



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Sistem irigatii plante

Proiectare cu microprocesoare

Autor:

Arduszădan Darius

Grupa: 6

Îndrumător de laborator:

Diaconescu Andrei

FACULTATEA DE AUTOMATICĂ
ȘI CALCULATOARE

Cuprins

1 Descriere generală a proiectului	2
2 Componente și tehnologii utilizate	2
2.1 Hardware	2
2.2 Software și Biblioteci	2
3 Diagrama de Conectare	3
4 Funcționalități și implementare	3
4.1 Monitorizarea mediului (Senzori)	3
4.2 Automatizarea (Modul AUTO)	4
4.3 Control de la distanță (Wi-Fi și Web Server)	4
5 Manual de utilizare	5
6 Concluzii	5

1 Descriere generală a proiectului

Proiectul **”Sistem de Irigare Smart”** este un dispozitiv IoT (Internet of Things) conceput pentru a automatiza îngrijirea plantelor. Practic, este un ”grădinar digital” care monitorizează constant umiditatea din sol și temperatura aerului. Dacă pământul devine prea uscat, sistemul pornește automat o pompă de apă.

Pe lângă automatizare, sistemul se conectează la Wi-Fi și creează o pagină web accesibilă de pe telefon, permitând utilizatorului să vadă starea plantei și să o ude manual de la distanță, dacă dorește.

2 Componete și tehnologii utilizate

2.1 Hardware

- **ESP32 Development Board** – ”Creierul” proiectului, care are Wi-Fi integrat și procesează datele.
- **Senzor Umiditate Sol (Capacitiv v1.2)** – Măsoară cât de ud este pământul. Am ales varianta capacativă pentru că nu ruginește în timp (spre deosebire de cei rezistivi).
- **Senzor DHT11** – Măsoară temperatură și umiditatea din aer.
- **Modul Releu 5V (2 Canale)** – Funcționează ca un întreceptor controlat electronic pentru a porni/opri pompa.
- **Mini Pompă de apă submersibilă** – Asigură udarea efectivă a plantei.
- **Breadboard și fire** – Pentru conectarea componentelor fără lipire.

2.2 Software și Biblioteci

Proiectul a fost programat în mediul **Arduino IDE**, folosind limbajul C++. Bibliotecile principale utilizate sunt:

- **WiFi.h** – Pentru conectarea placii ESP32 la routerul de internet.
- **WebServer.h** – Pentru crearea interfeței web (butoane și afișaj pe telefon).
- **DHT.h** – Bibliotecă specializată pentru citirea senzorului de temperatură.

3 Diagrama de Conectare

Pentru o mai bună înțelegere a modului în care componentele interacționează fizic, am inclus mai jos schema electrică a proiectului.

Aceasta evidențiază un aspect critic al implementării: **separarea surselor de alimentare**. Pompa și releul (consumatori mari) sunt alimentate separat de placă ESP32 și senzori (logică), având doar masa (GND) comună. Această configurație previne reșterea plăcii din cauza fluctuațiilor de curent.

