

Smart Pump - Pompă de Apă Inteligentă

Nume student: Chiriac Gabriel

Adresă e-mail: gabriel.chiriac3@student.tuiasi.ro



Nume student: Oancea Cosmin-Marian

Adresă e-mail: cosmin-marian.oancea@student.tuiasi.ro



Nume student: Dumea Emilian

Adresă e-mail: emilian.dumea@student.tuiasi.ro

Nume student: Munteanu Radu-Stefan

Adresă e-mail: radu-stefan.munteanu@student.tuiasi.ro



Motivația alegerii temei:

Am ales această temă pentru a explora aplicațiile practice ale microcontrolerelor Raspberry Pi Pico 2W în domeniul IoT (Internet of Things). Proiectul oferă oportunitatea de a învăța despre gestionarea inteligentă a resurselor de apă și integrarea componentelor electronice precum senzori, actuatori și module de comunicare Wi-Fi într-un sistem coerent și eficient.

Prin dezvoltarea acestui proiect, ne-am propus să înțelegem cum senzorii pot fi utilizați pentru monitorizarea continuă a condițiilor mediului (nivel apă și umiditate sol) și cum aceste date pot

influența controlul automatizat al unei pompe de apă. Relevanta practică a proiectului constă în aplicabilitatea sa directă în agricultură, grădini inteligente și automatizarea sistemelor de irigații.

Rezumat

Proiectul constă într-un sistem intelligent implementat pe un microcontroler Raspberry Pi Pico 2W. Sistemul include un senzor de nivel al apei, un senzor pentru măsurarea umidității solului, o pompă de apă, un LED de semnalizare și o interfață web integrată pentru control și monitorizare.

Sistemul colectează datele de la senzori, le prelucrează și activează automat pompa atunci când umiditatea solului scade sub un prag setat și există suficientă apă disponibilă. Utilizatorul poate controla și manual pompă prin intermediul unei interfețe web simplificate. Codul sursă este scris în MicroPython și utilizează protocoale HTTP și Wi-Fi pentru comunicare și control de la distanță.

Importanța în domeniul Embedded Systems și IoT

Proiectul demonstrează utilizarea eficientă a unui microcontroler compact și versatil (Raspberry Pi Pico 2W) într-un sistem IoT. Sistemul propus este relevant pentru automatizarea inteligentă și eficientă a resurselor naturale, exemplificând utilizarea senzorilor și a controlului automatizat într-o aplicație practică și utilă.

Ce se învață din replicarea proiectului?

1. Programare și Microcontrolere:

- Programare în MicroPython pentru controlul GPIO
- Gestionarea conexiunilor Wi-Fi și dezvoltarea unui server web integrat

2. Electronică aplicată:

- Integrarea senzorilor și actuatorilor în circuite electronice simple
- Utilizarea tranzistoarelor pentru controlul dispozitivelor cu putere mai mare decât permite direct microcontrolerul

3. Senzori și Actuatori:

- Folosirea ADC-urilor pentru citirea analogică a senzorilor
- Controlul pompei și al LED-ului în funcție de valorile citite

4. Comunicare în rețea:

- Protocol HTTP pentru interfață utilizator
- Monitorizare și control la distanță prin browser web

Implementare

- Asamblare hardware: Conectarea senzorilor, LED-ului și pompei la microcontroler, conform schemei electrice
- Dezvoltare software: Scrierea și testarea codului în MicroPython, configurarea conexiunii Wi-Fi și interfeței web
- Testare și optimizare: Verificarea funcționalității automate și manuale, ajustarea pragurilor pentru performanță optimă

Resurse utilizate

- Microcontroler: Raspberry Pi Pico 2W
- Senzor de nivel al apei: ADC digital
- Senzor umiditate sol: ADC digital
- Pompa de apă: Controlată prin tranzistor/MOSFET
- LED: Indicator al stării sistemului
- Cabluri și Breadboard: Conectare și testare rapidă

