



Sistem automat de hranire pentru animale de companie

~Arduino Uno~

Student: Berciu Oana-Georgiana

Profesor: Rednic Ana

Cuprins:

- 1.Introducere : scop si motivatie
2. Cercetare bibliografică și soluții existente
3. Solutia propusa si implementarea
 - 3.1 Descriere tehnica
 - 3.2 Descrierea teoretica a algoritmului
 - 3.3 Implementare
 - 3.3.1 Implementare Hardware
 - 3.3.2 Implementarea Software
 - 3.3.3 Debuging
4. Testare
 - 4.1 Probleme intampinate in timpul implementarii
 - 4.2 Calibrare
 - 4.3 Evolutia proiectului
- 5.Concluzie

1.Introducere : scop si motivatie

Automatizarea proceselor de zi cu zi devine din ce în ce mai importantă, inclusiv în domeniul îngrijirii animalelor de companie. Unul dintre aspectele esențiale atunci când decizi să crești un animal este respectarea unui program regulat de hrănire. În lipsa proprietarului, acest lucru poate deveni dificil de realizat în mod constant și precis.

Scopul acestui proiect este realizarea unui sistem automat de hranire pentru animale de companie, capabil să distribuie hrana la o ora prestabilită, fără intervenția utilizatorului în momentul hrănirii. Sistemul este construit în Arduino Uno și utilizează componente hardware (servo-motor, modul RTC, tastatura, afisaj LCD).

Motivatia alegerii acestui proiect consta in dorinta de a imbina notiuni de electronica, programare embedded și interacțiune om-mașina, într-o aplicație practică și utilă. Deși există soluții comerciale pentru hrănirea automată a animalelor, implementarea proprie oferă flexibilitate, cost redus și o mai bună înțelegere a funcționării sistemelor embedded.

Proiectul nu își propune să reinventeze un concept complet nou, ci să implementeze o soluție existentă într-o manieră simplă, robustă și ușor de extins, adaptată cerințelor unui utilizator obișnuit.

2. Cercetare bibliografică și soluții existente

Ideea unui sistem automat de hrănire pentru animale de companie nu este una complet originală, existând deja pe piață produse comerciale care implementează funcționalități similare. Pentru a fi mai ușor de implementat proiectul propus am analizat un proiect documentat public, disponibil la următorul link:

<https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino#complete-arduino-code-for-automatic-pet-feeder-programming>

Proiectul utilizează un microcontroler Arduino Uno, un modul RTC DS3231 pentru menținerea timpului, un servo-motor pentru acționarea mecanismului de distribuție a hranei, un afișaj LCD 16×2 și o tastatură matricială pentru setarea orei de hrănire. Hranirea este declansată atunci când ora curentă coincide cu ora introdusă de utilizator.

Criteriu	Soluție existentă	Proiect propus
Consum de energie	Scazut	Scazut
Dificultate implementare	Mai ridicată pentru că are mai multe conexiuni hardware	Medie (are interfața I2C și mai puține conexiuni)

Cost resurse si implementare	Mediu	Scazut
Precizia timpului	Ridicata	Ridicata -folosesc un modul RTC asemanator
Interfata utilizator	LCD +Keypad 4x4 +Buton	LCD+Keypad 4x4
Scop educational	Da	Da

Analiza arata ca solutia existenta si proiectul propus au același principiu de functionare, insa difera la nivel de optimizare si simplitate. Solutia existenta pune accent pe precizia timpului, in timp ce proiectul implementat de mine urmareste reducerea complexitatii hardware, a costului si cresterea fiabilitatii printr-o schema mai simpla si o logica software mai clara.

3. Solutia propusa si implementarea

3.1 Descriere tehnica

Arduino Uno coordoneaza toate componentele sistemului. Pentru asigurarea unei referințe corecte de timp, este utilizat un modul RTC DS1307, care pastreaza ora chiar si in cazul intreruperii alimentarii. Pe baza timpului citit de la acest modul, Arduino decide momentul declansarii procesului de hranire.

