Pixel Pet

Introducere

Pixel Pet este un joc interactiv în care utilizatorul îngrijește un animal virtual, hrănindu-l și jucându-se cu el. Starea pet-ului este reflectată printr-un LED RGB și transmisă prin consola serială. Scopul este de a crea un pet virtual cu care utilizatorii pot interacționa, folosind tehnici de programare pe microcontrolere, cum ar fi întreruperile, PWM și UART. Ideea a fost inspirată de jocurile de tip Tamagotchi, folosind componentele disponibile. Proiectul ajută la înțelegerea programării pe microcontrolere și interacțiunea hardware-software.

Descriere generală

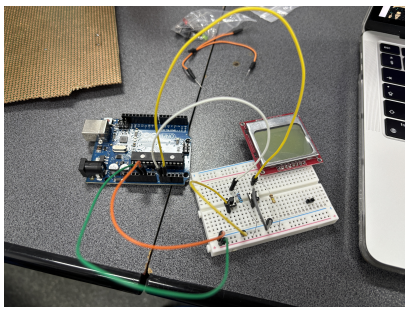
Proiectul se bazeaza pe 3 componente principale:

- Intreruperi: Detectarea apăsării butoanelor pentru a interacționa cu pet-ul.
- PWM: Controlul unui LED RGB pentru a reflecta starea animalului (fericit/trist).
- UART: Comunicarea cu utilizatorul prin consola serială pentru mesaje de stare și comenzi de interacțiune.

Modul de interacțiune:

- Butoanele sunt conectate la pinii de întrerupere ai microcontrolerului, iar la apăsarea acestora se activează întreruperile care modifică starea pet-ului.
- LED-ul RGB este controlat prin PWM pentru a reflecta starea pet-ului (verde pentru fericire, roșu pentru tristețe).
- Comunicarea serială prin UART permite utilizatorului să trimită comenzi pentru hrănire sau joacă și vizualizează starea pet-ului.

Hardware Design



Lista de piese:

- Microcontroler: Arduino UNO
- 3 butoane
- 2 LED-uri RGB
- Display Nokia 5110
- fire tata-tata
- 2 rezistori

Software Design

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare: Arduino IDE
- Limbaj de programare: C++
- librării şi surse 3rd-party:

1. SPI.h - Librărie standard Arduino pentru comunicare SPI
2. Adafruit_GFX.h - Librărie grafică pentru afișaje, dezvoltată de Adafruit
3. Adafruit_PCD8544.h - Librărie specifică pentru controlul afișajului Nokia 5110 (PCD8544), dezvoltată de Adafruit

- algoritmi principali:

1. Sistemul de management al nevoilor:
    1. - Algoritm de foame: Nivelul de foame crește gradual în timp (5% la fiecare 30 secunde)
    2. - Algoritm de hrănire: Activat de un buton fizic, reduce nivelul de foame cu 20% pentru fiecare hrănire completă
2. Sisteme de animație
    1. - Animație de dans: Deplasează pisica pe axa Y într-o mișcare oscilantă
    2. - Animație de alergare: Mișcă pisica pe axa X cu revenire pe partea stângă a ecranului când depășește marginea dreaptă
    3. - Animație de hrănire: Simulează hrănirea prin căderea obiectului “mâncare” de la partea superioară a ecranului către poziția pisicii
3. Sistem de indicare vizuală a stării prin LED-uri
    1. - LED verde: Indică stare de satisfacție completă, aprins când nivelul de foame este 0%
    2. - LED roșu: Indică stare de foame critică, aprins când nivelul de foame depășește 70%

