

Maze-Runner: Robot Autonom Wall-Follower

Introducere

Acest proiect consta in realizarea unui robot mobil autonom capabil sa navigheze printr-un labirint, mentinand o traiectorie stabila prin urmarirea peretilor (Wall-Following).

- **Ce face:** Utilizeaza trei senzori ultrasonici (HC-SR04) pentru a masura distanta fata de obstacole. Pe baza datelor, un algoritm PID ajusteaza viteza motoarelor via PWM pentru a mentine robotul la o distanta fixa de perete.
- **Scopul lui:** Realizarea unui sistem de navigare autonoma bazat pe reactii rapide la mediu prin bucle de feedback negativ, fara o harta predefinita.
- **Ideea de la care am pornit:** Explorarea roboticii mobile si implementarea unui algoritm de control clasic (PID) pentru a elimina mersul in "zig-zag".
- **De ce este util:** Pentru echipa, este un exercitiu esential de procesare a semnalelor prin intreruperi si implementare a controlului automat bare-metal. Pentru ceilalti, exemplifica generarea unui comportament autonom complex printr-o logica de control simpla.

Descriere generală

Fluxul de date planificat si interactiunea modulelor:

1. Preluare Date Sensori: La intervale regulate (ex. 50ms), MCU declanseaza senzorii ultrasonici si calculeaza distantele (Fata, Stanga, Dreapta) folosind Timere/Intreruperi.
2. Calcul Eroare: Se calculeaza diferenta dintre distanta laterala curenta si referinta dorita.
3. Procesare PID: Modulul central aplica algoritmul de control pe baza erorii pentru a calcula corectia necesara.
4. Generare Semnal: MCU genereaza semnalele PWM ajustate si pinii de directie.
5. Actionare: Driverul L298N primeste comenzile si modifica asimetric turatia celor doua motoare DC pentru a corecta traiectoria.

Logica de navigare foloseste o ierarhie stricta: EVITARE COLIZIUNE FRONTALA > URMARIRE PERETE LATERAL. Daca senzorul frontal scade sub un prag critic, robotul suprascrie bucla PID si executa o rotatie de urgenta.



Hardware Design

Design-ul fizic este conceput pentru a asigura stabilitatea robotului in timpul navigarii prin labirint,

utilizand un sasiu alungit 2WD care permite montarea strategica a senzorilor. Elementul central il reprezinta sistemul de detectie format din trei senzori ultrasonici, dispusi pentru a acoperi zonele frontala, stanga si dreapta, oferind robotului o "viziune" completa asupra mediului inconjurator.

Lista de piese:

- 1 x Microcontroler Arduino UNO R3 (pe placa de dezvoltare)
- 1 x Kit Sasiu Masina 2WD (include 2 motoare DC, roti, roata caster si suport baterii)
- 1 x Modul Driver Motoare Dual L298N
- 3 x Sensori Ultrasonici HC-SR04 (pentru detectia peretilor pe 3 directii)
- 1 x Pachet 4 baterii AA (alimentare forta motoare)
- 1 x Breadboard 170 puncte
- Fire de conexiune Dupont (Tata-Tata pentru breadboard, Mama-Tata pentru senzori si driver)
- Un switch

Board Pin	Function	Component	Direction	Description
Pin 4	Digital Out	Driver L298N (IN1)	Output	Directie motor stang
Pin 5	Digital Out	Driver L298N (IN2)	Output	Directie motor stang
Pin 6	Digital Out	Driver L298N (IN3)	Output	Directie motor drept
Pin 7	Digital Out	Driver L298N (IN4)	Output	Directie motor drept
Pin 10	PWM Out	Driver L298N (ENA)	Output	Viteza motor stang
Pin 11	PWM Out	Driver L298N (ENB)	Output	Viteza motor drept
Pin 2	Digital Out	Senzor Stanga (Trig)	Output	Trigger ultrasunete
Pin 3	Digital In	Senzor Stanga (Echo)	Input	Receptie ecou
Pin 8	Digital Out	Senzor Mijloc (Trig)	Output	Trigger ultrasunete
Pin 9	Digital In	Senzor Mijloc (Echo)	Input	Receptie ecou
Pin 12	Digital Out	Senzor Dreapta (Trig)	Output	Trigger ultrasunete
Pin 13	Digital In	Senzor Dreapta (Echo)	Input	Receptie ecou

Schematic



Software Design

Dezvoltarea software-ului se realizeaza la nivel bare-metal pentru microcontrolerul ATmega328P (Arduino UNO R3), avand la baza o bucla de control continua si esantionare directa (polling). Proiectul integreaza concepte fundamentale de **GPIO** (Laboratorul 0) si **Timere/PWM** hardware (Laboratorul 3).

