Doggy-clock

Introducere

Proiectul constă într-un robot ingenios, cu aspect de câine, care își asumă rolul unui ceas desteptător. În momentul în care ora dorită este atinsă, acesta nu doar sună, ci și cântă, apoi pornește pe un traseu predefinit. Scopul său este acela de a crea o experiență interactivă și distractivă pentru utilizator, încurajându-l să se ridice și să-l urmărească pentru a-l opri.

Ideea a pornit de la dorința de a combina utilul cu plăcutul, oferindu-mi o alternativă amuzantă și eficientă pentru ceasurile obișnuite. Consider că acest proiect este folositor atât pentru mine, asigurându-mi o modalitate inedită de a începe ziua, cât și pentru alții, oferindu-le o privire asupra creativității și inovației în domeniul roboților și designului. Mai mult, robotul este util și pentru a rezolva problema snooze-urilor constante ale alarmelor, ce pot rezulta într-o trezire întârziată.

Descriere generală

1. **Arduino Uno:** Este microcontrollerul principal care coordonează întregul sistem. Primește informații despre ora curentă de la modulul de ceas (clock) și monitorizează dacă ora setată de utilizator a fost atinsă.

2. **Modulul de ceas (Clock):** Furnizează informații despre ora curentă către Arduino Uno. Aceste informații sunt utilizate pentru a verifica dacă ora setată de utilizator a fost atinsă sau nu.

3. **Reached selected time?:** Această parte a programului verifică dacă ora curentă corespunde cu ora setată de utilizator. Dacă da, continuă cu acțiunile următoare. Dacă nu, revine la modulul de ceas pentru a verifica din nou ora.

4. **Buzzer and Engine:** Dacă ora setată de utilizator a fost atinsă, acest modul este activat. Buzerul emite sunetul de alarmă, iar motorul începe să se deplaseze pe traseul prestabilit.

5. **Button:** Acesta este butonul pe care utilizatorul îl poate apăsa pentru a opri alarmă și mișcarea robotului.

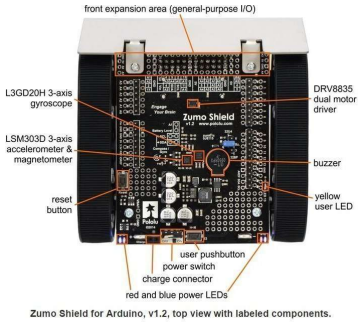
6. **If not pressed:** Dacă utilizatorul nu apasă butonul, sistemul revine la monitorizarea stării butonului, continuând să emită sunetul de alarmă și să se deplaseze pe traseu până când butonul este apăsat.

7. **If pressed:** Dacă utilizatorul apasă butonul, sistemul oprește sunetul de alarmă și mișcarea motorului.

Hardware Design

Listă de piese:

- Zumo Shield v1.2
- Placă Plusivo
- Ecran pentru afișare
- RTC DS1307
- Senzor infraroșu

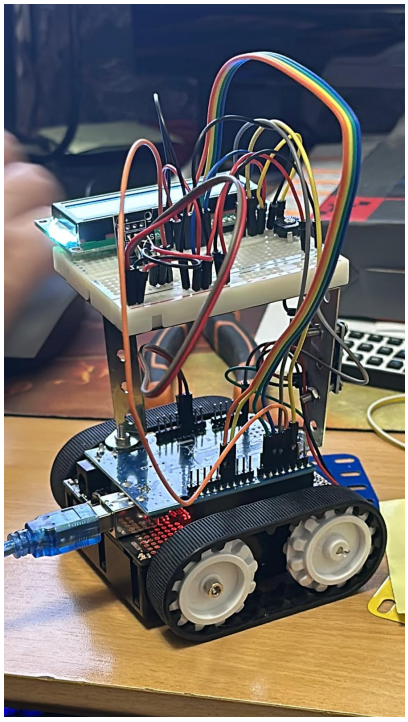


Zumo Shield for Arduino, v1.2, top view with labeled components.

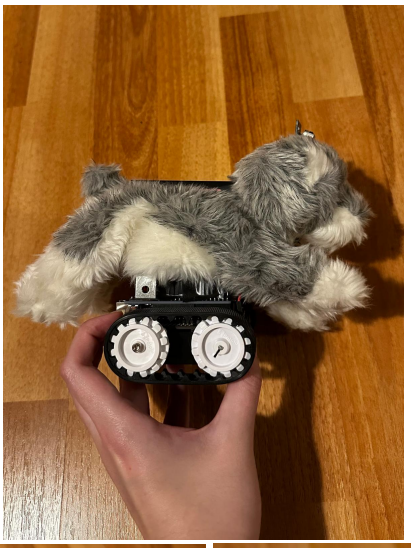
* In urma finalizării codului, am renunțat la led- ul ce era pe pin-ul 13, neavând nevoie de el, și am pus receiver-ul pe pin-ul respectiv.

