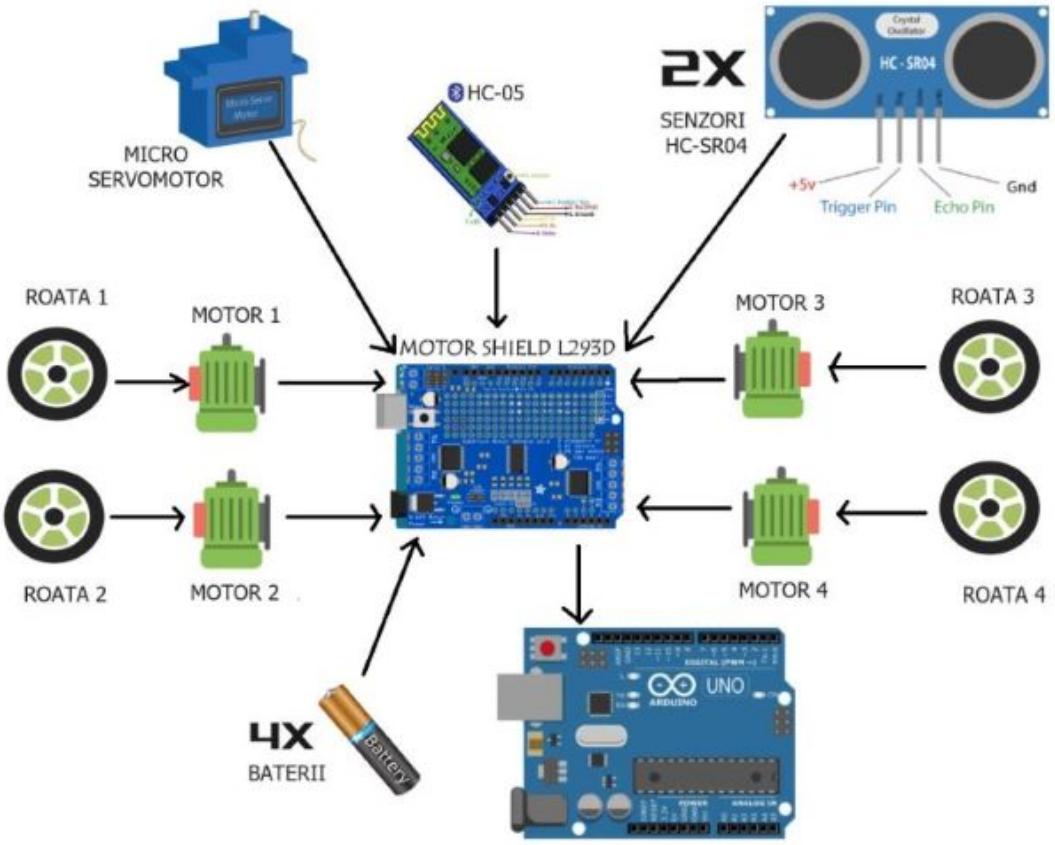
Smart Collision-Avoidance Car

Autor: Drăgan Tudor Mihai
Grupa: 333CD

Introducere

Smart Collision-Avoidance Car este un proiect ce implică dezvoltarea unei mașinuțe controlate autonom, echipată cu tehnologie avansată de detecție a obstacolelor și evitare a coliziunilor. Această mașinuță are capacitatea de a detecta obstacolele din mediul înconjurător și de a evita coliziunile prin ajustarea traiectoriei sale. Cu ajutorul algoritmilor inteligenți și a senzorilor adecvați, Smart Collision-Avoidance Car reprezintă o soluție sigură și eficientă pentru a naviga în spații înguste sau cu trafic dens, fără a necesita intervenția umană constantă.

Descriere generala



Smart Collision-Avoidance Car este un sistem robotic autonom conceput pentru a naviga și evita obstacolele în medii controlate. Proiectul utilizează componentele hardware și software pentru a realiza un vehicul capabil să opereze independent, cu capacitatea de a fi comandat de la distanță prin Bluetooth.

Hardware Design

Listă piese:

1 X Kit masinuta ce contine: 2 x Placi acrilic 4 x Cauciucuri 4 x Motoare 3-6V cu reductor 1 x Suport baterii 4AA 1 x Set fire motoare 4 x Encodere 8 x Suporti acrilic pentru motoare 6 x Piloni hexagonali 30mm 8 x Suruburi M3x30mm 16 x Suruburi M3x5mm 10 x Piulite M3

Arduino Uno: Este un microcontroler bazat pe ATmega328P. Acesta servește drept creier al vehiculului, gestionând toate operațiunile de input și output. Arduino Uno interpretează datele de la senzori și emite comenzi către motor shield pentru controlul motoarelor.