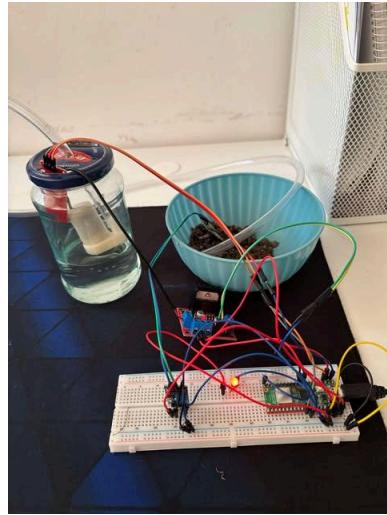
Implementare practică și interfața utilizator

Prototipul hardware

Imaginea de mai jos prezintă prototipul fizic realizat pentru sistemul Smart Pump. Acesta include microcontrolerul Raspberry Pi Pico 2W montat pe o placă de tip breadboard, împreună cu senzorii de nivel al apei și umiditate a solului, pompa de apă și conexiunile necesare pentru alimentare și comunicare.

Componenta esențială este microcontrolerul care preia datele de la senzori și controlează funcționarea pompei în funcție de pragurile definite. Circuitele sunt asamblate cu fire de conexiune colorate pentru o identificare clară a semnalelor și alimentării.

Această configurație permite testarea funcțională rapidă și ajustarea parametrilor înainte de realizarea unei versiuni finale mai compacte și optimizate.

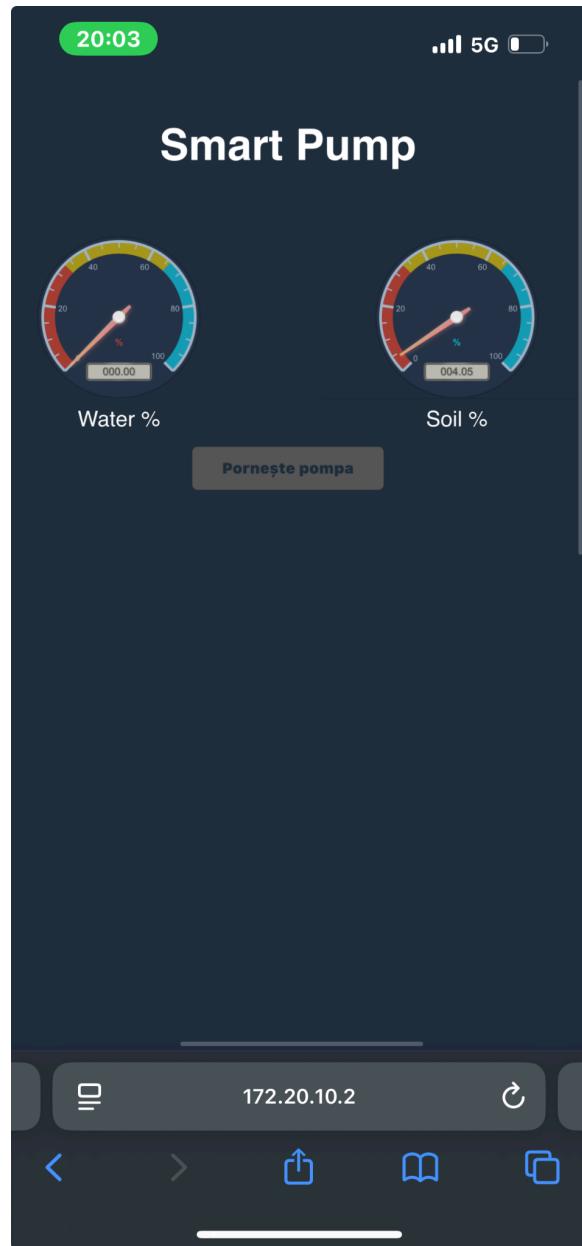


Interfață web

Imaginea următoare arată interfață web generată și servită de Raspberry Pi Pico 2W, accesibilă prin rețeaua Wi-Fi. Interfață afișează două indicatori grafici (gauge) care arată în timp real procentul de umiditate a solului și nivelul apei disponibile.