Interactiunea cu utilizatorul se realizeaza printr-un afisaj LCD 16×2 si o tastatura matriciala. Afisajul LCD este utilizat pentru a afisa starea sistemului si mesajele de ghidare, iar tastatura permite introducerea orei de hranire intr-un format simplu,selectarea tipului de portie pe care il doresti(mare sau mica),selectarea optiunii de golire automata. De asemenea,starea sistemului mai este semnalizata si prin ajutorul unui led,care se aprinde in momentul in care are loc hranirea.Dupa setare, sistemul functioneaza fara alte interventii.

Distribuirea hranei este realizata prin intermediul unui servo-motor, care actioneaza un mecanism de deschidere a recipientului. Servo-motorul este comandat de Arduino sa se rotească la un unghi prestabilit pentru o durata de timp diferita in functie de tipul portiei pe care il selectezu, permitand eliberarea unei cantitati controlate de hrana, dupa care revine la poziția initiala.

Sistemul impiedica hranirea de mai multe ori in aceeași zi. Aceasta protectie este resetata automat la inceputul unei noi zile. In ansamblu, solutia ofera o implementare simpla, fiabila si usor de extins, potrivita pentru un proiect academic de tip embedded.

3.2 Descrierea teoretică a algoritmului

Algoritmul sistemului automat de hrănire este conceput pentru a funcționa continuu, pe baza timpului real furnizat de modulul RTC și a comenzilor introduse de utilizator prin intermediul tastaturii. Sistemul rulează într-o buclă infinită și verifică permanent dacă sunt îndeplinite condițiile necesare pentru declanșarea procesului de hrănire sau a altor funcții auxiliare.

La pornirea sistemului, sunt inițializate toate componentele hardware: placa Arduino, modulul RTC, afișajul LCD, tastatura matricială, servo-motorul și LED-ul de stare. Servo-motorul este poziționat în starea inițială (închis), iar pe afișajul LCD este prezentat un mesaj care indică faptul că sistemul este pornit și pregătit pentru configurare.

În timpul funcționării normale, Arduino citește periodic ora curentă de la modulul RTC. În paralel, sistemul monitorizează tastatura pentru a detecta comenzile utilizatorului. La apăsarea tastei corespunzătoare intrării în modul de configurare, sistemul trece într-un mod dedicat de setare a orei de hrănire.

În acest mod, utilizatorul introduce ora de hrănire în format HHMM, folosind tastatura. Algoritmul verifică dacă au fost introduse exact patru cifre. Dacă formatul este corect, valorile introduse sunt convertite în valori numerice pentru oră și minut și sunt salvate în memorie. În cazul unui format incorect, utilizatorul este informat prin intermediul afișajului LCD și este solicitat să reintroducă datele.

După setarea corectă a orei de hrănire, sistemul intră într-un al doilea mod de configurare, destinat selectării porției de hrană. Utilizatorul poate alege între o porție mică și o porție mare, utilizând taste dedicate. Alegerea este salvată și confirmată printr-un mesaj afișat pe LCD. Ulterior, afișajul prezintă simultan tipul porției selectate și ora de hrănire setată, pentru confirmarea vizuală a configurației.

În modul normal de funcționare, la fiecare iterație a buclei principale, algoritmul compară ora curentă cu ora de hrănire setată. Dacă cele două valori coincid, dacă a fost selectată o porție validă și dacă animalul nu a fost hrănit în ziua respectivă, sistemul declanșează procesul de hrănire.

Procesul de hrănire constă în rotirea servo-motorului la un unghi prestabilit, corespunzător mecanismului de dozare. Durata de menținere a servo-motorului în poziția deschisă depinde de porția selectată (mică sau mare), permițând eliberarea unei cantități corespunzătoare de hrană. După finalizarea procesului, servo-motorul revine în poziția inițială. Pe durata hrănirii, LED-ul este activat ca indicator vizual, iar starea sistemului este afișată pe LCD.

Pentru a preveni hrănirea multiplă în aceeași zi, algoritmul utilizează o variabilă de protecție care este activată după finalizarea hrănirii. Această variabilă blochează

declanșarea unui nou ciclu de hrănire până la începutul unei noi zile, moment în care starea este resetată automat.

În plus, sistemul oferă o funcție de golire manuală a recipientului, activată exclusiv la apăsarea unui buton dedicat de pe tastatură. La declanșarea acestei comenzi, servo-motorul se deschide o singură dată și rămâne în poziția deschisă pentru o perioadă mai lungă de timp, permițând golirea completă a recipientului. După expirarea timpului prestabilit, servo-motorul revine la poziția inițială, iar utilizatorul este informat printr-un mesaj afișat pe LCD.

