

MimiPet: Companion digital interactiv

Introducere

MimiPet este un sistem autonom de divertisment care aduce la viață un companion digital interactiv. Ce face: Dispozitivul simulează ciclul de viață și stările emoționale ale unui personaj virtual. Acesta este afișat pe un ecran OLED de 128×64 pixeli și își schimbă comportamentul în timp real. MimiPet, prin intermediul senzorilor, poate detecta dacă în cameră este lumină sau întuneric, reacționează la zgomote puternice (cum ar fi bătăile din palme) și comunică cu utilizatorul prin expresii faciale animate și semnale acustice. Utilizatorul poate avea grija de MimiPet prin intermediul butoanelor, care oferă funcționalități precum hranire, somn și joacă.

Descriere generală



Proiectul este structurat pe o arhitectură hardware-software integrată, unde microcontrolerul ATmega328P acționează ca punct central de control.

Componentele principale ale sistemului:

Unitatea Logică (Logic Engine): Este nucleul software care rulează pe microcontroler. Acesta primește datele brute de la senzori (prin ADC și GPIO), calculează starea de spirit a animalului și trimite comenzi către restul modulelor.

Sistemul de Input (Senzori):

- **Senzorul de Lumină (LDR):** determină ciclul zi/noapte.
- **Senzorul de Sunet (KY-037):** permite interacțiunea prin sunete ambientale.
- **Butoanele (K1-K4):** oferă control manual pentru acțiuni precum hrănire și joacă.
- **Potențiometrul:** permite reglarea fină a parametrilor jocului.

Sistemul de Output (Feedback):

- **Graphics Driver:** gestionează afișarea animațiilor pe ecranul OLED prin protocolul I2C.
- **Sound Engine:** generează tonuri și melodii prin semnale PWM trimise către buzzer.
- **LED-urile de stare:** oferă indicatoare vizuale rapide pentru alerte critice.

Interacțiunea dintre module: Interacțiunea dintre module este bidirecțională la nivel logic: senzorii modifică starea internă a programului, iar programul actualizează imediat feedback-ul vizual și sonor pentru utilizator.

Hardware Design

Lista componentelor hardware utilizate:

- Microcontroler ATmega328P
- Display OLED 0.96" (SSD1306)
- Senzor de Sunet KY-037
- Fotorezistență (LDR)
- Buzzer Pasiv
- Potentiometru 10k Ω
- 4 Butoane Tactile
- Breadboard și fire Dupont
- Rezistențe de 220 Ω și 10k Ω
- LED-uri

Tabelul de mai jos prezintă alocarea pinilor microcontrolerului și justificarea alegerii fiecărui pin:

Pini I2C (comunicare display OLED):

Pin 27 (PC4/SDA) — linia de date I2C. Ales pentru că este pinul hardware TWI dedicat al ATmega328P.
Pin 28 (PC5/SCL) — linia de ceas I2C. Pin hardware TWI dedicat.

Pini ADC (intrări analogice):

Pin 23 (PC0/ADC0) — citește tensiunea din voltage divider-ul LDR + rezistor 10 k Ω . Valori ADC mari indică lumina puternică, valori mici indică întuneric.

Pin 24 (PC1/ADC1) — citește ieșirea analogică (A0) a senzorului KY-037 pentru nivelul sonor ambiental.

Pin 25 (PC2/ADC2) — citește poziția cursorului potentiometrului (valori 0-1023).

Pini digitali cu funcții speciale:

Pin 4 (PD2/INT0) — ieșirea digitală (D0) a senzorului KY-037. Ales specific pentru funcția de întrerupere externă INT0, permițând detectarea zgometelor fără polling continuu.

Pin 5 (PD3/OC2B) — semnalul de control al buzzer-ului pasiv. Ales pentru ieșirea PWM a Timer2, necesară generării tonurilor de frecvență variabilă.

