

Sistem automat de irigare pentru plante

Savu Bogdan 332 CB

Introducere

Voi realiza un sistem cu un Arduino care monitorizează umiditatea solului unei plante folosind un senzor. Când umiditatea scade sub un prag definit, se activează automat o pompă de apă pentru udarea plantei. Sistemul va monitoriza și nivelul apei din rezervor, iar când acesta scade sub un prag critic, va afișa un mesaj de avertizare pe un ecran OLED (probabil va afișa și în rest nivelul curent al apei pe ecran) și va activa un buzzer pentru alertare sonoră. De asemenea va avea și 2 butoane care vor permite ajustarea pragului de umiditate.

Descriere generală

Sistemul include următoarele module:

Modul de citire a umidității solului Acest modul folosește un senzor analog de umiditate conectat la o intrare ADC a Arduino-ului. Valorile sunt citite periodic și comparate cu un prag predefinit pentru a decide dacă planta are nevoie de apă. Acest modul se bazează pe conceptele din Laboratorul 4 (ADC).

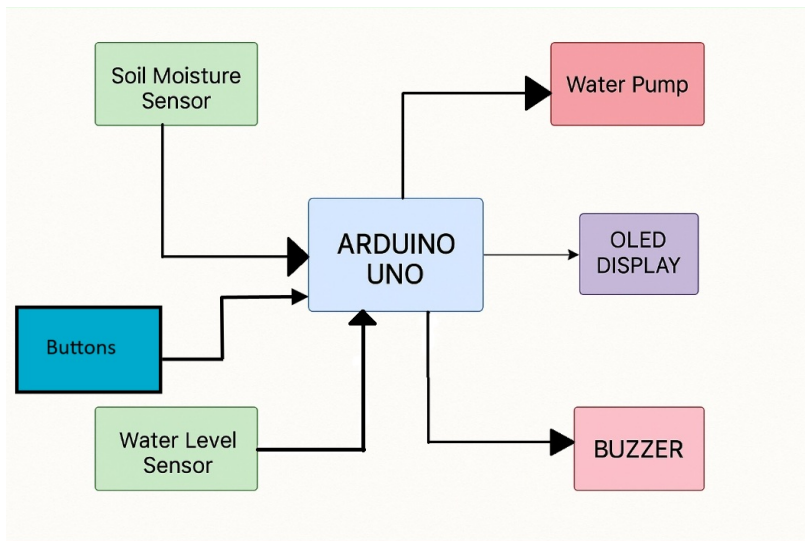
Modul de control al pompei de apă Atunci când solul este prea uscat, Arduino-ul activează o pompă de apă pentru a uda planta. Pompa este controlată printr-un semnal PWM generat de un timer hardware. Acest modul utilizează conceptele din Laboratorul 3 (Timere, PWM).

Modul de monitorizare a nivelului apei Pentru a evita funcționarea pompei în gol, sistemul include un senzor simplu de nivel (de tip "float switch") conectat la un pin digital configurat ca GPIO. Dacă nivelul apei este sub un prag minim, se oprește pompa și se declanșează alertele. Acest modul este legat de Laboratorul 0 (GPIO).

Modul de afișare și alertare Informațiile relevante (nivelul curent de apă, starea pompei, umiditatea solului) sunt afișate pe un ecran OLED conectat prin I2C. În plus, un buzzer este activat pentru avertizarea sonoră atunci când rezervorul este aproape gol. Acest modul folosește conceptele din Laboratorul 6 (I2C) pentru afișaj și GPIO pentru buzzer.