Prin această structură, algoritmul asigură o funcționare sigură, controlată și flexibilă a sistemului automat de hrănire, oferind atât funcții automate, cât și control manual, adaptate nevoilor utilizatorului.

3.3 Implementare

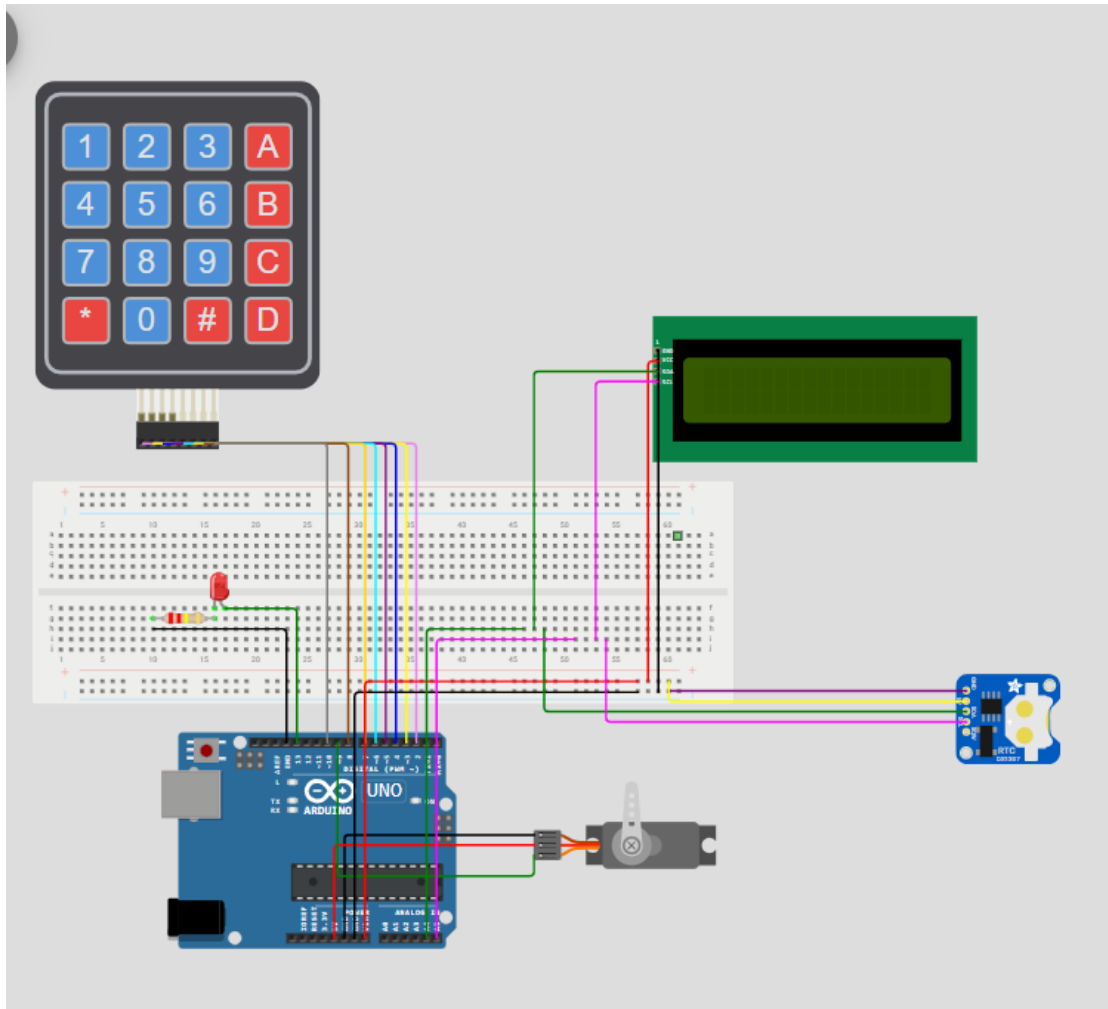
3.3.1 Implementare Hardware

Implementarea hardware a sistemului este realizată în jurul plăcii Arduino Uno, care funcționează ca unitate centrală de control. Toate componentele sunt conectate direct la Arduino, formând un sistem embedded compact și ușor de înțeles.