Pini digitali pentru butoane (intrări cu pull-up intern):

Pin 6 (PD4) — butonul K4 de pe modulul OLED

Pin 11 (PD5) — butonul K3

Pin 12 (PD6) — butonul K1

Pin 13 (PD7) — butonul K2

Butoanele sunt configurate ca intrări cu rezistența de pull-up internă activată prin registrul PORTD. Apăsarea unui buton trage pinul la LOW.

Pini digitali pentru LED-uri (ieșiri):

Pin 14 (PB0) — LED verde, prin rezistor de limitare 220 Ω

Pin 15 (PB1) — LED roșu, prin rezistor de limitare 220 Ω

Pini de alimentare:

Pin 7 (VCC) și Pin 20 (AVCC) — alimentare la 5V. AVCC este conectat la VCC pentru funcționarea ADC-ului.

Pin 8 (GND) și Pin 22 (GND1) — masă.

Schema electrica

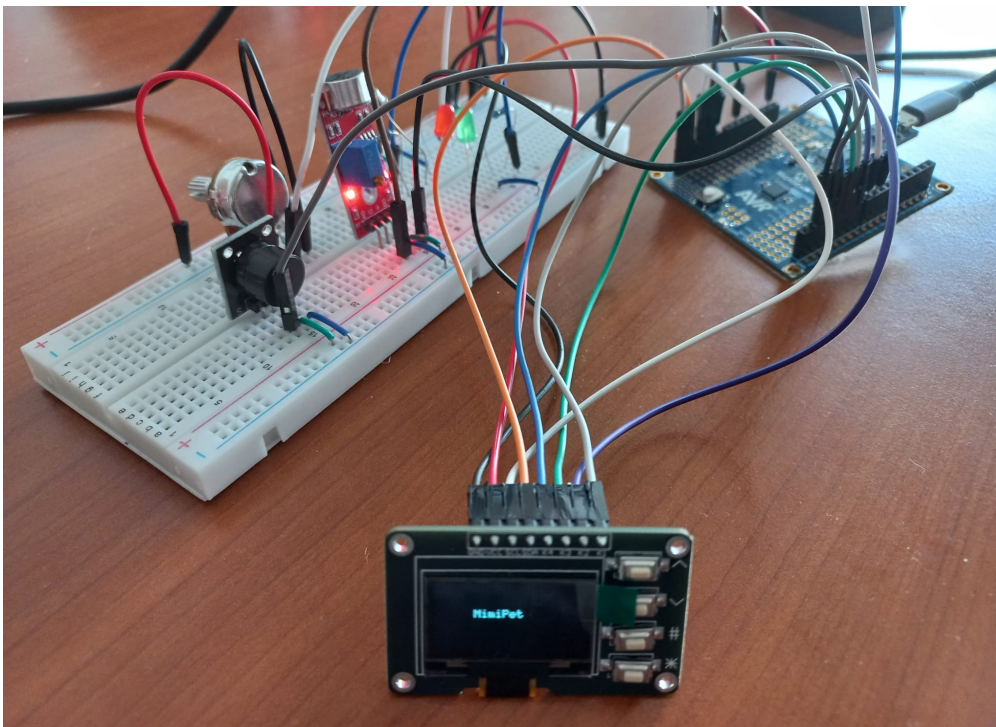
Schema electrica prezinta conexiunile complete dintre ATmega328P si toate componentele periferice. Modulele prefabricate (display OLED cu butoane, senzor KY-037, buzzer pe PCB) sunt reprezentate prin headere cu pinii etichetati.

Alimentarea la 5V si masa sunt distribuite prin sinele breadboard-ului, conectate la pinii VCC si GND ai placii ATmega328P. Comunicarea cu display-ul OLED se realizeaza prin protocolul I2C pe pinii hardware dedicati (SDA si SCL). Fotorezistenta formeaza un voltage divider cu rezistorul de 10 kOhm, iar punctul de mijloc este citit pe canalul ADC0.



