

Sistem automat de monitorizare și îngrijire a unei plante

Constantin Robert-Mihai 332CB

Introducere

Prezentarea pe scurt a proiectului vostru:

Proiectul constă în realizarea unui sistem automatizat de monitorizare a stării unei plante, folosind senzori și actuatori controlați de un microcontroler. Sistemul măsoară umiditatea solului printr-un senzor de umiditate. Dacă nivelul de umiditate este sub pragul optim, o pompă submersibilă este activată automat pentru a uda planta. Pentru a monitoriza nivelul apei din rezervorul de udare, se folosește un senzor ultrasonic orientat vertical, iar pe suprafața apei plutește o bucată de polistiren care reflectă undele sonore. Nivelul apei este afișat în timp real printr-o bandă de LED-uri roșii, fiecare LED indicând un anumit nivel. Când nivelul apei scade sub un prag critic, un buzzer este activat pentru a alerta utilizatorul. Proiectul este util pentru oamenii care vor să se bucure de plante în casă fără stresul că acestea ar putea muri subit.

Descriere generală

Acest proiect are ca scop dezvoltarea unui sistem automatizat și inteligent pentru monitorizarea și îngrijirea unei plante de apartament, folosind un set de senzori și actuatori controlați de un microcontroler Plusivo ESP8266. Este conceput special pentru persoanele care își doresc să aibă grijă de plante, dar nu dispun întotdeauna de timpul necesar sau uită să le ude regulat. Prin automatizarea acestor procese esențiale, proiectul contribuie la menținerea sănătății plantei fără efort din partea utilizatorului.

Funcționalități principale:

Monitorizarea umidității solului Sistemul folosește un senzor de umiditate introdus în pământul ghiveciului pentru a detecta nivelul de umiditate. Datele sunt citite constant de microcontroler. Dacă valoarea scade sub un prag predefinit (considerat optim pentru planta respectivă), sistemul declanșează udarea automată.

Udarea automată a plantei Pentru irigare se utilizează o pompă submersibilă alimentată separat, controlată electronic printr-un tranzistor MOSFET sau un releu conectat la ESP8266. Pompa este activată doar când este necesar, prevenind udarea excesivă.

Monitorizarea nivelului apei din rezervor Pentru a evita funcționarea pompei în gol, se implementează un senzor ultrasonic orientat vertical spre suprafața apei din rezervor. Pe apă plutește un obiect reflector (ex. polistiren) care reflectă undele ultrasonice înapoi spre senzor, permițând calculul înălțimii apei.

Afișarea vizuală a nivelului apei În funcție de distanța detectată de senzorul ultrasonic, o bandă de LED-uri roșii este aprinsă progresiv, fiecare LED reprezentând un anumit nivel al apei. Astfel, utilizatorul poate verifica vizual cu ușurință câtă apă mai este în rezervor.

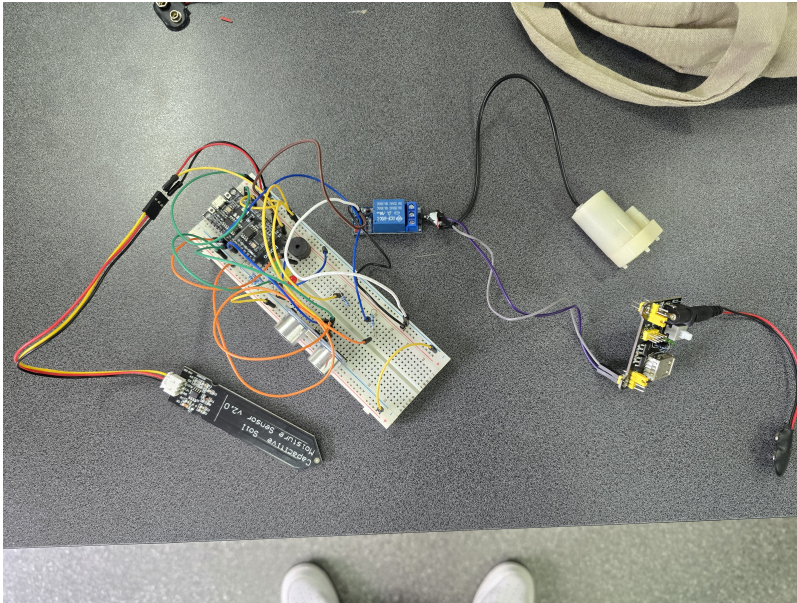
Alertă sonoră pentru nivel scăzut de apă Dacă apa scade sub un prag critic, este activat un buzzer piezoelectric pentru a alerta utilizatorul că rezervorul trebuie reumplut. Această funcție previne defectarea pompei și întreruperea udării.