* **Mediu de dezvoltare:** Codul este scris in C, utilizand Visual Studio Code cu extensia PlatformIO pentru compilare si upload. * **Librarii si surse 3rd-party:** Se folosesc exclusiv bibliotecile standard AVR (`<avr/io.h>` si `<util/delay.h>`). Nu se utilizeaza framework-ul standard Arduino, controlul perifericelor realizandu-se prin manipularea directa a registrilor. * **Algoritmi si structuri de date:**

- **Algoritm PD (Proportional-Derivativ):** O bucla de feedback negativ implementata prin variabile globale, care calculeaza eroarea de pozitionare fata de peretele din dreapta si aplica o corectie directa asupra vitezei motoarelor.
- **Logica de navigare ierarhica:** Controlul urmeaza o structura decizionala stricta in bucla principala (`if`), unde evitarea coliziunii frontale are prioritate absoluta si poate suprascrie temporar algoritmul de urmarire a peretelui.

* **Surse si functii implementate:**

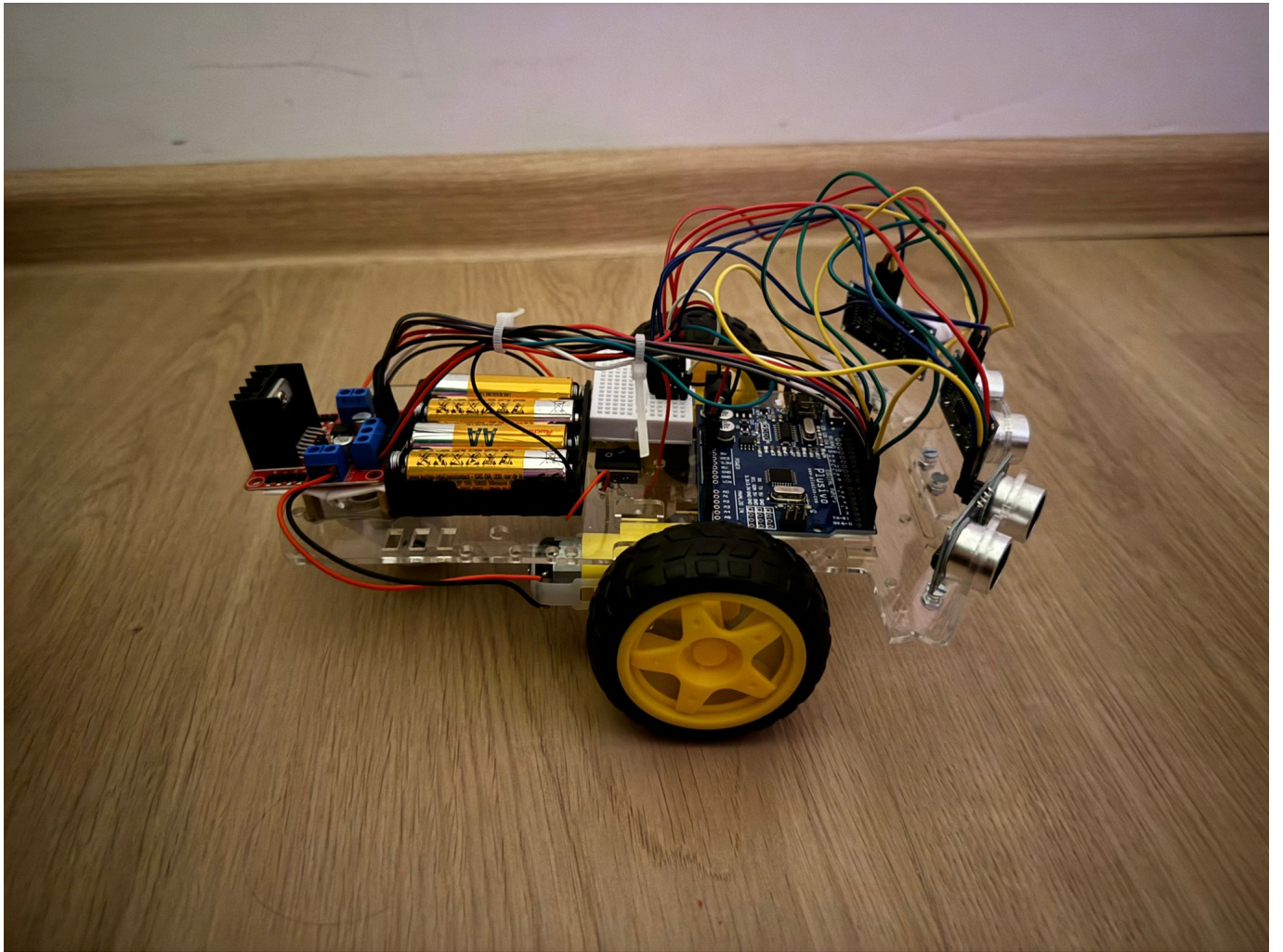
- ``init_hardware()``: Configureaza pinii GPIO ca intrari/iesiri pentru senzori si puntea H. Initializeaza Timer1 (PWM pe 8 biti pentru controlul ENA) si Timer2 (Fast PWM pentru controlul ENB) cu un prescaler de 8.
- ``read_distance(...)``: Declanseaza senzorul ultrasonic HC-SR04 si calculeaza durata impulsului de Echo prin tehnica de polling (esantionare continua a pinului intr-o bucla blocanta cu timeout), returnand distanta convertita in centimetri.
- ``set_motors(speedL, speedR)``: Gestioneaza pinii de directie (IN1-IN4) din portul D si incarca factorul de umplere calculat in registrele de comparare ``OCR1B`` (motor stanga) si ``OCR2A`` (motor dreapta).

