

AquaBuddy - Sistem Inteligent de Asistență pentru Plante

Introducere

Proiectul reprezintă un sistem inteligent de gestionare a irigațiilor. Dispozitivul monitorizează nivelul de umiditate din sol și controlează o pompă de apă pentru a menține un mediu optim de creștere. Utilizatorul interacționează cu sistemul printr-o interfață formată dintr-un ecran LCD și 3 butoane de control, care permit setarea pragurilor și alegerea modului de lucru.

Ideea a pornit de la nevoia de a automatiza îngrijirea plantelor pe perioade lungi (concedii, sesiuni), asigurând în același timp o utilizare eficientă a apei. Utilitatea proiectului oferă siguranța că planta primește apă exact când are nevoie, protejând-o de ofilire sau de putrezirea rădăcinilor cauzată de udarea excesivă.

Descriere generală

Proiectul automatizează sistemul de irigare a plantelor folosind 3 moduri principale de funcționare, oferind utilizatorului flexibilitate maximă în întreținerea plantelor:

- **Modul Automat:** Sistemul se bazează strict pe senzorul de umiditate din sol. Utilizatorul setează un prag minim din butoane, iar dacă pământul se usucă și scade sub această valoare (de exemplu sub 40%), pompa pornește singură până când planta primește suficientă apă.
- **Modul Cronometrat:** Aici senzorul este ignorat, iar irigarea se face pe bază de timp. Din meniul de pe ecranul LCD, utilizatorul poate seta la ce interval de timp să pornească pompa (de exemplu, să ude o dată la 12 ore sau la 2 zile), fix ca un sistem clasic de grădinărit.
- **Modul Hibrid:** Sistemul ține cont simultan de ambele praguri (cel de umiditate și cel de timp). Dacă pământul se usucă foarte repede din cauza căldurii, pompa pornește dictată de senzor. Dacă pământul este încă relativ umed, dar a expirat timpul setat pe cronometru, pompa pornește dictată de timer. Practic, oricare dintre cele două condiții este atinsă prima, declanșează udarea plantei.

Navigarea prin aceste moduri și setarea pragurilor de timp sau umiditate se face din butoanele fizice de pe plăcuță, având mereu pe ecranul LCD un meniu clar cu starea sistemului.



Module hardware

Modulul de achiziție și monitorizare - cuprinde senzorul de umiditate și circuitele de condiționare a semnalului. Acest ansamblu are rolul de a furniza date analogice constante despre starea solului către unitatea centrală.

Modulul de alimentare și management energetic - include mufa de alimentare (DC/USB) și regulatoarele de tensiune necesare pentru a asigura pragurile de 5V pentru logică și curentul necesar pentru pompă.

Modulul de pompare și execuție - format din pompa submersibilă și etajul de putere (Driver-ul MOSFET). Acesta primește comanda de la microcontroler și acționează fizic asupra mediului prin transportul apei.

Modulul de interfață utilizator - ansamblul format din ecranul LCD și butoanele de control, care permite comunicarea bidirecțională între om și sistem.

Module software

Modulul de control central - "creierul" aplicației care implementează cele 3 regimuri (Automat, Cronometrat, Hibrid). Acesta procesează input-urile și decide starea actuatorilor folosind o mașină de stări.

Modulul interactiv de meniu - componenta software care gestionează navigarea pe LCD, afișarea datelor în timp real și preluarea comenzilor de la butoane prin tratarea întreruperilor.

Modulul de drivere periferice - cuprinde codul low-level pentru comunicarea I2C (pentru LCD), configurarea ADC-ului (pentru senzor) și generarea semnalelor de control pentru motor.

Modulul de gestiune a timpului - utilizează timerele hardware ale ATmega328P pentru a asigura precizia intervalelor de udare și pentru a gestiona resetarea automată a cronometrului în modul hibrid.

