

Sistem automat pentru udarea plantelor

Introducere

Sistemul automat pentru udarea plantelor verifica umiditatea din sol periodic si o mentine la un nivel constant prin pornirea unui sistem de udare prin picurare. Cu acest sistem fiecare planta are senzorul ei si e conectata la un rezervor, individual sau distribuit. Nivelul de umiditate poate fi setat de catre utilizator in functie de nevoile plantei si de tipul solului. De asemenea, sistemul poate fi accesat prin internet cu ajutorul unei aplicatii. Alte functionalitati pe care le ofera vor fi: afisarea temperaturii, log-uri pentru cand a udat, nivelul de apa din rezervorul fiecarei plante.

Descriere generală



Cu ajutorul telefonului sau a laptop-ului voi comunica prin wifi cu microcontroller-ul ESP32, astfel incat sa obtin de la el informatii in timp real despre temperatura ambientala si umiditatea din aer/sol din zona plantelor.

Hardware Design

Lista de componente:

- ESP32-C6 Super Mini,
- senzor de umiditate,
- senzor de temperatura,
- electrovalva,
- breadboard

Schema electrica:



Explicarea necesitatii componentelor:

1. ESP32-C6 Super Mini = microcontroller-ul care va gestiona conexiunea cu telefonul, va primi input de la senzor si va comanda pornirea sistemului de udare
2. Battery 5V = sursa de alimentare pentru valva, era insuficient curentul primit de la microcontroller
3. Soil moisture sensor = senzorul care masoara umiditatea din sol si pe baza caruia se va decide pornirea apei

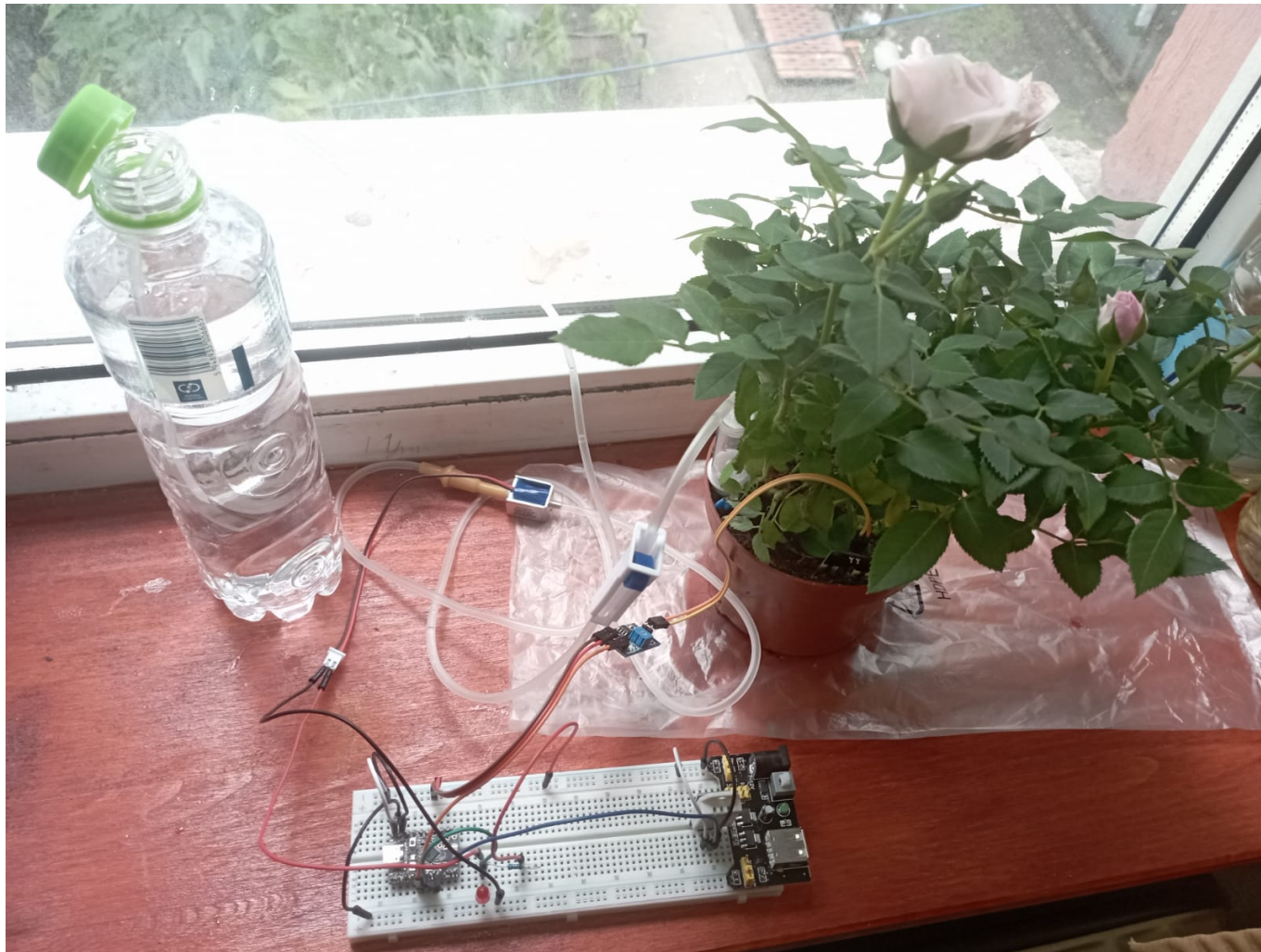
4. Valve = "robinetul" care va porni apa din rezervor