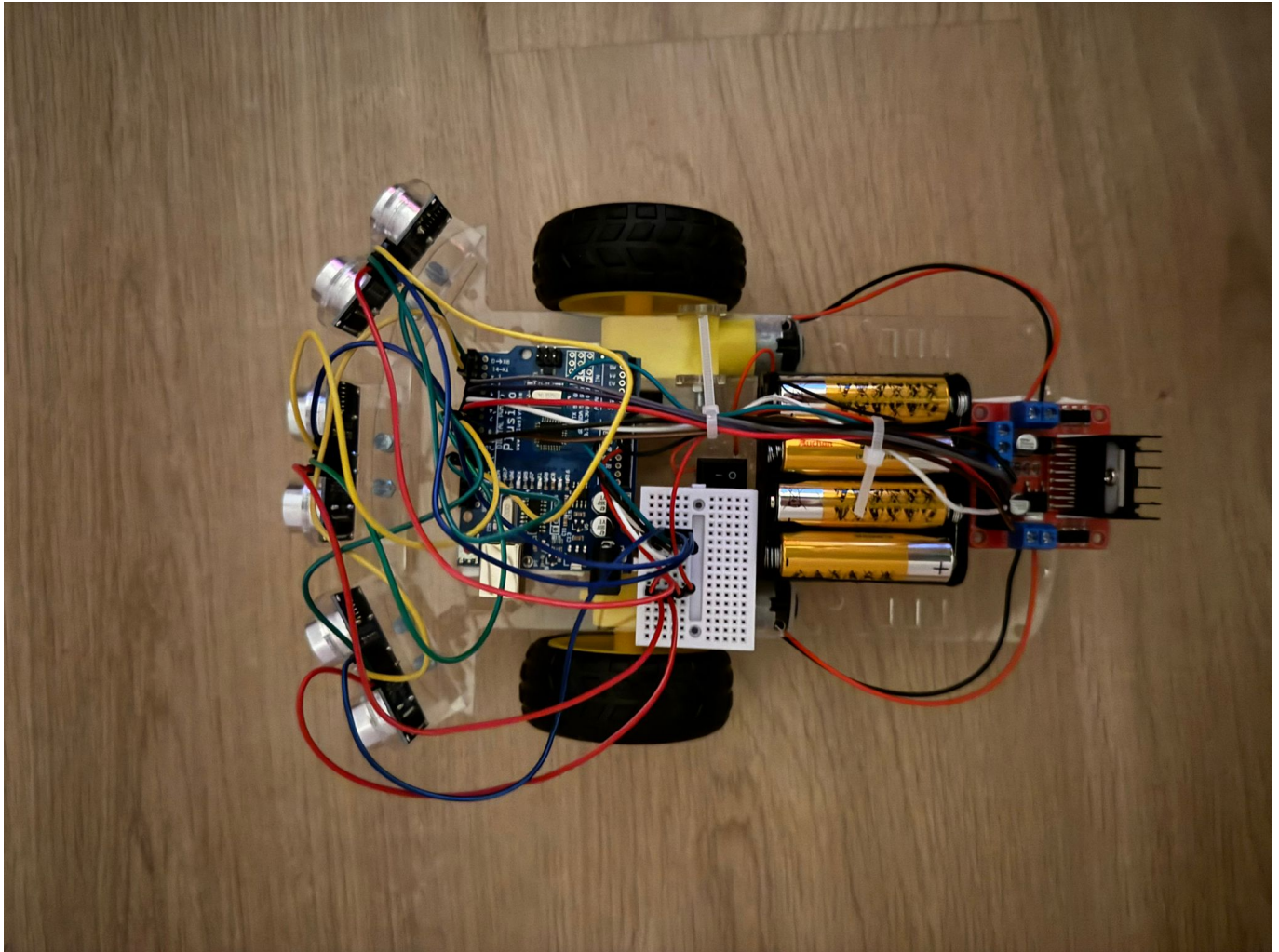
Cod Proiect

<https://github.com/mihneapopesq/Wall-Follower/>

Jurnal

Data	Activitate
6 mai	A ajuns sasiul necesar robotului si driverul de motoare.
11 mai	Plasare comanda pentru mini breadboard.
13 mai	Asamblarea fizica a robotului.
15 mai	Realizarea lipiturilor pentru componentele robotului.
19 mai	Plasare comanda pentru carcusele senzorilor ultrasonici (stabilitate mecanica).
20 mai	Inceperea implementarii software.





Bibliografie/Resurse

- [Tutorial asamblare sasiu](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/florin.stancu/mihnea.popescu1811>



Last update: **2026/05/25 01:39**