Modul de ajustare dinamică a pragului de umiditate Sistemul include două butoane fizice conectate la pini digitali (12 și 13), configurate ca intrări cu rezistență de pull-up. Apăsarea unuia

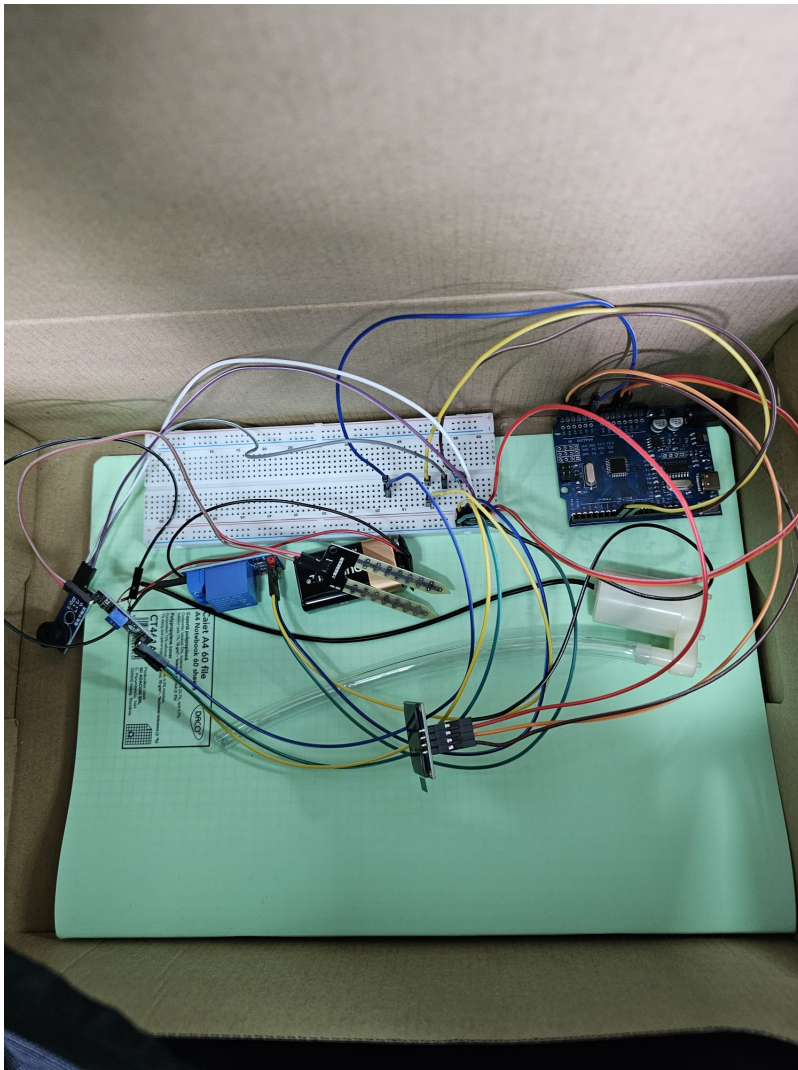
dintre butoane crește pragul de umiditate cu 10 unități, iar celălalt îl scade cu 10. Astfel, utilizatorul poate ajusta manual pragul de activare a pompei, în funcție de preferințele sale. La fiecare apăsare, valoarea curentă a pragului este afișată temporar pe ecranul OLED timp de 2 secunde, apoi ecranul revine la afișarea normală. Acest modul folosește conceptele din Laboratorul 0 (GPIO) și adaugă un plus de interactivitate sistemului.



