

Obstacle Avoiding Car - Anghelescu Albert-Mihai 331CD

Introducere

Obstacle Avoiding Car reprezintă o platformă mobilă autonomă proiectată pentru detectarea și evitarea obstacolelor din mediul înconjurător. Sistemul utilizează Arduino Nano pentru procesarea datelor și coordonarea componentelor hardware, împreună cu un senzor ultrasonic montat pe un servomotor pentru scanarea mediului.

Scopul proiectului este implementarea unui sistem embedded capabil să realizeze deplasare autonomă și reacție în timp real la apariția obstacolelor.

Acest proiect este util atât din perspectivă educațională, deoarece implică lucrul cu microcontrolere, actuatori și senzori, cât și din perspectivă practică, reprezentând o bază pentru sisteme autonome mai complexe.

Descriere generală

Sistemul este alcătuit dintr-o platformă cu 4 roți, controlată de un Arduino Nano. Controlul motoarelor este realizat prin intermediul unui driver L298N, care permite comandarea direcției și vitezei motoarelor utilizând semnale PWM generate de microcontroler.

Detectarea obstacolelor se realizează cu ajutorul unui senzor ultrasonic HC-SR04 montat pe un servomotor SG90. Servomotorul permite rotirea senzorului pe diferite unghiuri, oferind posibilitatea scanării mediului în direcțiile stânga, centru și dreapta.

Pe baza informațiilor primite de la senzor, microcontrolerul decide direcția optimă de deplasare, evitând obstacolele prin oprire, schimbare de direcție sau rotație.

Schema bloc a sistemului:



Interacțiunea modulelor:

- Arduino controlează întregul sistem
- L298N controlează alimentarea și direcția motoarelor
- SG90 rotește senzorul ultrasonic
- HC-SR04 furnizează date despre distanța până la obstacole

Hardware Design

Lista de componente hardware:

- Arduino Nano
- L298N Motor Driver Module
- HC-SR04 Ultrasonic Sensor
- SG90 Micro Servo Motor
- 4 x motoare DC cu reductor (3-6V)
- 4 x roți compatibile
- 2 x baterii 18650

Descriere hardware:

- Arduino Nano reprezintă unitatea centrală de control
- L298N permite controlul simultan al celor 4 motoare (2 pe stânga și 2 pe dreapta)
- HC-SR04 măsoară distanța față de obstacole
- SG90 rotește senzorul pentru scanarea mediului

Schema electrica:



Software Design

Mediu de dezvoltare: PlatformIO Biblioteci: doar biblioteci standard AVR (`avr/io.h`, `avr/interrupt.h`, `util/delay.h`).

Implementare

`hcsr04.c` — driver pentru senzorul ultrasonic folosind Timer2 (prescaler 64, 4us/tick). `HCSR04_measure()` trimite pulsul de trig de 10us, asteapta frontul crescator al echo-ului, numara tick-urile cat echo-ul e HIGH (cu tracking manual de overflow), si converteste in centimetri prin formula $\text{ticks} * 4\text{us} / 58$.

`l298n.c` — driver pentru motoare cu functii de directie (`forward`, `backward`, `turn_left`, `turn_right`, `stop`) si control viteza prin `set_speed(left, right)`. Viteza e controlata prin PWM pe Timer0 in Fast PWM mode (OC0A = ENB, OC0B = ENA, ~980Hz). Virajele sunt de tip pivot — rotile din partea opusa merg invers pentru rotatie pe loc.

`servo.c` — semnal PWM 50Hz generat cu Timer1 in Fast PWM mode cu TOP = ICR1 = 40000 (prescaler 8, 16MHz). `servo_write(us)` primeste o valoare in microsecunde si seteaza OCR1A = $us * 2$.

`main.c` — automat de stari cu 3 stari:

- **FORWARD**: robotul merge inainte si masoara distanta la fiecare 50ms. La detectia unui obstacol sub 45cm, face o a doua masurare dupa 20ms pentru confirmare (eliminarea fals-pozitive), dupa care opreste motoarele si trece in SCAN.
- **SCAN**: servo-ul se roteste la stanga si se masoara distanta, apoi la dreapta si se masoara din nou, dupa care revine la centru. Intre fiecare pozitie exista o pauza de 400ms pentru stabilizarea servo-ului.
- **TURN**: robotul vireaza pivot spre directia cu distanta mai mare masurata in SCAN, timp de 300-400ms, dupa care trece inapoi in FORWARD.

Rezultate Obținute

Video: https://youtube.com/shorts/PWHoilhkw_k?feature=share

Concluzii

Github: https://github.com/AnghelescuAlbert/Object_Avoiding_Car/tree/master

Download

Jurnal

Bibliografie/Resurse

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/farhad_ali.gul/anghelescu.albert



Last update: **2026/05/19 15:35**