Componentele hardware utilizate sunt:

- Arduino Uno
- modul RTC DS1307
- afișaj LCD 16x2 cu interfața I2C
- tastatura matriceală 4x4
- servo-motor
- LED de semnalizare

Modulul RTC și afișajul LCD sunt conectate la Arduino prin magistrala I2C (pini SDA și SCL), ceea ce reduce numărul de pini utilizați și simplifică schema electrică. Tastatura matriceală este conectată la pini digitali separați pentru rânduri și coloane, permițând citirea corectă a tastelor apasate. Servo-motorul este conectat la un pin digital PWM și este utilizat pentru acționarea mecanismului de distribuție a hranei. LED-ul este conectat la un pin digital și este folosit ca indicator vizual în timpul procesului de hrănire.



3.3.2 Implementarea Software

Implementarea software este realizata folosind Arduino IDE si limbajul C/C++. Programul este structurat modular, fiecare parte avand un rol bine definit.

Functia `setup()` este responsabila de initializarea sistemului. In aceasta etapa sunt configurati pinii de intrare si iesire, este initializat modulul RTC, afisajul LCD, tastatura si servo-motorul. De asemenea, este afisat un mesaj de pornire care confirma ca sistemul este functional.

Functia `loop()` reprezinta logica principala a aplicatiei. In cadrul acesteia, Arduino citeste in mod repetat ora curenta de la modulul RTC si verifica intrarile de la tastatura. Daca utilizatorul solicita setarea orei de hranire, sistemul intra in modul de configurare. In caz contrar, se realizeaza comparatia dintre ora curenta si ora setata.

Functia `seteazaOraHranire()` este utilizata pentru introducerea orei de hranire. Aceasta functie citeste valorile introduse de utilizator prin tastatura, verifica daca formatul este corect si salveaza ora si minutul in variabile dedicate. In acelasi timp, afisajul LCD ofera feedback utilizatorului.

După setarea corectă a orei de hrănire, software-ul intră într-un mod suplimentar de configurare, gestionat de funcția `selecteazaPortie()`. Această funcție permite utilizatorului să aleagă tipul de porție (mică sau mare) prin apăsarea tastelor corespunzătoare de pe tastatură. Alegerea este salvată în memorie și confirmată atât prin intermediul afișajului LCD, cât și prin mesaje transmise prin interfața serială. Ulterior, pe afișajul LCD sunt prezentate simultan tipul porției selectate și ora de hrănire setată, pentru confirmarea finală a setărilor realizate.

Atunci când condițiile de hranire sunt îndeplinite, software-ul comanda servo-motorul să execute o mișcare controlată, permitând eliberarea hranei. După finalizare, sistemul revine în starea de așteptare.

3.3.3 Debuging

Procesul de depanare a fost realizat folosind monitorul serial din Arduino IDE și observarea comportamentului sistemului în timp real. Au fost afișate valori precum ora curentă, ora setată și starea procesului de hranire pentru a verifica corectitudinea algoritmului.

Principalele probleme întâmpinate au fost:

- sincronizarea inițială a modului RTC
- citirea corectă a tastaturii
- introducerea corectă a orei de hranire
- declansarea servo motorului la ora introdusă de utilizator

Implementarea finală este stabilă și permite funcționarea autonomă a sistemului după configurarea inițială.

4. Testare

4.1 Probleme întâmpinate în timpul implementării

Ca în orice proiect hardware, sistemul nu a funcționat corect din prima încercare. În timpul implementării au apărut mai multe probleme, care au necesitat ajustări hardware și software.

Una dintre problemele apărute în timpul implementării a fost legată de intrarea în modul de setare a orei de hranire. Inițial, în momentul în care utilizatorul apăsă tasta corespunzătoare pentru setare (tasta „*”), sistemul intra în modul de configurare, însă nu permitea introducerea completă a orei. După introducerea unei singure cifre, sistemul ieșea automat din modul de setare și revenea în bucla principală, afișând în mod repetat ora curentă citită de la modulul RTC.