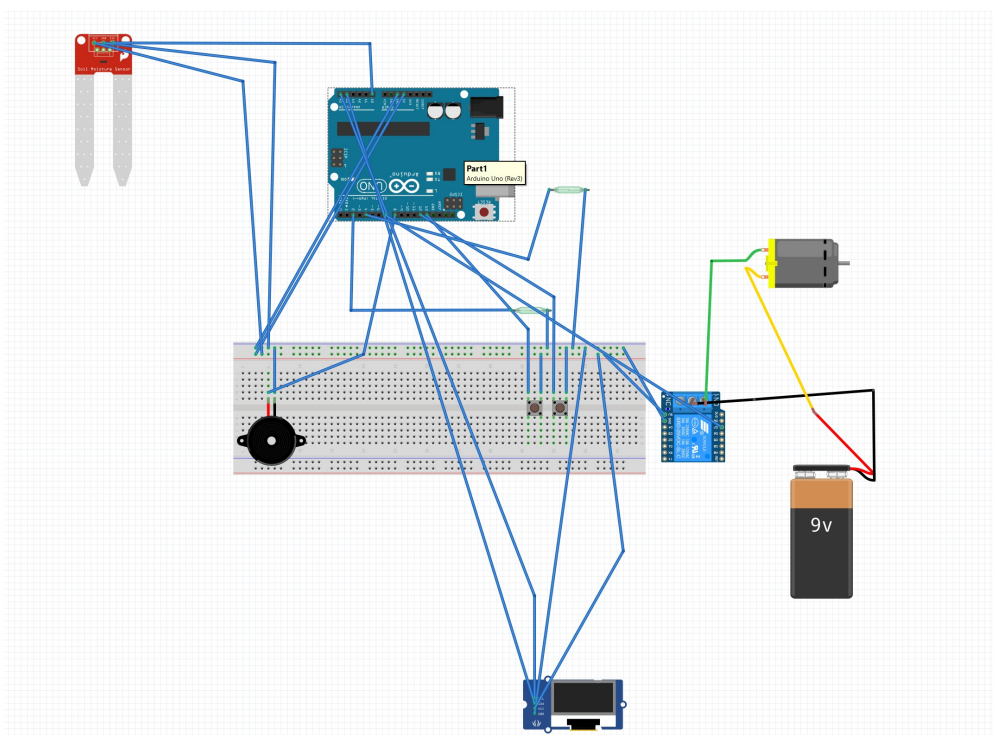
Hardware Design

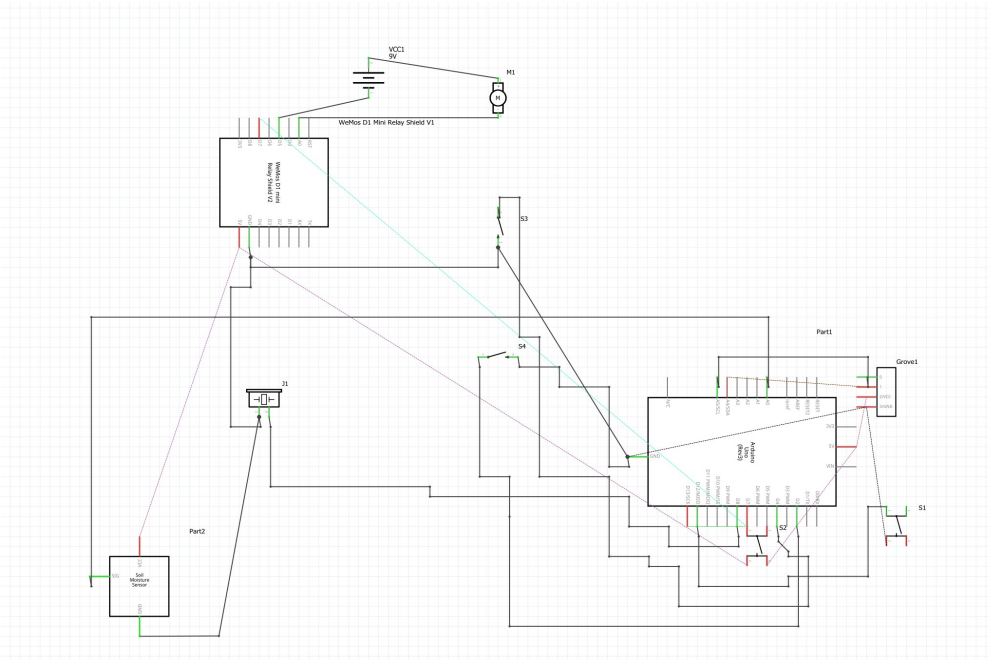
Lista piese:

- Arduino Uno
- Senzor de umiditate sol
- Pompă de apă
- Senzori nivel apă
- Buzzer
- Ecran OLED I2C
- Butoane
- Releu



Mai jos este design-ul schemei elctrice realizat in fritzing. Piesele nu corespund in totalitate cu cele utilizate in realitate, intrucat nu au fost toate disponibile.