De asemenea, există un buton pentru pornirea manuală a pompei, care devine activ numai dacă nivelul apei este suficient pentru operare în siguranță.

Interfața este optimizată pentru dispozitive mobile, oferind o experiență intuitivă și vizuală, astfel încât utilizatorul să poată monitoriza și controla sistemul de oriunde.



Descriere detaliată componentă Smart Pump

Raspberry Pi Pico 2W (x1)



Raspberry Pi Pico 2W este microcontrolerul central al sistemului Smart Pump, care gestionează întregul proces de monitorizare și control al pompei de apă. Această versiune a Pico adaugă conectivitate Wi-Fi integrată (2.4 GHz IEEE 802.11n), permitând comunicarea fără fir și controlul de la distanță printr-o interfață web simplificată.

Microcontrolerul utilizează un procesor ARM Cortex-M0+ dual-core, cu o frecvență de până la 133 MHz, oferind performanță suficientă pentru citirea senzorilor analogici, gestionarea intrărilor și ieșirilor digitale și procesarea datelor în timp real.

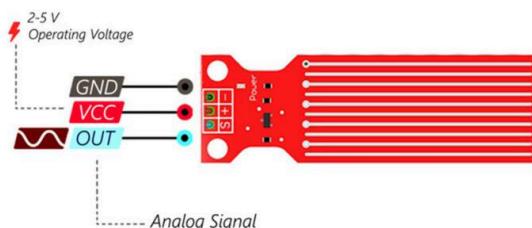
Datorită suportului pentru programarea în MicroPython, Raspberry Pi Pico 2W simplifică dezvoltarea rapidă a aplicațiilor IoT și embedded, făcându-l ideal pentru proiecte precum Smart Pump, unde conectivitatea Wi-Fi și controlul automatizat sunt esențiale. Raspberry Pi Pico 2W (x1)

Raspberry Pi Pico 2W este microcontrolerul central al sistemului Smart Pump, care gestionează întregul proces de monitorizare și control al pompei de apă. Această versiune a Pico adaugă conectivitate Wi-Fi integrată (2.4 GHz IEEE 802.11n), permitând comunicarea fără fir și controlul de la distanță printr-o interfață web simplificată.

Microcontrolerul utilizează un procesor ARM Cortex-M0+ dual-core, cu o frecvență de până la 133 MHz, oferind performanță suficientă pentru citirea senzorilor analogici, gestionarea intrărilor și ieșirilor digitale și procesarea datelor în timp real.

Datorită suportului pentru programarea în MicroPython, Raspberry Pi Pico 2W simplifică dezvoltarea rapidă a aplicațiilor IoT și embedded, făcându-l ideal pentru proiecte precum Smart Pump, unde conectivitatea Wi-Fi și controlul automatizat sunt esențiale.

Senzor nivel apă (Water Level Sensor)



Senzorul de nivel al apei măsoară nivelul apei folosind o ieșire analogică.

- **Pin-uri:**
 - **GND (masă):** conectat la pin-ul GND al Raspberry Pi Pico.
 - **VCC:** alimentare cu tensiune între 2 V și 5 V; se conectează la pin-ul 3.3 V al Raspberry Pi Pico.
 - **OUT:** semnalul analogic care variază proporțional cu nivelul apei; este conectat la ADC-ul pinului GP26.

Deoarece răspunsul senzorului nu evoluă strict liniar în funcție de adâncimea de imersie, a fost necesar să realizăm o măsurătoare empirică și să identificăm o funcție de regresie care să urmărească cât mai bine valorile înregistrate.