Hardware Design

Module hardware:

- **Placă de dezvoltare (Microcontroler ATmega328P):** „Creierul” sistemului. Preia datele analogice și digitale, procesează logica automatului finit (State Machine) folosind timere hardware interne și comandă atât interfața, cât și elementele de execuție.
- **Senzor umiditate sol** - utilizat pentru a măsura în timp real nivelul de hidratare al substratului plantei și pentru a trimite datele către convertorul ADC.
- **Pompă de apă (3V - 6V)** - elementul de execuție principal, responsabil pentru procesul de irigare propriu-zis.
- **Driver MOSFET (IRF520)** - folosit pentru a comanda pompa de apă, permițând controlul unor curenți mari prin semnale de joasă putere de la microcontroler (suportă și control prin PWM).

- **Display LCD 16×2 (I2C)** – utilizat pentru a afișa umiditatea curentă, modul de funcționare selectat și pentru a naviga în meniul de configurare.
- **Butoane de control** – folosite pentru interacțiunea cu utilizatorul, permițând schimbarea regimului de funcționare (Automat/Cronometrat/Hibrid) și setarea pragurilor.



Descrierea conexiunilor și a pinilor utilizați

1. Display LCD 1602 (Modul I2C)

- **PC4 (SDA) și PC5 (SCL):** Pinii hardware dedicați magistralei I2C pe ATmega328P.
- **5V și GND:** Alimentarea logică a ecranului, preluată direct de pe placa microcontrolerului.

2. Senzor de Umiditate a Solului (FC-28)

- **PC0:** Pin analogic folosit pentru a citi căderea de tensiune de pe senzor. Utilizează convertorul analog-digital (ADC) intern pentru a transforma valoarea într-un număr (0-1023).
- **PD7:** Alimentarea senzorului (VCC) nu este legată permanent la 5V, ci este controlată prin acest pin. Senzorul este pornit doar 10 milisecunde pe secundă (Duty Cycle 1%) exclusiv în momentul citirii, pentru a preveni distrugerea electrozilor prin fenomenul de electroliză.
- **GND:** Conectat la masa comună a sistemului.

3. Modul de Putere MOSFET (IRF520) - Control Pompă

- **PD6:** Pin digital configurat ca ieșire ce folosește hardware-ul intern (Timer0) pentru a genera un semnal Fast PWM. Acest lucru permite controlul vitezei pompei, implementarea unui efect de "Soft-Start" (accelerare treptată) și dozajul volumetric precis al apei.
- **VCC și GND (Logic):** Alimentarea de 5V preluată de pe placă pentru referința tranzistorului și aprinderea LED-ului indicator de stare.
- **VIN / GND_IN (Borne șurub):** Conectate la o sursă de alimentare separată dedicată pompei, cu masa (GND) pusă în comun cu placa de dezvoltare.
- **V+ / V- (Borne șurub):** Traseul prin care se livrează tensiunea către pompa submersibilă de apă.

4. Blocul de Butoane (Meniu, Up, Down)

- **PD2 (INT0) - Buton Meniu:** Pin digital ales specific deoarece suportă întreruperi externe hardware directe. Permite ieșirea instantanee din sleep sau din bucle lungi la apăsare.
- **PD3 (INT1) - Buton UP:** Pin ales din același motiv, având vector de întrerupere hardware dedicat (INT1).
- **PD4 (PCINT20) - Buton DOWN:** Pin ales pentru a menține toate butoanele pe același port (Portul D). Deoarece nu are un INT dedicat, se folosește facilitatea de Pin Change Interrupt (PCINT).
- **Conexiunea la masă:** Toate butoanele fac contact direct la **GND** atunci când sunt apăstate, eliminând necesitatea rezistențelor externe prin utilizarea modulelor de Pull-Up interne ale microcontrolerului.

Software Design

Mediul de dezvoltare:

- Aplicația a fost dezvoltată folosind **PlatformIO** (Visual Studio Code).
- S-a utilizat toolchain-ul standard **avr-gcc** pentru compilarea codului C++.

Librării și surse 3rd-party:

- S-au folosit librăriile standard din `avr-libc`: `<avr/io.h>` (pentru regiștri), `<avr/interrupt.h>` (pentru vectorii de întreruperi) și `<stdlib.h>`.
- Comunicarea I2C (TWI) și protocolul de trimitere a pachetelor de 4 biți către display-ul LCD au fost implementate direct pe regiștri.
- S-a eliminat complet utilizarea funcțiilor blocante din `<util/delay.h>`, trecându-se la o arhitectură 100% Non-Blocking bazată pe timere hardware.