Dovada functionare cel puțin o componenta

Pentru validarea display-ului OLED, s-a realizat mai intai un scan al bus-ului I2C care a confirmat prezenta dispozitivului la adresa 0x3C. Ulterior, s-a scris un driver minimal pentru controlerul SSD1306, lucrând direct pe registrele TWI (Two Wire Interface) ale ATmega328P, fara a folosi biblioteci externe. Programul initializeaza display-ul prin secventa standard de comenzi SSD1306 (configurare clock, multiplex, charge pump, mod de adresare orizontala) si apoi scrie caractere pe ecran folosind un font 5x7 definit manual in cod. Testul a constatat in afisarea textului "MimiPet" centrat pe ecran, confirmand functionarea corecta a comunicarii I2C si a display-ului.



Software Design

Stadiul actual al implementarii software

Proiectul a atins un stadiu functional complet si robust. Arhitectura software este bazata pe un automat finit (FSM - Finite State Machine) complet non-blocant. Am implementat logica de "metabolism" pentru 3 nevoi distincte (Foame, Joaca, Somn) care cresc prin interactiune si scad in timp real cu viteze diferite. Afisajul OLED a fost programat la nivel de registru, incluzand generarea matematica a interfeței grafice (barele de status si literele H, P, S). Sensorii sunt integrati deplin: senzorul de lumina dispune de o rutina de calibrare automata, iar senzorul de sunet foloseste intreruperi hardware externe pentru reactie instantanee.

Motivarea alegerii bibliotecilor in cadrul proiectului

Pentru a intelege in profunzime functionarea microcontrolerului ATmega328P si pentru a optimiza spatiul de memorie, am luat decizia de a NU folosi biblioteci externe complexe (precum librariile clasice pentru OLED). Am utilizat exclusiv bibliotecile standard AVR (`<avr/io.h>`, `<avr/interrupt.h>`, `<util/delay.h>`). Driverul I2C, secventa de initializare si sistemul de desenare pe display sunt scrise integral de la zero, manipuland direct registrele hardware (TWI).

Elementul de noutate al proiectului

Spre deosebire de un joc "Tamagotchi" clasic controlat doar prin butoane, elementul de noutate al lui MimiPet consta in constientizarea mediului ambient. Animăluțul reacționează organic la mediul utilizatorului: intra singur in starea de somn cand lumina din camera se stinge si se trezeste instantaneu la sunete ascutite (ex: batai din palme). In plus, elementele de UI (barele de nevoi) nu sunt pre-randate ca imagini statice, ci sunt calculate si desenate procedural din cod la fiecare frame, o abordare inedita pentru economisirea memoriei Flash.

Justificarea utilizarii functionalitatilor din laborator

- **Protocolul I2C (TWI):** Utilizat pentru transferul in bloc al cadrelor de animatie catre ecranul OLED.
- **Convertorul Analog-Digital (ADC):** Folosit pentru a citi in timp real valorile fotorezistentei (pentru detectia luminii) si ale potentiometrului (utilizat pentru a regla dinamic luminozitatea ecranului OLED, fara a rescrie codul).
- **Intreruperi Externe (ISR pe INT0):** Esentiale pentru senzorul de sunet KY-037. Zgomotele scurte sunt citite imediat, deoarece intreruperea opreste microcontrolerul din operatiunile lente (cum ar fi desenarea pe ecran) pentru a asigura trezirea animăluțului.
- **Timere si logica Non-Blocanta:** Logica de timp (similara cu `millis()` din Arduino) a inlocuit functiile blocante de tip `delay`. Astfel, animatiile, scaderea nevoilor (metabolismul) si citirea senzorilor ruleaza in paralel.

Scheletul proiectului, interactiunea si validarea

Arhitectura se bazeaza pe o bucla infinita `while(1)` care gestioneaza o masina de stari cu 5 ramuri (IDLE, EATING, PLAYING, SLEEPING, HUNGRY). **Interactiunea:**

1. **Sistemul de Input:** Sensorii, intreruperile si butoanele modifica variabilele globale de stare si nivelurile barelor (0-100).
2. **Metabolismul:** Un bloc de cod scade automat barele in functie de trecerea timpului (ex: Foamea scade cel mai repede, Somnul cel mai greu).
3. **Sistemul de Override:** Daca orice bara ajunge la 0, sistemul "forteaza" starea de tristete (HUNGRY) si aprinde un LED rosu de avertizare.
4. **Sistemul de Output:** Preia starea curenta si redeseneaza ecranul doar cand este necesar, declansand concomitent sunete corespunzatoare in buzzer.