- **Măsurători empirice cu pas de 0,1 cm**

Am introdus senzorul în apă cu pași de 0,1 cm, notând pentru fiecare adâncime valoarea analogică (în unități digitale) pe care o raporta senzorul. În acest fel am obținut un set de date discrete de forma:

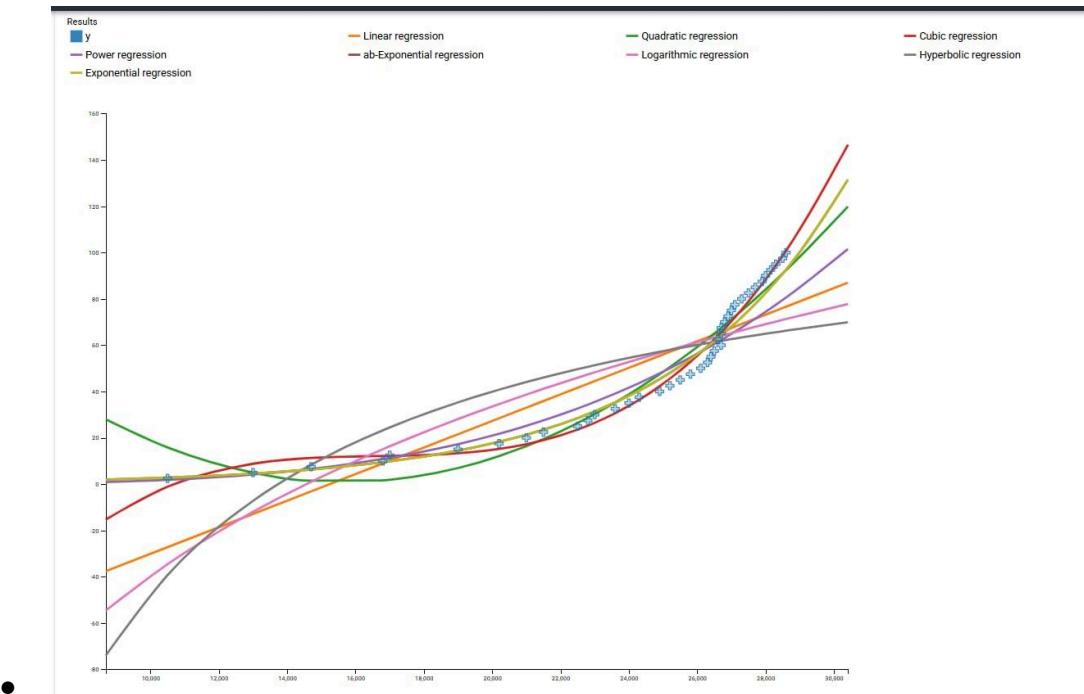
	A	B	C
1	0	8000	0
2	2.5	10500	2.5
3	5	13000	5
4	7.5	14700	7.5
5	10	16800	10
6	12.5	17000	12.5
7	15	19000	15
8	17.5	20200	17.5
9	20	21000	20
10	22.5	21500	22.5
11	25	22500	25
12	27.5	22800	27.5
13	30	23000	30
14	32.5	23600	32.5
15	35	24000	35
16	37.5	24300	37.5
17	40	24900	40
18	42.5	25200	42.5
19	45	25500	45
20	47.5	25800	47.5
21	50	26100	50
22	52.5	26300	52.5
23	55	26400	55
24	57.5	26500	57.5
25	60	26700	60
26	62.5	26600	62.5
27	65	26700	65
28	67.5	26700	67.5
29	70	26800	70
30	72.5	26900	72.5
31	75	27000	75
32	77.5	27100	77.5
33	80	27300	80
34	82.5	27500	82.5
35	85	27700	85
36	87.5	27900	87.5
37	90	28000	90
38	92.5	28150	92.5
39	95	28300	95
40	97.5	28500	97.5
41	100	28600	100
42			

• Analiza diferitelor regresii

După ce am strâns valorile pentru fiecare adâncime, am reprezentat grafic punctele obținute (semnal senzor în funcție de adâncime). Apoi am testat mai multe tipuri de regresii (lineară, exponențială, logaritmică, polinomială de grad 2, polinomială de grad 3 etc.) pentru a determina care model matematic se potrivește cel mai bine.