Algoritmi și structuri implementate:

- **Arhitectură Non-Blocking (10ms System Tick):** „Inima” sistemului este **Timer1**, configurat în modul CTC pentru a genera o întrerupere la exact 10 milisecunde. Această variabilă globală acționează ca un ceas al sistemului de operare, permițând rularea multitasking-ului cooperativ fără a opri procesorul.
- **Debounce Software Asincron:** Citirea butoanelor se face prin întreruperi externe (INT0, INT1, PCINT2), care doar setează niște flaguri. Tratarea lor se face în bucla principală folosind un filtru de timp (20 de ticks = 200ms) raportat la ceasul sistemului, ignorând vibrațiile mecanice fără a utiliza funcții de delay.
- **Control Volumetric și Soft-Start (PWM):** Udarea nu se face pe bază de timp brut, ci volumetric (ex: 50 ml). S-a implementat o constantă a debitului pompei, iar sistemul calculează dinamic fracțiunile de 10ms necesare pentru cantitatea dorită. Suplimentar, s-a utilizat Timer0 pe pinul PD6 (Fast PWM) pentru a crea o rampă de accelerare a motorului (Soft-Start), eliminând presiunea bruscă de pe furtun.
- **State Machine (UI & Operare):** Logica principală este structurată pentru a comuta între 4 ecrane de interfață (**Monitorizare Acasă, Setare Prag Sol, Setare Timp Udare, Setare Volum Apă**) și 3 moduri operaționale (**Auto, Timed, Hybrid**), garantând că display-ul este actualizat (refresh) doar atunci când apar modificări sau o dată pe secundă, economisind timp de procesare pe magistrala lentă I2C.
- **Monitorizare și Debugging (USART):** Pe parcursul dezvoltării, s-a implementat comunicarea serială prin interfața USART. Aceasta a fost esențială pentru diagnosticarea în timp real a sistemului (monitorizarea valorilor brute citite de convertorul ADC de la senzorul de sol, verificarea stărilor din automatul finit și validarea calculelor matematice pentru dozajul volumetric), permițând o separare clară și rapidă a problemelor de tip hardware de cele software.

Rezultate Obținute



Rezultatul final este un sistem de irigații automatizat stabil și complet portabil, fiind alimentat independent de la o baterie externă. Sistemul monitorizează precis umiditatea solului, livrează volumetric cantitatea exactă de apă prin accelerarea treptată a pompei (Soft-Start), rulează o arhitectură non-blocking ce previne înghețarea codului și oferă utilizatorului o interfață fluidă, cu feedback instantaneu pe ecranul LCD.

Un video demonstrativ se poate găsi în secțiunea de jurnal, unde este demonstrată întreaga funcționalitate a proiectului (în afara cutiei).

Concluzii

Dezvoltarea acestui proiect a reprezentat o oportunitate excelentă de a transpune conceptele teoretice de microprocesoare într-o aplicație practică de utilitate reală. Cea mai importantă reușită a proiectului a fost implementarea unei arhitecturi software complet non-blocking (multitasking cooperativ bazat pe timere hardware), care oferă sistemului un timp de răspuns instantaneu la interacțiunea cu utilizatorul.

Provocări întâmpinate și rezolvate: Pe parcursul dezvoltării, interfațarea motorului de curent continuu al pompei cu partea logică a ridicat două probleme majore, ambele rezolvate prin decizii tehnice specifice:

- **Izolarea alimentării:** Inițial, alimentarea pompei direct din linia de 5V a microcontrolerului cauza

căderi bruște de tensiune la pornirea motorului (din cauza curentului mare de start), ceea ce destabiliza complet funcționarea sistemului. Soluția a fost separarea planurilor de putere: pompa este acum alimentată dintr-un pachet dedicat de baterii externe, păstrând doar masa (GND) comună cu placa logică pentru referința semnalului PWM.

- **Gestionarea interferențelor:** La oprirea bruscă a pompei se genera tensiune de autoinducție, iar parazitii electromagnetici corupeau memoria RAM a ecranului. Problema a fost mascată din software: s-a implementat o rutină asincronă de Auto-Recovery care detectează oprirea pompei și forțează o reinițializare rapidă a ecranului (sub 60ms), restabilind interfața fără blocaje perceptibile.