Robot

Inițial:



Produs final:



Demo-ul video se găsește în drive-ul de mai jos!

Software Design

Descrierea codului aplicației (firmware):

- Mediu de dezvoltare: Arduino IDE
- Librării și surse 3rd-party:
  1. `IRremote.h` pentru gestionarea semnalului IR
  2. `Wire.h` pentru comunicația I2C
  3. `LiquidCrystal.h` pentru afișajul LCD
  4. `RTClib.h` pentru modulul RTC
  5. `ZumoShield.h` pentru controlul motorului și buzzer-ului
- Algoritmi și structuri pe care plănuiți să le implementați:
  1. Setarea și afișarea datei și orei folosind RTC DS1307
  2. Setarea unei alarme folosind un senzor IR pentru input
  3. Redarea unei melodii și mișcarea robotului la activarea alarmei
- (Etapa 3) Surse și funcții implementate:
  1. Funcții pentru setarea orei și datei alarmei
  2. Funcții pentru afișarea datei și orei pe LCD
  3. Funcție pentru redarea unei melodii și controlul motoarelor Zumo la activarea alarmei

Laboratoare utilizate

1. **Laboratorul 1: UART (Universal Asynchronous Receiver-Transmitter):**

- Comunicarea serială pentru debugging și afișarea mesajelor în serial monitor.
- De exemplu, inițializarea comunicării seriale în setup() (Serial.begin(19200);) și utilizarea Serial.println() pentru a afișa mesaje de diagnosticare și rezultate IR.

2. **Laboratorul 2: Întreruperi (Interrupts):**

- În cazul gestionării senzorului IR, unde întreruperile sunt folosite pentru a detecta semnalul de la telecomandă.
- De exemplu, funcția irrecv.enableIRIn(); pornește receptorul IR, care utilizează întreruperi pentru a recepționa semnalele.

3. **Laboratorul 3: Timere. PWM (Pulse Width Modulation):**

- Controlul motoarelor Zumo prin PWM pentru a ajusta viteza acestora.
- De exemplu, utilizarea funcțiilor motors.setLeftSpeed(speed); și motors.setRightSpeed(speed); pentru a controla viteza motoarelor.

4. **Laboratorul 6: I2C (Inter-Integrated Circuit):**

- Comunicarea cu modulul RTC DS1307 și afișajul LCD.
- De exemplu, inițializarea comunicației I2C în setup() cu Wire.begin(); și folosirea librăriei RTClib.h pentru a interacționa cu modulul RTC.

Rezultate Obţinute

În urma realizării proiectului, am obținut următoarele rezultate:

- Implementarea cu succes a setării orei și datei alarmei folosind telecomanda IR
- Afișarea corectă a datei și orei curente pe ecranul LCD
- Redarea unei melodii prestabilite și mișcarea robotului Zumo la activarea alarmei
- Sincronizarea corectă a timpului folosind RTC DS1307

Deși intenția inițială a fost ca robotul să scoată sunete asemănătoare unui lătrat, nu am reușit să reproduc acest efect folosind note muzicale. În schimb, am optat pentru redarea unei melodii scurte ca ton de apel. În plus, am renunțat la utilizarea unui buton fizic în favoarea telecomenzii, considerând că aceasta oferă mai multă libertate în controlul robotului.

Concluzii

Proiectul de realizare a unui robot cu funcționalitate de alarmă s-a dovedit a fi un succes. Utilizarea senzorului IR pentru setarea orei și datei alarmei a simplificat interacțiunea utilizatorului cu sistemul. Afișajul LCD a permis verificarea ușoară a timpului curent și a setărilor alarmei. Implementarea

codului a demonstrat că robotul poate reda o melodie și poate efectua mișcări în momentul activării alarmei, oferind astfel un exemplu practic de utilizare a componentelor hardware și a bibliotecilor software disponibile pentru Arduino.

Codul inclus pentru gestionarea afișajului LCD, a senzorului IR, a modulului RTC și a controlului motorului și buzzer-ului a funcționat conform așteptărilor, iar proiectul poate fi extins cu ușurință pentru a include funcționalități suplimentare, cum ar fi controlul prin aplicații mobile sau integrarea cu alte module de senzori pentru detectarea obstacolelor sau a liniilor traseului.

Download

https://drive.google.com/drive/folders/1oPTOUkiUTXF1mzVAzsIgkYLWNibOeZcv?usp=drive_link

Bibliografie/Resurse

https://github.com/pololu/zumo-shield-arduino-library/tree/master

Mai multe materiale se găsesc în drive-ul de mai sus.

Export to PDF