Rezultatul testării este prezentat în figura de mai jos:

- Punctele albastre reprezintă măsurătorile efective.
- Curba roșie (regresia cubică) urmărește cel mai bine distribuția datelor.
- Celelalte curbe (galbenă – exponențială, violetă – putere, verde – quadratică, portocalie – liniară, roz – logaritmică, gri – hiperbolică) au fost incluse doar pentru comparație și deviază mai mult de la forma reală a punctelor.



• Rezultatul regresiei cubice

Din analiza vizuală și din calculul erorii pătratice medii, am ales ca model final o funcție cubică de forma:

$$y(x) = A_3 x^3 + A_2 x^2 + A_1 x + A_0$$

unde:

- x este semnalul brut al senzorului (în unități digitale),
- y este nivelul apei, exprimat în procente (%),
- A_3, A_2, A_1, A_0 sunt coeficienții găsiți prin metoda celor mai mici pătrate.

Coeficienții obținuți sunt:

$$A_3 = 3.950844075038 \cdot 10^{-11}$$

$$A_2 = -1.8232912289820 \cdot 10^{-6}$$

$$A_1 = 0.0279015$$

$$A_0 = -131.227$$

Prin urmare, funcția finală este:

$$y(x) = (3.950844075038 \cdot 10^{-11}) \cdot x^3 - (1.8232912289820 \cdot 10^{-6}) \cdot x^2 + (0.0279015) \cdot x - 131.227$$

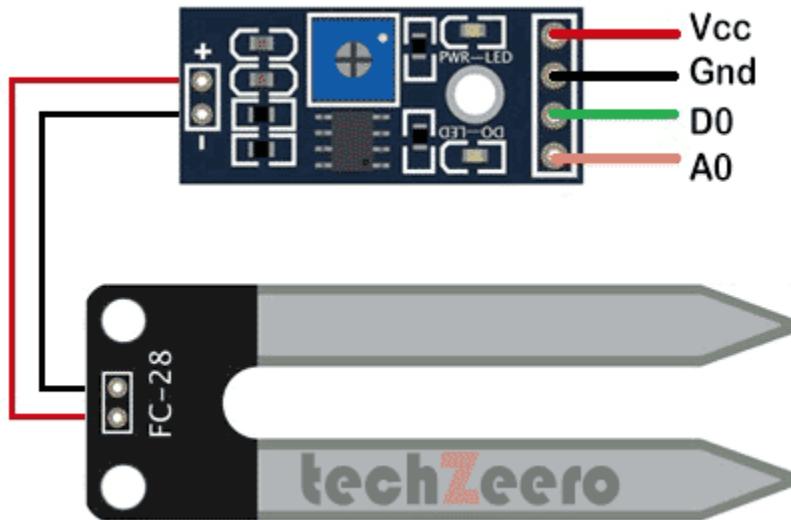
• Verificarea modelului

- Pentru câteva valori de test (semnal brut), am comparat nivelul de apă calculat cu cel măsurat fizic.
- Pe întreg intervalul de lucru, am obținut o eroare maximă de $\pm 1\%$ față de măsurările manuale, ceea ce este suficient pentru aplicația noastră.

• Funcție practică

Acest senzor ne ajută să detectăm nivelul de apă din rezervor, astfel încât să nu punem pompa să funcționeze în gol. Atunci când nivelul măsurat scade sub pragul minim definit (ex. 20 %), microcontroller-ul oprește pompa și aprinde LED-ul de avertizare; când nivelul revine peste prag, pompa poate fi repornită în condiții de siguranță.

Senzor umiditate sol (Soil Moisture Sensor)



Acest senzor detectează nivelul umidității solului, generând un semnal analogic.

- **Pin-uri:**
 - **VCC:** alimentare la pin-ul 3.3 V al Raspberry Pi Pico.
 - **GND:** pin-ul GND al Raspberry Pi Pico.
 - **A0 (Analog):** conectat la ADC-ul pinului GP27 pentru citirea analogică.
 - **D0 (Digital):** semnal digital ce poate indica atingerea unui prag specific (optional, nefolosit aici).