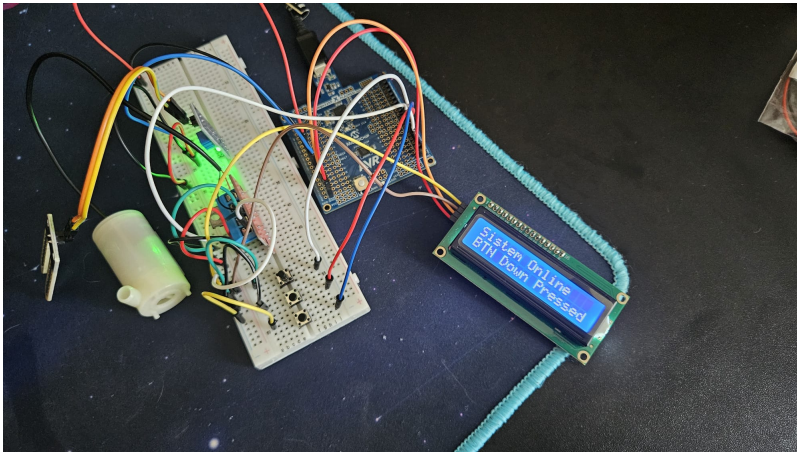
Download

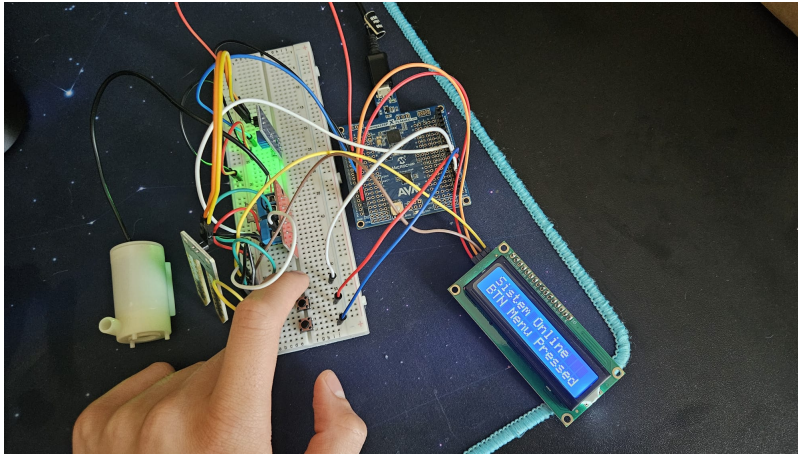
[Github Proiect](#)

Jurnal

16.05.2026

- Pentru etapa de hardware am făcut un cod provizoriu ca să demonstrez că toate componentele sunt funcționale: Cele 3 butoane afișează pe LCD când sunt apăstate. Senzorul trimite o valoare citită la USART odată la 30 de secunde, iar pompa poate fi activată din USART prin comanda "on".