Motor Shield L293D: Este un driver de motor care permite Arduino să controleze până la patru motoare DC prin intermediul semnalelor PWM (Pulse Width Modulation). Shield-ul protejează Arduino de suprasarcini și facilitează controlul direcției și vitezei motoarelor.

Motoare DC (Motor 1, Motor 2, Motor 3, Motor 4): Acestea sunt motoare cu reduceri, fiecare atașat la o roată a vehiculului. Motoarele sunt controlate individual pentru a permite manevre precise și pentru a ajusta traiectoria vehiculului în funcție de necesități.

Roți (Roata 1, Roata 2, Roata 3, Roata 4): Fiecare roată este echipată cu un encoder care furnizează feedback despre viteza și distanța parcursă, contribuind la controlul mai precis al poziționării și navigației.

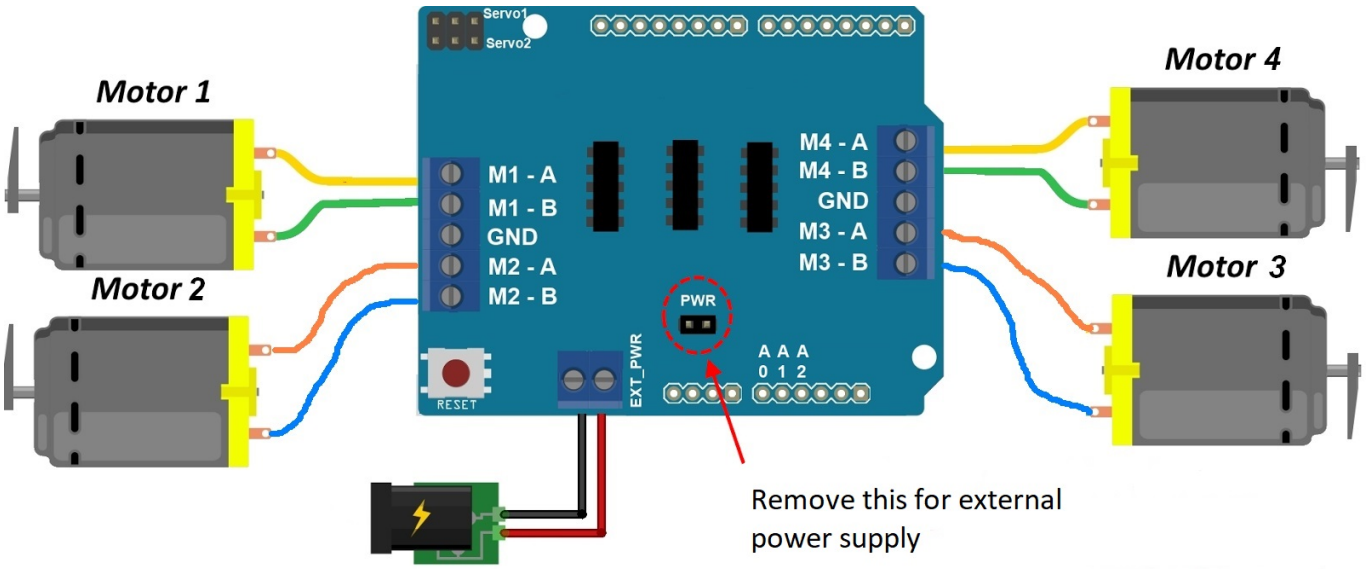
Senzori ultrasonici HC-SR04: Acești senzori emit unde ultrasonice care se reflectă înapoi la senzor după întâlnirea unui obstacol. Distanța până la obstacol este calculată prin măsurarea timpului pe care îl ia unda să se întoarcă. Această informație este utilizată pentru a detecta și evita obstacolele din calea vehiculului.