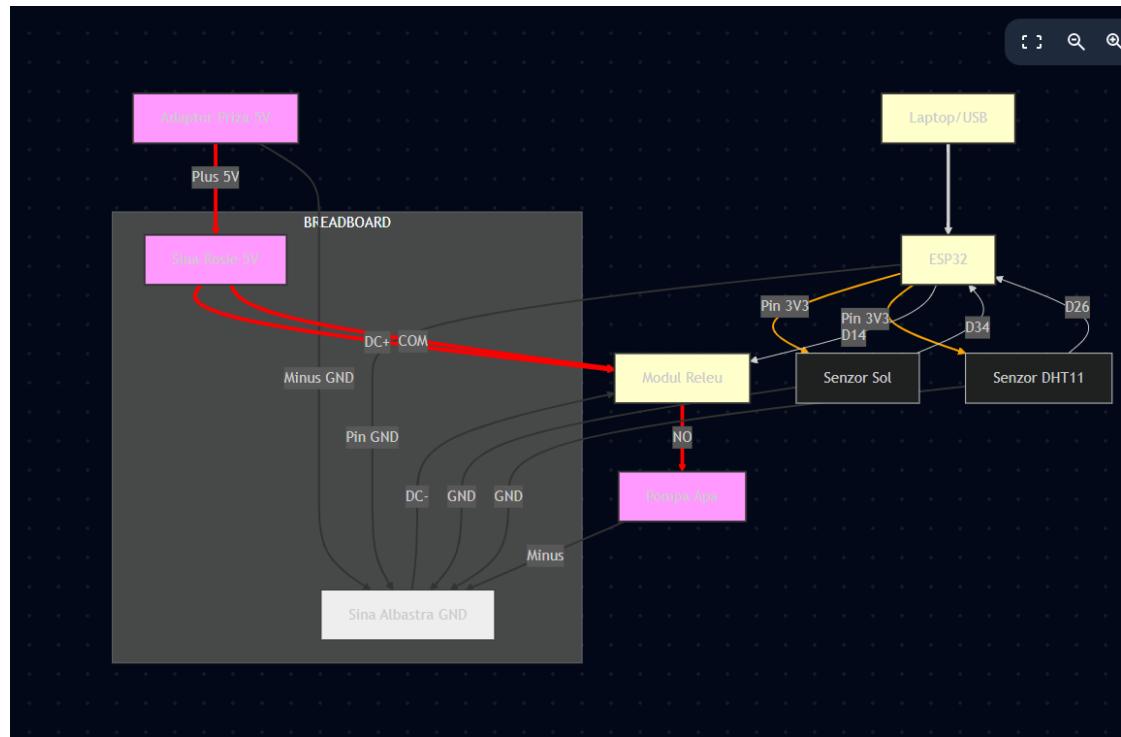


Figura 1: Schema electrică a componentelor

4 Funcționalități și implementare

4.1 Monitorizarea mediului (Senzori)

Sistemul citește datele de la cei doi senzori la fiecare 200 de milisecunde.

Cum funcționează: Senzorul de sol trimite o valoare analogică între 0 și 4095. Valoarea 4095 înseamnă aer (uscat complet), iar valorile mici (aprox. 1500-1600) înseamnă

apă (ud). În cod, am transformat aceste valori brute în procente (0-100%) pentru a fi ușor de înțeles de către om.

```
1 // Citire valoare bruta (0 - 4095)
2 int analogVal = analogRead(PIN_SOL);
3
4 // Transformare in procente (calibrare experimentală)
5 // 4095 = Uscat (0%), 1600 = Ud (100%)
6 umiditate_sol_procent = map(analogVal, 4095, 1600, 0, 100);
```

Listing 1: Conversia valorilor senzorului de sol

4.2 Automatizarea (Modul AUTO)

Aceasta este funcția principală care ține planta în viață. Sistemul verifică constant dacă umiditatea solului scade sub un prag critic (setat la 30%).

Logica: Dacă senzorul detectează pământ uscat (sub 30%), ESP32 trimite un semnal (LOW) către releu, care închide circuitul și pornește pompa. Când umiditatea revine la normal, pompa se oprește.

```
1 if (mod_auto == true) {
2     if (umiditate_sol_procent < prag_uscat) {
3         digitalWrite(PIN_RELÉU, LOW); // Porneste pompa
4         stare_pompa = "UDARE\u25b2(Auto)";
5     } else {
6         digitalWrite(PIN_RELÉU, HIGH); // Opreste pompa
7         stare_pompa = "Standby\u25b2(Umed)";
8     }
9 }
```

Listing 2: Logica de automatizare

4.3 Control de la distanță (Wi-Fi și Web Server)

Utilizatorul poate intra de pe telefon pe adresa IP a dispozitivului pentru a vedea datele și a controla pompa.

Implementare: Am creat un server web simplu care servește o pagină HTML. Când utilizatorul apasă un buton pe telefon (de exemplu ”PORNESTE”), browserul trimite o cerere către ESP32, care execută comanda și dezactivează temporar modul automat.

```
1 // Cand se apasa butonul de pornire manuala
2 server.on("/pompa_on", []() {
```

```

3 |     mod_auto = false; // Trecem pe manual
4 |     digitalWrite(PIN_REL, LOW); // Activam releul
5 |     stare_pompa = "PORNITA\u2022(Manual)";
6 |     server.send(200, "text/html", getHTML()); // Actualizam pagina
7 |

```

Listing 3: Gestionarea comenziilor de pe telefon

5 Manual de utilizare

Sistemul este gândit să fie simplu de utilizat:

1. **Pornirea:** Se alimentează sistemul (ESP-ul la USB, Pompa la priză).
2. **Accesarea interfeței:** Se deschide browserul pe telefon/laptop și se introduce adresa IP afișată în Serial Monitor (ex: 192.168.1.x).
3. **Monitorizare:** Pe ecran apar valorile curente: Umiditate Sol, Temperatură, Umiditate Aer.
4. **Control:**
 - Butonul **Mod AUTOMAT**: Lasă sistemul să decidă singur când udă.
 - Butoanele **PORNESTE / OPreste**: Oferă control manual instantaneu asupra pompei.

6 Concluzii

Proiectul îndeplinește toate cerințele unui sistem IoT funcțional. Reușește să mențină o plantă hidratată fără intervenție umană, dar oferă și flexibilitatea controlului manual prin internet.

Este o soluție practică pentru perioadele de vacanță sau pentru persoanele care uită să își ude plantele. Sistemul este stabil și separă alimentarea de putere (pompă) de cea logică (senzori) pentru a evita erorile.