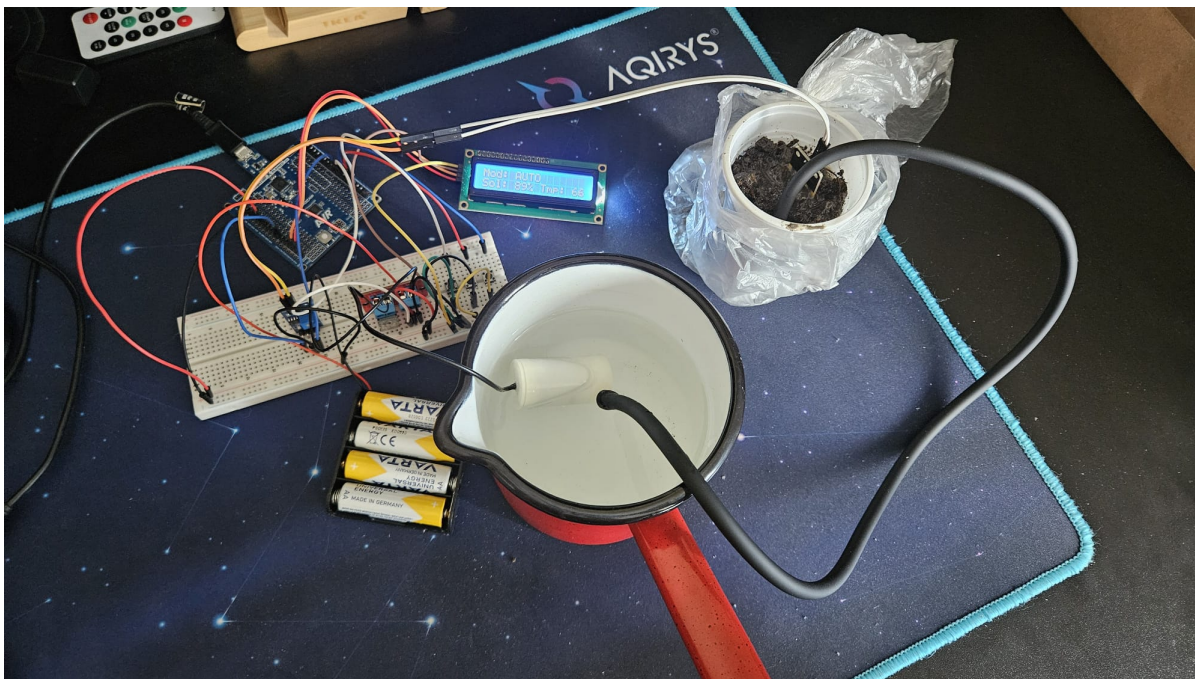


<https://youtu.be/5-j5IsSoZKQ>

Video pentru demonstrarea funcționalității pompei și a senzorului (vizibil pe USART - 1022/1023 senzorul fiind lăsat liber)

23.05.2026

- Pentru etapa software am implementat codul final (mai prezintă mici bug-uri ce vor fi rezolvate). Am schimbat alimentarea pompei trecând la o sursă separată de curent pe 4 baterii (6V). Am schimbat VCC-ul senzorului la pinul PD7 pentru a reduce consumul de curent și pentru a evita efectul de electroliză care poate degrada senzorul.



https://youtu.be/iVb_wFHDsnE

Video care demonstrează toate funcționalitățile implementate în sistem. În clip sunt prezentate: navigarea prin interfață pentru selectarea modului de funcționare, configurarea pragurilor (timp și umiditate) și setarea volumului de apă dorit. De asemenea, este surprinsă declanșarea pompei (evidențiind efectul de accelerare treptată prin Soft-Start), inițiată atât pe baza temporizatorului, cât și a atingerii pragului critic de umiditate (pentru a simula un sol uscat și a forța declanșarea sistemului, senzorul a fost extras intenționat din pământ în timpul demonstrației).

Bibliografie/Resurse

1. Documentație Hardware (Datasheets):

- **Microchip ATmega328P Datasheet** – Pentru configurarea la nivel de regiștri a Timerelor (0 și 1), a convertorului analog-digital (ADC), a magistralei I2C (TWI) și pentru a utiliza pinii corecți.
- **Hitachi HD44780 LCD Controller** – Utilizat pentru a înțelege timpii de execuție (delay-urile necesare la inițializare) și setul de comenzi pe 4-biți.
- **NXP PCF8574 I2C Expander** – Documentația modulului de pe LCD, necesară pentru implementarea protocolului de comunicare.
- **Vishay IRF520 MOSFET** – Specificațiile tranzistorului de putere folosit pentru controlul pompei de apă.

2. Documentație Software și Mediu de Dezvoltare:

- **AVR C Runtime Library (avr-libc)** – Documentația oficială pentru bibliotecile folosite în proiect (<avr/io.h>, <avr/interrupt.h>, <avr/wdt.h>).

3. Resurse Academice:

- **OCW UPB** - Proiectare cu Microprocesoare – Suportul de curs și laboratoare (în special laboratoarele de Timere, ADC și I2C) care au stat la baza dezvoltării arhitecturii acestui proiect.

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/amarinescu0804>



Last update: **2026/05/24 19:34**