1 x Baterie externa pentru alimentare Arduino

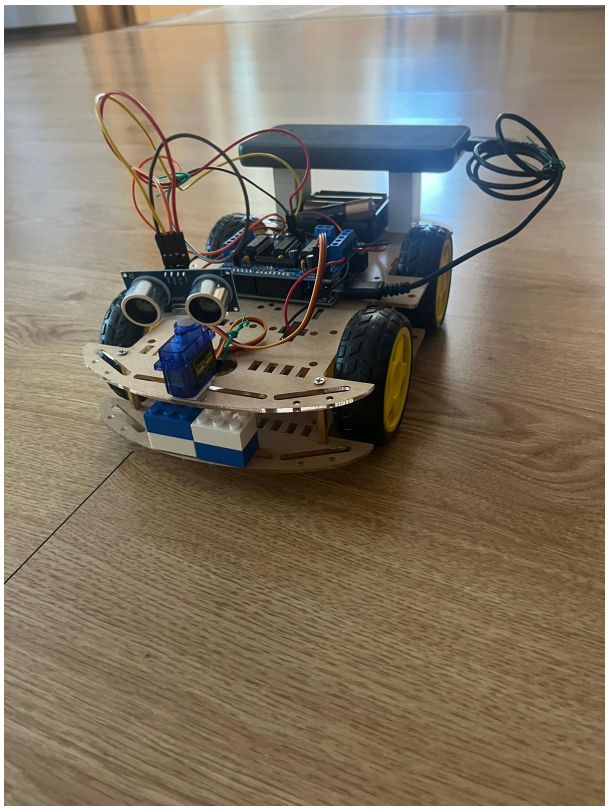
Baterii (4X Baterii): Alimentează motor shield și motoarele DC.

Schema electrica

Schema electrica pentru L293D Motor Shield



Implementare hardware



Software Design

Proiectul a fost dezvoltat în Arduino IDE.

Biblioteci Folosite:

AFMotor: Aceasta bibliotecă este utilizată pentru a controla motoarele DC prin intermediul unui Adafruit Motor Shield.

Servo: Aceasta bibliotecă este utilizată pentru a controla un servo motor, care rotește senzorul ultrasonic pentru a detecta obstacolele.

Funcționalitatea Codului: Codul implementează o mașinuță robotizată care poate detecta și evita obstacolele folosind un senzor ultrasonic și un servo motor. Mașinuța folosește un Arduino pentru a controla patru motoare DC prin intermediul unui motor shield.

Descrierea Codului

Definirea și Inițializarea Componentelor:

Definim pinii pentru senzorul ultrasonic și servo. Inițializăm obiectele pentru motoare și servo. Setăm viteza motoarelor la o valoare prestabilită.

Setup:

Configurăm pinii pentru senzorul ultrasonic. Atașăm servo-ul la pinul definit. Realizăm secvența de pornire Setăm viteza motoarelor.

Loop Principal:

Măsurăm distanța față de un obstacol folosind senzorul ultrasonic. Dacă distanța este mai mică sau egală cu un prag definit, mașinuța va efectua o secvență de evitare a obstacolului: Se oprește. Merge înapoi pentru un scurt timp. Verifică distanța în stânga și în dreapta. Se rotește în direcția cu mai mult spațiu liber. Dacă nu există obstacole, mașinuța merge înainte. Funcții pentru Mișcarea Mașinuței:

moveForward(): Funcție pentru mișcarea înainte a mașinuței.

moveBackward(): Funcție pentru mișcarea înapoi a mașinuței.

turnLeft(): Funcție pentru întoarcerea la stânga.

turnRight(): Funcție pentru întoarcerea la dreapta.

stopMotors(): Funcție pentru oprirea motoarelor.

Funcții pentru Detectarea Distanței:

measureDistance(): Măsoară distanța față de un obstacol folosind senzorul ultrasonic.

lookLeft() și lookRight(): Rotește servo-ul la stânga și la dreapta și măsoară distanțele în aceste direcții. Secvența de Pornire:

startSequence(): O secvență de mișcări ale servo-ului pentru a indica pornirea mașinuței.

Concluzii

Acest proiect a fost o oportunitate excelentă de a aplica cunoștințele teoretice într-un context practic. Am învățat să îmbin componente hardware diverse și să dezvolt un software robust pentru controlul acestora. Experiența a fost valoroasă, oferindu-mi o înțelegere mai bună roboticii de bază și a sistemelor integrate. Pe parcursul proiectului, am întâmpinat și depășit diverse provocări, mai ales prin faptul că bateriile nu furnizau suficient curent pentru toate cele 4 motoare. În final pot spun că acest proiect a contribuit semnificativ la dezvoltarea mea profesională și tehnică.

Download

self_autonoums_car.zip

Jurnal

- 28.04.2024: Alegerea temei de proiect
- 04.05.2024: Crearea paginii proiectului și realizarea introducerii și a descrierii generale.
- 14.05.2024 - 16.05.2024: Implementare hardware
- 22.05.2024 - 24.05.2024: Implementare software

Bibliografie/ Resurse

Resurse:

- https://www.youtube.com/watch?v=l62fjzmZHWo&t=437s
- https://srituhobby.com/obstacle-avoidance-robot-car/
- https://www.circuito.io/

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2024/iotelea/tudor_mihai.dragan

Last update: **2024/05/27 09:43**