5. DHT22 = senzor de umiditate si temperatura pentru a citi temperatura ambientala si umiditatea aerului

Descriere a functionarii:

- Senzorul de umiditate sol e citit in mod analogic pe pinul GP1.
- Senzorul DHT22 e citit in mod digital pe pinul GP0.
- Valva e controlata de pinul GP2, care e configurat in modul open drain astfel incat valva sa isi ia curentul de la baterie dar sa fie controlata pornirea de pe microcontroller.

Poza cu hardware-ul:



Software Design

Mediul de dezvoltare: ESP-IDF (Espressif IoT Development Framework)

Biblioteci si surse third-party:

Pentru ca ESP nu ofera acces low level atat de usor ca Arduino, am fost nevoit sa folosesc mai multe biblioteci din framework-ul ESP-IDF.

- esp_wifi.h - pentru conectarea la rețeaua wifi
- esp_timer.h - pentru măsurarea timpului în sistemul de loguri
- nvs_flash.h - non-volatile storage, necesar pentru modulul de wifi
- esp_http_server.h - pentru a răspunde la cereri http cu informații despre parametrii măsurati

Pe lângă asta, am avut nevoie de o bibliotecă custom pentru senzorul de umiditate și temperatura DHT22, care implementează un protocol one-wire special pentru citirea datelor.

Funcționalități implementate:

1. server http pe placuță care afișează o pagină cu butoane din care poți trimite cereri prin wifi
2. citire senzor de umiditate sol în mod analogic
3. citire senzor de temperatură și umiditate DHT22 în mod digital (folosind bibliotecă specializată)
4. comandarea valvei de pornire a apei atunci când umiditatea ambientală este sub un prag
5. citirea logurilor prin cereri http

Link git: <https://github.com/D9nni/PM-Plant-Watering-System>

Rezultate Obținute

La finalul proiectului, placuța se conectează la rețeaua wi-fi prestabilită și primește o adresă IP la care poate fi accesată din browser, unde afișează butoane prin care poți face cereri despre parametrii importanți. Proiectul se comportă ca o mică stație meteo mai degrabă, care îți poate răspunde care este umiditatea, temperatura și umiditatea solului în timp real.

Partea de pornire a apei a eșuat din cauza hardware-ului prost ales, am avut o valvă de aer în loc de o valvă de apă sau de o pompă, care consumă mult curent și care până la urmă nu mai funcționează deloc.

Concluzii

În concluzie, proiectul a eșuat ca produs final, am câștigat doar experiența. Lipsa componentei necesare care era o pompă submersibilă mică sau o valvă mai serioasă pentru apă a fost o problemă esențială, de asemenea a durat mult să găsesc cum să programez placuța mea care e una dintr-o serie mai rară. Ar fi trebuit să lucrez cu ceva asemănător cu ce am avut la laborator pentru rezultate mai bune.

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

12 mai - am cumparat ESP32, senzor de umiditate sol si senzor de temperatura si umiditate a aerului DHT11

14 mai - am lipit placuta la laborator, am testat componentele, am folosit pentru inceput cod Arduino. Nu am reusit sa gasesc in platformIO placuta mea. Am ars din greseala senzorul DHT11.

20 mai - am comandat alt senzor DHT22 si o valva de aer in loc de pompa de apa pe care ar fi trebuit sa o am.

22 mai - Am constatat ca valva de aer consuma mult curent si nu o pot lega direct la un pin, am incercat sa folosesc pinul in modul open drain si sa pun o rezistenta de pull up catre vcc, dar curentul obtinut era prea mic pentru a porni valva. Valva se deschidea totusi daca o puneam direct la vcc, dar daca puneam o rezistenta cat de mica nu mai mergea. M-am gandit sa adaug o baterie, dar nu era ok sa bag curentul care vine din baterie in pinul de la esp.

23 mai - am inceput dezvoltarea codului pentru proiect folosind esp-idf pe linux

25 mai - La laborator mi s-a recomandat ca solutie un tranzistor IRLZ44N, nu era pe stoc, am luat altceva care avea tensiunea de poarta de prag prea mare, nu am reusit nimic cu el.

27 mai - Am cautat IRLZ44N si.. surpriza, am primit IRLZ234N, care tot nu comuta cu impulsul dat de pinul de pe esp. Ca ultima solutie am pus un buton care sa deschida valva manual pentru a face o demonstratie, dar pana la urma pare ca valva nu mai functioneaza deloc. Am imbunatatit codul, transformand esp-ul intr-un server http care raspunde la requesturi despre parametrii de mediu si prin care sa poti porni apa.

Bibliografie/Resurse

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

[Export to PDF](#)

Datasheet ESP32-C6FH4:

https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/andrei_daniel.lungu



Last update: **2025/05/27 20:13**