4. Sistem de interacțiune
    1. - Interacțiune hardware: Utilizează un buton fizic pentru hrănirea animalului
    2. - Interacțiune software: Acceptă comenzi prin interfața serială pentru activarea diverselor animații și funcții
5. Mini-joc “Prinde Șoriceii”:
    1. - Algoritm de control al jocului: Sistem de stare pentru gestionarea fazelor de joc (în desfășurare, câștigat, pierdut)
    2. - Algoritm de mișcare a șoricelului: Deplasare pe axa Y cu viteză progresivă crescândă
    3. - Algoritm de detecție a coliziunilor: Verifică distanța relativă între pisică și șoricel pentru determinarea prinderii
    4. - Sistem de punctaj: Acordă puncte pentru fiecare șoricel prins și monitorizează progresul
    5. - Control dual cu butoane: Folosește 2 butoane fizice pentru mișcarea pisicii pe axa X în timpul jocului

• funcţii implementate:

1. Funcții principale
    1. - setup(): Inițializează componentele hardware, afișează mesaje de bun venit, configurează pinii pentru LED-uri și butoane
    2. - loop(): Buclă principală care verifică comenzile, butoanele, actualizează nivelul de foame și gestionează modul de joc
2. Funcții de verificare și procesare
    1. - checkCommands(): Verifică și procesează comenzile primite prin interfața serială
    2. - checkButton(): Verifică starea butonului cu debounce pentru a evita citirile false
    3. - checkHungry(): Actualizează nivelul de foame și generează mesaje de avertizare
    4. - updateLedStatus(): Controlează starea LED-urilor în funcție de nivelul de foame
3. Funcții de joc
    1. - startGame(): Inițializează variabilele de joc și pornește mini-jocul “Prinde Șoricii”
    2. - runMouseGame(): Gestionează logica principală a jocului (mișcarea șoricelului, verificarea coliziunilor)
    3. - checkGameButtons(): Procesează input-ul de la butoanele de control ale jocului cu debounce
    4. - spawnNewMouse(): Generează un nou șoricel pe o poziție aleatorie și crește progresiv viteza
    5. - checkCollision(): Verifică dacă pisica a prins șoricelul bazat pe pozițiile relative
    6. - handleGameWon()/handleGameLost(): Gestionează stările de sfârșit de joc
    7. - endGame(): Închide jocul și oferă recompense pisicii bazate pe performanță
    8. - updateGameDisplay(): Actualizează afișajul specific pentru modul de joc
4. Funcții de afișare
    1. - drawCat(): Desenează sprite-ul pisicii și informațiile de stare pe ecran
    2. - drawHearts(): Desenează inimi decorative animate (opțional)
    3. - drawFood(): Desenează obiectul “mâncare” în timpul procesului de hrănire
    4. - updateDisplay(): Actualizează întregul conținut al afișajului în modul normal
    5. - handleAnimations(): Gestionează toate tipurile de animații (dans, alergare, hrănire)

Rezultate Obţinute

Rezultate Obținute:

Funcționalități Implementate:

- Animal virtual complet: pisică interactivă cu sistem de foame
- Hrănire prin buton: reduce foamea cu 20%
- Animații multiple - dans, alergare, hrănire
- Mini-joc “Prinde Șoarecii” - cu 2 butoane de control
- Feedback vizual - LED verde/roșu + afișaj Nokia 5110
- Control dual - comenzi seriale + butoane fizice

Performanță Tehnică:

- Memorie optimizată - funcționează pe Arduino Uno
- Interfață responsivă - debouncing 50ms pentru butoane
- Update automat - foame +5% la 30 secunde
- Sistem de recompense - jocul reduce foamea cu 40%

Rezultat Final: Tamagotchi funcțional cu 7 comenzi, 3 butoane, mini-joc interactiv și gestionare completă a stării animalului virtual!

Concluzii

alexandra_londraliu.zip===== Download =====

Jurnal

- 12.05: am primit comanda cu piesele necesare
- 13.05: este gata o parte din hardware
- 16.05: am ars lcd-ul si am dat comanda de altul
- 18.05: am facut rost de alt lcd
- 20.05: gata o parte din software
- 22.05: gata software & hardware
- 25.05: am infrumusetat aspectul vizual al proiectului

Bibliografie/Resurse

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

Export to PDF