

Leafy (Tamagotchi pentru plante)

Introducere

Leafy este un sistem interactiv de monitorizare a sănătății plantelor, inspirat de Tamagotchi.

Dispozitivul citește în permanență doi parametri esențiali ai unei plante: umiditatea solului și nivelul de lumină; și le traduce într-un limbaj vizual pe care oricine îl înțelege imediat: un avatar animat pe un ecran OLED care zâmbeste când planta este fericită, plânge când îi este sete, i se văd doar ochii când stă la întuneric, și emite un sunet de atenționare dacă planta este neglijată mai mult timp.

Ideea a pornit de la o problemă simplă: majoritatea plantelor nu mor din lipsă de dragoste, ci din lipsă de atenție la momentul potrivit. În loc să verifici manual pământul din ghiveci sau să îți setezi remindere pe telefon, Leafy îți comunică singur starea plantei.

Proiectul poate fi util atât pentru persoanele care uită să ude plantele, cât și ca instrument educațional, transformând îngrijirea unei plante într-o experiență interactivă, de tip joc.

Descriere generală

Proiectul este structurat în trei module principale care interacționează prin intermediul microcontrollerului ATmega328P de pe placa Xplained Mini.

Modulul de intrare (senzori)

Doi senzori colectează date din mediul fizic al plantei: Senzor capacitiv de umiditate sol — trimite un semnal analogic proporțional cu cantitatea de apă din pământ, citit prin ADC0. Fotorezistență (LDR) + rezistență de 10kΩ — formează un divizor de tensiune al cărui punct median este citit prin ADC1, indicând intensitatea luminii ambiante. 1-2 butoane — permit interacțiunea cu meniul (de exemplu, schimbarea ecranului afișat sau resetarea unei alerte).

Unitatea de procesare

ATmega328P (placa Xplained Mini) îndeplinește toate funcțiile de calcul:

- Citește senzorii analogici prin convertorul ADC intern (10 biți).
- Compară valorile citite cu praguri predefinite și determină „starea de spirit” a plantei (fericită, neutră, tristă, în alertă).
- Generează frame-urile de animație ale avatarului și le trimite ecranului prin protocolul I2C.
- Comandă buzzerul prin PWM atunci când unul dintre parametri coboară sub nivelul critic.

Modulul de ieșire

- **Ecran OLED SSD1306 (128x64 px, I2C):** afișează avatarul Leafy în diferite stări (vesel, neutru, trist).
- **Buzzer piezo:** emite tonuri de atenționare prin PWM când planta are nevoie urgentă de apă sau lumină.

Fluxul de date

Senzorii trimit semnale analogice → ADC-ul microcontrollerului le convertește în valori digitale → firmware-ul evaluează starea plantei pe baza pragurilor configurate → ecranul afișează animația corespunzătoare → dacă situația este critică, buzzerul emite o alertă sonoră.



Hardware Design

Listă de componente

Componentă	Rol în proiect
ATmega328P Xplained Mini	Unitatea centrală de procesare
Ecran OLED SSD1306 (128x64, I2C)	Afișarea avatarului și a informațiilor de stare
Senzor capacitiv de umiditate sol	Citirea nivelului de apă din pământ (ieșire analogică)
Fotorezistență (LDR)	Detectarea intensității luminii ambiante
Buzzer piezo pasiv	Alerte sonore generate prin PWM
1-2 butoane (push-button)	Interacțiunea utilizatorului cu meniul

Conectare și scheme electrice (principiu)

- **Senzor umiditate:** VCC → 5V, GND → GND, OUT → ADC0 (PC0).
- **Circuit LDR:** LDR și rezistența de 10 kΩ conectate în serie între 5V și GND; nodul comun (mijlocul divizorului) → ADC1 (PC1). Când lumina crește, rezistența LDR scade și tensiunea pe nod crește.
- **Ecran OLED SSD1306:** VCC → 3.3V sau 5V (conform specificațiilor modulului), GND → GND, SDA → PC4 (SDA), SCL → PC5 (SCL).
- **Buzzer piezo pasiv:** un terminal → PB1 (OC1A, capabil de PWM), celălalt terminal → GND. Frecvența PWM dictează înălțimea sunetului.
- **Butoane:** un terminal → pin digital (ex. PD2, PD3), celălalt → GND; rezistențele interne de pull-up ale microcontrollerului sunt activate prin software (INPUT_PULLUP).

===== Software Design =====

Descrierea codului aplicației (firmware):


mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR) librării și surse 3rd-party (e.g. Procyon AVRlib) algoritmi și structuri pe care plănuieți să le implementați (etapa 3) surse și funcții implementate

===== Rezultate Obținute =====

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

==== 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 =====

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

==== Bibliografie/Resurse =====

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

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/darian_petre.duinea 

Last update: **2026/05/09 20:05**