a realizat creând un nou „Thing” și un nou „Certificat”:



[AWS IoT](#) > [Security](#) > Certificates

Certificates Info

X,509 certificates authenticate device and client connections. Certificates must be registered with AWS IoT and activated before a device or client can communicate with AWS IoT.

[Certificates](#) | Certificates you've transferred

Certificates (1)

Actions ▾

Add certificate ▾

| <input type="checkbox"/> | Certificate ID ▾ | Status ▾ | Created |
|--------------------------|--|----------|-----------------------------|
| <input type="checkbox"/> | 4d7ed773fdc6c79be89e7e1bb605a9fd3d6824432912ba9d211820c3c20a2714 | ✓ Active | December 11, 2024, 10:38:56 |

Certificatele se regasesc in AWS_IOT_Certificates si sunt utilizate atat de dispozitivul IOT, cat si de serverul web pentru a comunica si a pastra securitatea datelor.

Comunicatia MQTT

Se folosesc 4 topicuri:

A. de la dispozitivul ESP8266 la server

- ESP8266/umiditate_sol
- ESP8266/nivel_apă
- ESP8266/stare_pompa

B. de la server la ESP8266

- web_server/command -> se controleaza pompa printr-o comanda a utilizatorului

IV. Vizualizare si procesare de date

Pentru stocarea și vizualizarea datelor colectate, am utilizat **InfluxDB** și **Grafana**, două soluții populare pentru monitorizarea și analiza datelor în timp real.

1. Colectarea și Transmiterea Datelor

a. Dispozitiv IoT (ESP32/ESP8266)

- Datele sunt preluate de la senzori (umiditate sol și nivel apă)

- Fiecare citire este însoțită de un timestamp generat prin sincronizarea SNTP pentru a asigura o ordine cronologică precisă.
- Datele sunt formate în JSON și transmise prin MQTT către brokerul MQTT (Amazon IoT Core).

b. Serverul Flask

- Serverul primește datele prin intermediul unui broker MQTT (Amazon IoT Core).
- Aceste date sunt procesate și stocate într-o bază de date InfluxDB pentru persistență și analiză ulterioară.

2. Procesarea Datelor

a. Procesarea la Nivel de Server

Se creează serii temporale specifice în baza de date InfluxDB pentru:

- Umiditatea solului.
- Nivelul apei din rezervor.
- Starea pompei (activată/dezactivată)

b. Utilizarea Interogărilor

Detaliile de construcție a dashboard-ului sunt următoarele:

- Numele dashboard-ului
- Formatul numelor seriilor de timp: STATIE.METRICA (exemplu: ESP8266.nivel_apa);
- Interval de grupare a datelor: 1 secundă;
- Politica de grupare a datelor: medie aritmetică;
- Perioada de refresh automat al dashboard-ului: 5 secunde;

c. Integrarea cu Grafana

- Grafana este utilizată pentru vizualizarea datelor în timp real și istorice.
- Dashboard-urile sunt configurate pentru a afișa
- Evoluția umidității solului în timp și nivelul current
- Evoluția nivelului apei din rezervor în timp și nivelul current

IV. Funcționalități de Alertare

Se alertează utilizatorul în momentul când acționează pompa (se porneste / opreste).

V. Securitate

Comunicarea dintre dispozitive, serverul web și baza de date este securizată folosind criptare și autentificare, astfel încât să fie prevenite atacurile și accesul neautorizat.

1. Criptarea Comunicării

a. Utilizarea WiFiClientSecure pentru Conexiuni Criptate

- Am utilizat biblioteca **WiFiClientSecure** pentru a realiza conexiuni criptate între dispozitivul IoT (ESP32/ESP8266) și brokerul MQTT (Amazon IoT Core).
- **WiFiClientSecure** implementează protocolul TLS (Transport Layer Security), asigurând criptarea end-to-end a datelor transmise.

b. Implementarea Certificatelor X.509

- Pentru a valida identitatea dispozitivelor și a serverelor, s-au utilizat **certIFICATE X.509**.
- Aceste certificate sunt esențiale pentru autentificarea bidirecțională între client și brokerul MQTT.

c. Certificatul CA:

- Este utilizat pentru a valida identitatea serverului MQTT.
- Asigură că dispozitivul se conectează la brokerul MQTT corect și nu la un server intermediar falsificat.

d. Certificatul clientului și cheia privată RSA:

- Sunt utilizate pentru a valida identitatea dispozitivului IoT.
- Acestea sunt generate și gestionate prin AWS IoT Core în timpul configurării dispozitivului („Thing”).

Pornirea aplicatiei

Pornirea aplicatiei se face folosind scriptul run.sh, care contine comenzile:

- `sudo docker compose -f docker-compose.yml up --build -d ->` crearea imaginilor de docker ale bazei de date si aplicatiei de vizualizare
- `python3 main.py ->` pornirea serverului