Înțial, am vrut să urmăram aceeași pașă ca și pentru senzorul de nivel al apei – adică măsurători empirice pe probe de pământ cu umidități cunoscute, graficarea valorilor și ajustarea unei curbe de regresie. Din păcate, nu am reușit să obținem un număr suficient de mare de mostre de sol cu umidități precise (adică probe cu procent de apă cunoscut) pentru fiecare pas de măsurare.

Prin urmare, în codul final am implementat direct conversia ADC → procent umiditate, fără calibrare detaliată pe probe de referință:

```
raw_s = read_soil_raw()  
s_pct = 100 - raw_s * 100 / 65535
```

- `read_soil_raw()` returnează un număr între 0 și 65535, proporțional cu tensiunea de la senzor:
 - Dacă senzorul este complet în apă (sol foarte umed), `raw_s` tinde spre 0.
 - Dacă senzorul este complet în aer (sol foarte uscat), `raw_s` tinde spre 65535.
- Expresia `raw_s * 100 / 65535` produce procentul de „uscat” al solului:
 - Când `raw_s = 0` → `raw_s * 100 / 65535 = 0` (0 % uscat).
 - Când `raw_s = 65535` → `raw_s * 100 / 65535 = 100` (100 % uscat).
- Inversând scala cu $100 - (\dots)$, obținem procentul de umiditate:
 - Când `raw_s = 0` (sol complet umed) → `s_pct = 100 %`.
 - Când `raw_s = 65535` (sol complet uscat) → `s_pct = 0 %`.

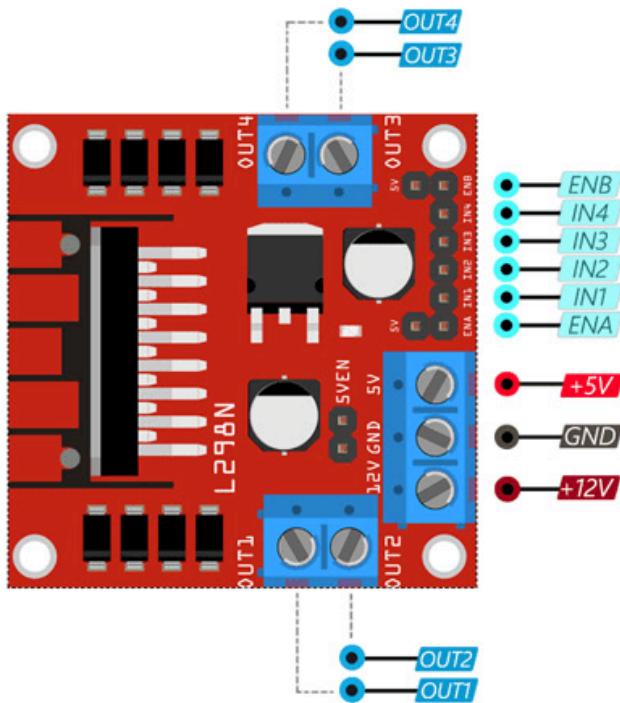
Astfel, `s_pct` indică direct procentul de umiditate al solului, crescând odată cu cantitatea de apă din sol și scăzând pe măsură ce solul se usucă.

Prin această metodă am obținut rapid o măsurare suficientă pentru aplicație, evitând necesitatea unui proces laborios de calibrare empirică pe multiple tipuri de probe de sol.

• Funcție practică

În firmware, valoarea `s_pct` ne ajută să decidem când să trimitem semnal la pompa de apă. De exemplu, dacă `s_pct` scade sub un prag țintă (`SOIL_TARGET`), Pico-ul activează pompa pentru a crește umiditatea solului. Odată ce `s_pct` revine peste prag, pompa se oprește automat.

Modul driver motor L298N



Modulul L298N controlează pompa de apă prin comutarea tensiunii și curentului motorului.

