

Pet Feeder

Introducere

Proiectul constă în realizarea unui sistem mecatronic capabil să elibereze porții de hrană uscată pentru animalele de companie la intervale de timp prestabilite de utilizator. Sistemul monitorizează ora exactă și acționează un mecanism de dozare mecanic pentru a automatiza procesul de hrănire.

Scopul principal este crearea unui dispozitiv fiabil care să asigure nutriția animalelor de companie chiar și atunci când stăpânii nu sunt acasă, eliminând stresul legat de programul fix de masă.

Consider că acest proiect este util deoarece îmi oferă posibilitatea de a aplica practic noțiunile studiate în laborator într-un dispozitiv interactiv și vizual, care rezolvă o problemă reală de automatizare. Proiectul demonstrează cum, folosind resurse hardware limitate specifice sistemelor embedded, se poate crea un produs de tip „Smart Home” capabil să gestioneze sarcini critice în timp real, oferind în același timp un beneficiu direct utilizatorilor prin asigurarea unui regim alimentar controlat pentru animalele de companie.

Descriere generală

Proiectul este organizat în trei paliere principale care interacționează bidirecțional:

- Modulul de Intrare (Interfața):

Cuprinde butoanele de control și senzorul de timp (RTC). Acestea furnizează microcontrolerului datele necesare pentru programare și referința temporală precisă.

- Unitatea Centrală (ATmega328P):

Reprezintă nucleul software. Gestionează logica meniului, stochează setările în memoria permanentă (EEPROM) și decide momentul declanșării hrănirii prin compararea timpului curent cu cel programat.

- Modulul de ieșire (Execuția):

Include ecranul LCD pentru feedback vizual, buzzer-ul pentru alertă sonoră și servomotorul care acționează mecanismul fizic de distribuție a hranei.



Hardware Design

Bill of Materials

Componentă	Cantitate	Link	Preț unitar (RON)
Placă dezvoltare Arduino UNO	1	Link	30.93
Display LCD SPI	1	Link	13.99
Modul interfață I2C	1	Link	9.99
Buzzer pasiv	1	Link	2.99
Condensator electrolitic 1000μF	2	Link	5.08
Condensator ceramic 0.1uF	2	Link	6.30
Servomotor MG995	1	Link	44,53
Modul RTC DS3231	1	Link	14,74
Set fire Jumper	1	Link	11,72
Preț total			140.02 RON

Funcționalitatea Componentelor

Componentă	Rol în proiect
LCD 1602	Afișează instrucțiunile, feedback-ul pentru cod și rezultatul accesului
Butoane push	Îți permit să navighezi prin meniu, să schimbi orele de hrănire sau să dai o porție la cerere
Buzzer Pasiv	Scoate un sunet scurt înainte de eliberarea hranei și în timpul setărilor
Servomotor MG995	Învârte axul pentru a împinge bobitele de mâncare afară din rezervor
Modul RTC	Ține minte ora exactă (oră, minut, secundă) chiar și când aparatul este scos din priză
Condensator	Stocază energie și o eliberează rapid atunci când servomotorul pornește brusc
Adaptor Sursă 5V/2A	Transformă curentul de la priză în curent continuu sigur de 5V pentru tot circuitul

Schema Electrică



Pinout

Componentă	Pin Arduino	Tip Pin	Utilitate
LCD 1602	A4 (SDA)	Date I2C	Transmite datele text către ecran
LCD 1602	A5 (SCL)	Clock I2C	Sincronizează transmisia datelor între Arduino și LCD
Buton UP	3	Digital	Crește valoarea orelor/minutelor în meniul de setare
Buton DOWN	4	Digital	Scade valoarea orelor/minutelor în meniul de setare
Buton MENU	2	PCINT	Schimbă starea programului (afișare oră / setare mese)

Servomotor MG995	9	PWM	Controlează unghiul și mișcarea precisă a spiralei de hrană
Modul RTC	A4 (SDA)	Date I2C	Transmite datele de timp (oră, minut) către Arduino
Modul RTC	A5 (SCL)	Clock I2C	Sincronizează citirea timpului de la cipul de ceas
Buzzer pasiv	8	Digital	Trimite semnalul sonor
Linie de Tensiune	5V	Power	Distribuie tensiunea de 5V către magistrala de alimentare pozitivă a componentelor
Referință (GND)	GND	Power	Închide circuitul pentru toate componentele

Software Design

Proiectul va fi implementat folosind limbajul C și va utiliza perifericele hardware ale microcontrolerului pentru gestionarea procesului de automatizare:

- Interfața I2C pentru sincronizarea cu modulul de ceas (RTC) și transmiterea datelor către afișajul LCD.
- PWM (Pulse Width Modulation) pentru controlul precis al servomotorului MG995 și generarea alertelor sonore prin buzzer.
- GPIO (Input) pentru preluarea comenzilor de la butoane și navigarea în meniul de configurare.
- Memoria EEPROM pentru salvarea non-volatilă a programului de hrănire, prevenind pierderea datelor la căderi de tensiune.
- Timere hardware pentru monitorizarea timpului de funcționare a motorului (dozarea porției) și filtrarea semnalului butoanelor (debouncing).
- USART/Serial pentru calibrarea senzorilor și depanarea logicii de comparare a timpului în faza de testare.

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,

Last update:

2026/05/16 13:20 pm:prj2026:victor.stoica0203:robert.predeleanu <http://ocw.cs.pub.ro/courses/pm/prj2026/victor.stoica0203/robert.predeleanu>

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/victor.stoica0203/robert.predeleanu> 

Last update: **2026/05/16 13:20**