Aceasta problema a fost cauzata de faptul ca citirea tastaturii si afisarea orei curente se realizau simultan in bucla principala, fara o separare clara intre modul normal de functionare si modul de setare. Astfel, logica principala intrerupea procesul de introducere a datelor.

Problema a fost rezolvata prin utilizarea unei functii dedicate pentru setarea orei, in care executia este blocata temporar pana la introducerea completa a celor patru cifre si confirmarea acestora. In acest mod, sistemul ramane in modul de setare pana la finalizarea corecta a configurarii, permitand introducerea completa a orei de hranire.

Au existat si probleme mecanice minore, legate de pozitia initiala a servo-motorului si de cantitatea de hrana eliberata, care au necesitat calibrare.

4.2 Calibrare

Pentru a obtine o functionare stabila, au fost realizate mai multe teste prin modificarea unor parametri ai sistemului si observarea comportamentului acestuia.

1. Testarea unghiului servo-motorului

Unghiul de rotatie al servo-motorului influenteaza direct cantitatea de hrana distribuita.

Unghi servo	Cantitate hrana	Observatii
45 grade	Prea mica	Mecanismul nu se deschide suficient, hrana cade partial in afara farfuriei
90 grade	Optima	Mecanismul se deschide suficient pentru ca hrana sa cada perfect in farfurie

2. Testarea duratei de activare a servo-motorului

Durata de mentinere a servo-motorului in pozitia deschisa a fost testata pentru a controla cantitatea de hrana.

Durata deschidere	Rezultat
300 ms	Cantitate hrana insuficienta
800 ms	Cantitate hrana corecta
1200 ms	Cantitate hrana excesiva

4.3 Evolutia proiectului

Starea finala a proiectului nu este identica cu prima versiune functionala. Proiectul a trecut prin mai multe etape de dezvoltare.

In prima faza, a fost testat doar Arduino impreuna cu servo-motorul, pentru a verifica miscarea mecanica. In aceasta etapa nu existau RTC, LCD sau tastatura.

In a doua faza, a fost adaugat modulul RTC, pentru a verifica citirea corecta a timpului si declansarea actiunii la o ora fixa. Ulterior, au fost integrate afisajul LCD si tastatura, ceea ce a crescut complexitatea software, dar a imbunatatit interactiunea cu utilizatorul.

În etapa finală de dezvoltare, proiectul a fost extins prin introducerea unor funcționalități suplimentare, precum selecția porției de hrană (mică sau mare) și funcția de golire manuală a recipientului, activată la apăsarea unui buton. De asemenea, au fost realizate calibrări mecanice fine pentru servo-motor, stabilindu-se unghiurile și timpii optimi de funcționare în vederea asigurării unei dozări corecte și a unei funcționări fiabile.

5.Concluzie

Scopul proiectului a fost indeplinit, sistemul functionand corect si stabil dupa finalizarea etapelor de testare si calibrare.

Pe parcursul dezvoltarii proiectului au fost necesare mai multe adaptari. Au fost corectate probleme legate de setarea orei de hranire, citirea tastaturii si declansarea repetata a servo-motorului. De asemenea, au fost realizate teste pentru calibrarea unghiului servo-motorului, a duratei de activare si a pozitionarii mecanismului de hranire, astfel incat cantitatea de hrana distribuita sa fie constanta si sigura pentru animal. Aceste teste au contribuit la obtinerea unei solutii functionale si fiabile.

Solutia propusa poate fi utila altor utilizatori care doresc o metoda simpla si accesibila de automatizare a procesului de hranire. Datorita utilizarii unor componente comune si a unei arhitecturi clare, sistemul este usor de inteles si de adaptat. Proiectul are si o valoare educationala importanta, fiind potrivit pentru studenti sau pasionati care ar dori sa invete concepte de programare embedded si integrare hardware-software.

Din punct de vedere al imbunatatirilor practice, sistemul poate fi extins prin adaugarea mai multor ore de hranire pe zi, controlul mai precis al cantitatii de hrana, integrarea unei alimentari pe baterii pentru situatii de urgenta sau adaugarea unui modul de comunicatie pentru control de la distanta. Aceste imbunatatiri ar transforma

proiectul intr-o solutie mai avansata, pastrand in acelasi timp baza simpla si robusta realizata in cadrul acestui proiect.