- **Pin-uri relevante:**
 - **IN1, IN2, IN3, IN4:** controlul direcției motorului.
 - **ENA, ENB:** enable pentru canale (permite comutarea curentului către motor; aici pot primi semnal PWM pentru reglajul vitezei).
 - **OUT1 – OUT4:** ieșirile către motoare.
 - **+5V și GND:** alimentarea logică pentru modul.
 - **+12V:** alimentarea de putere pentru motor.
- **Jumpere adăugate și explicație**
 - **Jumper pe 5V EN:** Am montat acest jumper pentru a activa regulatorul intern de 5 V al modulului (derivat din intrarea de 12 V). Fără jumper, modul nu ar furniza tensiunea de 5 V necesară logicii interne, iar pinii IN1–IN4 nu ar putea fi comandați corect.
 - **Jumper pe ENA:** În cazul în care punem jumper pe ENA, pinul ENA rămâne legat în permanent la nivel HIGH (5 V), ceea ce înseamnă că acel canal de motor este activ mereu la putere maximă, fără posibilitatea de reglaj prin PWM. Motorul primește întotdeauna tensiunea completă de 12 V și rulează la viteză maximă. Dacă dorim să reglăm viteza pompei prin PWM, nu trebuie să punem jumper pe ENA, ci să conectăm pinul ENA la un pin PWM al Raspberry Pi Pico. Astfel:

- **Fără jumper pe ENA:** Pico poate trimite un semnal PWM către ENA, modificând factorul de umplere (duty cycle) și, implicit, viteza motorului.
- **Cu jumper pe ENA:** pinul ENA este mereu la 5 V, deci nu mai există semnal PWM; canalul este activat permanent la puterea maximă disponibilă.

Pompa de apă DC



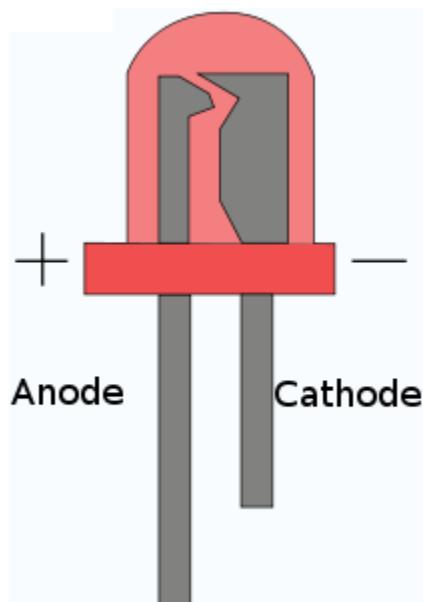
Pompa este un actuator folosit pentru pomparea apei în sistem.

- **Specificații隐含:**
 - Tensiune de alimentare: 12 V DC (livrată prin driverul L298N)
 - Current nominal: depinde de model (în jur de 0,2–0,5 A)
 - Debit: tipic 1–2 L/min, adâncime de refulare ~1–2 m
- **Pin-uri/conductoare:**
 - **Fir Roșu (+):** conectat la ieșirea driverului motor L298N (OUT1/OUT2 sau OUT3/OUT4, în funcție de canal). Acesta primește +12 V atunci când driverul este activ și permite pornirea pompei.

- **Fir Negru (-):** conectat la masa comună a sistemului (GND). Masa comună trebuie să fie comună între Raspberry Pi Pico, driverul L298N și sursa de 12 V.
- **Mod de conectare:**
 - Asigurați-vă că driverul L298N este alimentat la 12 V (la bornă +12 V → pompa poate trage curentul necesar).
 - Setați jumper-ul **5V EN** pe driverul L298N pentru a activa regulatorul intern de 5 V (pentru logica modulului).
 - Dacă doriți viteză constantă maximă, plasați jumperul pe **ENA** (sau **ENB**, în funcție de canal). Dacă doriți reglaj PWM, scoateți jumperul de pe ENA și conectați pinul ENA la un pin PWM al Raspberry Pi Pico.
 - Conectați firul roșu al pompei la unul dintre terminalele OUT ale driverului L298N (de exemplu, OUT1).
 - Conectați firul negru al pompei la masa comună (GND) a sistemului.
- **Observații practice:**
 - Pompa va porni doar dacă driverul L298N furnizează +12 V pe terminalul OUT corespunzător (în funcție de semnalele IN1/IN2 și ENA).
 - Dacă ENA este la nivel HIGH permanent (jumper montat), pompa va primi întotdeauna tensiunea de 12 V completă și va funcționa la viteză maximă.
 - Dacă ENA primește semnal PWM de la Pico (jumper demontat), viteza pompei poate fi reglată în funcție de factorul de umplere al semnalului PWM.