Validarea a fost realizata modular: s-a testat mai intai comunicarea I2C "bare-metal", apoi s-au calibrat senzorii vizualizand datele pe interfata UART (Serial Monitor), iar la final s-a validat integrarea tuturor modulelor in masina de stari prin testarea edge-case-urilor (ex: apasarea butoanelor in timpul somnului).

Calibrarea elementelor de senzistica

- **Senzorul de lumina (LDR):** S-a implementat o rutina de "Auto-Calibrare" in software. La pornirea placutii, sistemul masoara lumina ambientala din primele secunde, calculeaza o medie si stabileste dinamic pragurile de intuneric si lumina. Astfel, dispozitivul functioneaza optim in orice incapere.
- **Senzorul de sunet (KY-037):** Calibrarea s-a facut hardware, reglati potentiometrul multitura integrat pe modul. Sensibilitatea a fost ajustata pentru a ignora conversatiile si zgomotul de fond, dar pentru a trimite un semnal LOW logic la un stimul auditiv puternic si scurt (bataie din palme la 20cm).

Optimizari realizate (Cum, de ce si unde)

- **Generarea Procedurala a UI-ului (Memorie):** In loc sa stocam zeci de frame-uri cu bare de diferite inaltimi in memoria program, barele si literele (H, P, S) sunt calculate bit cu bit printr-o functie ajutatoare si desenate suprapus pe pagina OLED. Aceasta optimizare a redus masiv ocuparea memoriei Flash.
- **Eliminarea blocajelor (Performanta):** Inlocuirea tuturor functiilor `_delay_ms()` cu evaluari de tip timestamp (`current_time - last_time`) in bucla principala. Astfel, animatiile sunt fluide, iar microcontrolerul ramane mereu receptiv la input-urile de la ADC.
- **Protectia variabilelor concurente (Stabilitate):** Deoarece starea animalutului poate fi modificata atat de bucla principala, cat si asincron de intreruperea hardware de la microfon, variabilele critice au fost declarate cu modificatorul `volatile`. Asta a prevenit optimizarile agresive ale compilatorului C si a eliminat bug-urile in care sistemul parea sa ignore anumite input-uri.

Demo Video

<https://youtu.be/OmsvD13Wltw?is=IjdXu4bG6vpTGldU>

Concluzii

Dezvoltarea proiectului MimiPet a fost o experienta de invatare extrem de valoroasa, care mi-a permis sa transform conceptele teoretice de sisteme embedded intr-un dispozitiv interactiv si complet functional. Desi scrierea driverelor direct pe registre si implementarea unui automat finit non-blocant au reprezentat provocari tehnice reale, aceste etape m-au ajutat sa inteleg in profunzime arhitectura microcontrolerului ATmega328P. In final, sunt foarte mandra ca am reusit sa dau viata unui companion digital autonom care raspunde organic la mediul inconjurator, demonstrand cat de mult potential poate fi scos dintr-un hardware cu resurse limitate printr-o optimizare software corecta.

Jurnal

- 25.04.2026 - alegerea temei proiectului si a componentelor necesare.
- 26-28.04.2026 - achizitionarea componentelor.
- 08.05.2026 - inceputul redactarii documentatiei, descrierea generala a proiectului.
- 14-16.05.2026 - crearea schemei electrice, verificarea functionalitatii a componentelor.
- 16-24.05.2026 - lucrul asupra partii de software.

Bibliografie/Resurse

[Datasheet ATmega328P Xplained Mini](#)

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/cezar.zlatea/otilia.rudnic>



Last update: **2026/05/25 01:33**