Hardware Design

Piese hardware folosite:

- plăcuță Plusivo ESP8266
- senzor umiditate sol
- pompă de apă submersibilă
- senzor ultrasonic distanță
- 3 led-uri
- releu
- sursa de curent
- buzzer
- baterie 9V



Software Design

Aplicația reprezintă un sistem automat de irigare pentru plante, care monitorizează umiditatea solului și nivelul apei dintr-un rezervor. Când solul este uscat, sistemul pornește automat o pompă de apă. În paralel, monitorizează nivelul apei folosind un senzor ultrasonic și aprinde un LED corespunzător (roșu, galben sau verde). Dacă nivelul apei este scăzut, un buzzer redă o melodie de avertizare.

Mediu de dezvoltare:

- Arduino IDE
- Placă: ESP8266 Plusivo
- Limbaj: C/C++ specific platformei Arduino

Librării și surse 3rd-party:

- pitches.h: fișier header care definește frecvențele notelor muzicale, necesar pentru funcția tone()

Algoritmi și structuri implementate:

Citirea umidității solului:

Se folosește senzorul capacitiv conectat la pinul A0

Dacă valoarea citită depășește pragul `soilThreshold`, pompa este activată timp de 5 secunde printr-un releu

Măsurarea nivelului apei:

Se folosește senzorul ultrasonic HC-SR04 (pini TRIG și ECHO)

Timpul măsurat este convertit în distanță (în cm)

În funcție de valoarea distanței, este aprins un LED corespunzător (roșu = nivel scăzut, galben = mediu, verde = plin)

Avertizare sonoră:

Dacă nivelul apei este prea scăzut, se redă o melodie o singură dată, folosind un buzzer

Melodia este definită prin două tablouri (`melody[]` și `durations[]`) și redată cu `tone()` și `delay()`

Controlul LED-urilor:

Toți pinii LED-urilor sunt setați pe OUTPUT

Se aprinde un singur LED în funcție de distanța măsurată, celelalte fiind stinse

Funcții implementate:

`setup()` – inițializează portul serial, pini și componente

`loop()` – execută logica principală: citirea senzorilor, activarea pompei, actualizarea LED-urilor și cântatul melodiei

`measureEchoTime()` – trimite și primește impulsuri ultrasonice pentru a calcula distanța

`updateLEDs()` – aprinde LED-ul corespunzător în funcție de nivelul apei

playMelody() – redă melodia de avertizare sonoră prin buzzer

Rezultate Obținute

Senzorul de umiditate funcționează corect și transmite valori în timp real către microcontroler.

Pompa de apă pornește automat când nivelul de umiditate scade sub pragul stabilit și se oprește după udare.

Releul controlează în mod fiabil alimentarea pompei fără întârzieri sau blocări.

Senzorul ultrasonic detectează cu acuratețe nivelul apei din rezervor, pe baza reflexiei de pe flotor, prevenind pompa din a merge în gol.

LED-urile indică vizual nivelul apei în funcție de distanța măsurată, în trepte clare.

Buzzerul se activează corect când apa scade sub nivelul minim setat, avertizând utilizatorul.

Concluzii

- Am învățat să lucrez cu senzori și actuatori conectați la un microcontroler
- Am înțeles mai bine cum funcționează un releu și un senzor ultrasonic
- Mi-a plăcut să integrez partea hardware cu cea software, oferindu-mi o perspectivă în dezvoltarea proiectelor embedded

Download

Fișierele ce alcătuiesc componenta software

[proiectpm.zip](#)

Jurnal

24.04.2025: Am ales tema proiectului și am început documentarea cu privire la piesele necesare

30.04.2025: Am comandat piesele

06.05.2025: Am început construcția hardware a proiectului

13.05.2025: Am început dezvoltarea partii software

20.05.2025: Modificări finale și retușuri ale proiectului

Bibliografie/Resurse

<https://www.plusivo.com/electronics-kit/66-plusivo-wireless-super-starter-kit-with-esp8266-programmable-with-arduino-ide.html>

<https://projecthub.arduino.cc/tmekinyan/playing-popular-songs-with-arduino-and-a-buzzer-546f4a>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/rnedelcu/robert.constantin04>



Last update: **2025/05/28 13:52**