În acest mod, pompa de apă este intercalată pe canalul de putere al modulului L298N și poate fi controlată de către Raspberry Pi Pico prin semnalele digitale de direcție și enable.

LED indicator



LED-ul indică starea sistemului (pornit/oprit) și avertizează atunci când nivelul apei devine prea scăzut sau revine peste pragul critic.

- **Pin-uri:**
 1. **Anod (+):** conectat printr-o rezistență (de ~220 Ω) la pin-ul GP15 al Raspberry Pi Pico, pentru control.
 2. **Catod (-):** conectat direct la masa comună (GND).
- **Comportament:**
 1. Când nivelul apei **scade sub pragul critic** (de exemplu, sub 30 % în funcție de calibrare), microcontroller-ul activează LED-ul (trimite GP15 → HIGH), semnalizând o stare de atenționare „apă redusă”.
 2. Când nivelul apei **revine peste pragul critic** (de exemplu, peste 90 %), microcontroller-ul oprește LED-ul (trimite GP15 → LOW), indicând că problema a fost rezolvată și nivelul apei este din nou suficient.
- **Explicație detaliată:**
 1. În fiecare ciclu de măsurare, firmware-ul citește semnalul de la senzorul de nivel al apei și calculează procentul de umplere (`w_pct`).
 2. Dacă `w_pct` este **sub pragul inferior** (configurat, de ex., la 30 %), LED-ul se aprinde, atrăgând atenția asupra necesității de a alimenta sistemul cu apă sau de a verifica pompa.
 3. După ce pompa sau alimentarea cu apă aduce nivelul înapoi **peste pragul superior** (configurat, de ex., la 90 %), LED-ul se stinge automat, semnalizând revenirea la starea normală.
 4. Această logică cu două praguri (hysteresis) previne aprinderea/dezaprinderea frecventă a LED-ului în cazul unor fluctuații minore din jurul valorii critice.

Astfel, LED-ul servește atât ca indicator de avertizare („nivel apa scăzut”), cât și ca indicator de revenire la condiție de funcționare normală, fără bătăi de cap.

Cod

Microcontrolerul Raspberry Pi Pico 2W orchestrează monitorizarea continuă a nivelului apei și a umidității solului prin senzori analogici, automatizând astfel procesul de irigare în funcție de condițiile mediului. Prin conexiunea Wi-Fi, se pune la dispoziție o interfață web intuitivă, accesibilă de la distanță, unde pot fi vizualizate în timp real valorile citite și se poate comanda manual pompa atunci când este necesar.

Aplicația, implementată în MicroPython, cumulează următoarele funcționalități:

- Gestionarea conexiunii Wi-Fi și configurarea serverului HTTP.
- Citirea valorilor de la senzorii de nivel apa (prin ADC) și de umiditate sol.
- Convertirea semnalelor analogice în procente (inclusiv o aproximare cubică pentru senzorul de nivel apă).
- Logică de control automat al pompei bazată pe praguri configurabile (nivel minim/apropiere de gol și întă de umiditate sol).

- Interfață web modernă, creată cu HTML și JavaScript, care afișează grafic nivelul apei și umiditatea solului și permite pornirea manuală a pompei.

Cod Github: https://github.com/cosmin-web/proiect-SM_Smart-Pump/tree/main

Bibliografie și Resurse utile

- Documentație oficială Raspberry Pi Pico W:
<https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>
- MicroPython Documentation: <https://docs.micropython.org/en/latest/>
- Canvas Gauges: <https://canvas-gauges.com/>