Software Design

Descrierea codului aplicației (firmware)

Mediu de dezvoltare: aplicația a fost dezvoltată utilizând **Arduino IDE**, un mediu de dezvoltare integrat open-source, compatibil cu plăcile Arduino. Acesta oferă suport pentru încărcarea codului pe microcontroler, debugging prin Serial Monitor și gestionarea bibliotecilor externe.

Librării și surse 3rd-party:

Wire.h - permite comunicarea I2C între microcontroler și dispozitive periferice (în acest caz, ecranul OLED).

Adafruit_GFX.h - bibliotecă grafică oferită de Adafruit, necesară pentru a desena text și forme grafice pe ecran.

Adafruit_SSD1306.h - driver specializat pentru afișajele OLED bazate pe controllerul SSD1306 (folosit pentru controlul afișajului OLED 128×64).

Algoritmi și structuri implementate:

Citirea umidității solului cu ajutorul unui senzor analogic, conectat la pinul A0. Umiditatea este citită periodic și comparată cu un prag pentru a decide dacă este nevoie de activarea pompei.

Verificarea nivelului apei din rezervor folosind doi senzori digitali de nivel (conectați la pini digitali cu pull-up activat). Se determină trei stări posibile: LOW, MEDIUM și FULL.

Controlul pompei de apă se face printr-un releu conectat la pinul 7. Pompa se activează doar dacă umiditatea este sub un anumit prag și a trecut suficient timp de la ultima activare (cooldown).

Generarea unei alerte sonore printr-un buzzer activat în cazul în care nivelul apei este LOW. Buzzerul este controlat cu funcțiile `tone()` și `noTone()`, iar durata este limitată la 2 secunde pentru a evita poluarea fonică.

Afișare în timp real pe un ecran OLED: se afișează umiditatea solului și nivelul de apă. Dacă rezervorul este gol, pe ecran apare mesajul "Umpleți rezervorul".

(Etapa 3) Surse și funcții implementate:

`setup()` - inițializează toate resursele hardware: pini I/O, comunicare I2C, display OLED, Serial Monitor.

`loop()` - funcția principală care rulează continuu și:

citește valorile de la senzori,

activează/dezactivează pompa și buzzerul în funcție de condiții,

actualizează conținutul afișajului OLED.

Alte funcționalități auxiliare:

controlul timpului cu `millis()` pentru evitarea blocajelor,

comunicare serială cu `Serial.print()` pentru debugging.

Rezultate Obținute

Rezultatele obținute includ un sistem funcțional de irigare automată, capabil să detecteze umiditatea solului și nivelul apei, să activeze pompa la nevoie și să afișeze informațiile relevante pe un ecran OLED. Sistemul oferă și alertă sonoră în cazul unui nivel scăzut de apă.

<https://youtube.com/shorts/W2ttdbYQxm8>

Concluzii

- A fost un proiect interesant care m-a ajutat să învăț rapid bazele Arduino și să le aplic practic.
- Deși au fost momente dificile, am reușit să creez un sistem funcțional și am învățat foarte multe din proces.
- Integrarea componentelor hardware și scrierea codului de la zero mi-au oferit o înțelegere mai clară a modului în care funcționează sistemele embedded.

Download

O arhivă cu fișierele obținute în urma realizării proiectului:

[arhiva_cod.zip](#)

Jurnal

24.04.2025: Am ales tema proiectului și am schițat varianta la care aș vrea să ajung.

29.04.2025: Am comandat piesele necesare.

06.05.2025: Am început construcția părții hardware a proiectului.

13.05.2025: Am început implementarea software-ului pentru proiect.

Bibliografie/Resurse

<https://docs.arduino.cc/libraries/hx711/>

<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

<https://www.youtube.com/watch?v=ciD3ILxgXzU>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/rnedelcu/bogdan.savu2407>



Last update: **2025/05/28 13:42**