

Maze Solver - Stan Andreea-Cristina 333CA

Introducere

- Ce presupune proiectul?

Proiectul presupune construirea unui robot autonom capabil să găsească calea de ieșire dintr-un labirint cu pereți opaci. Acesta va avea atașați senzori care, împreună cu algoritmi de navigare, va explora labirintul și va găsi soluția optimă pentru a ajunge la ieșire.

- Care este scopul lui?

Scopul proiectului este de a oferi o soluție practică pentru situațiile în care este necesară explorarea sau navigarea în medii restrictive și periculoase.

- Care a fost ideea de la care am pornit?

Ideea a pornit ca un sprijin pentru zonele afectate de conflicte armate unde sunt prezente mine terestre, întrucât existența unui robot capabil să le detecteze și să ofere o soluție sigură și eficientă pentru eliminarea lor ar putea fi de mare ajutor.

- De ce e util?

Poate contribui la salvarea vieților și la reducerea impactului negativ asupra oamenilor, unde un robot autonom capabil să se deplaseze într-un mediu restrictiv poate fi de mare ajutor.

Descriere generală

Schema bloc:



Robotul are atașat în partea din față 6 senzori ce îl ajută să se mențină pe pistă și să se deplaseze corespunzător. Plăcuța Arduino este plasată central și realizează conexiunea între toate celelalte elemente componente (conexiune directă cu driverul de motor ce este legat de cele 4 motoare, senzorii analogici, alimentarea realizată cu ajutorul bateriilor, ecranul LCD).

Hardware Design

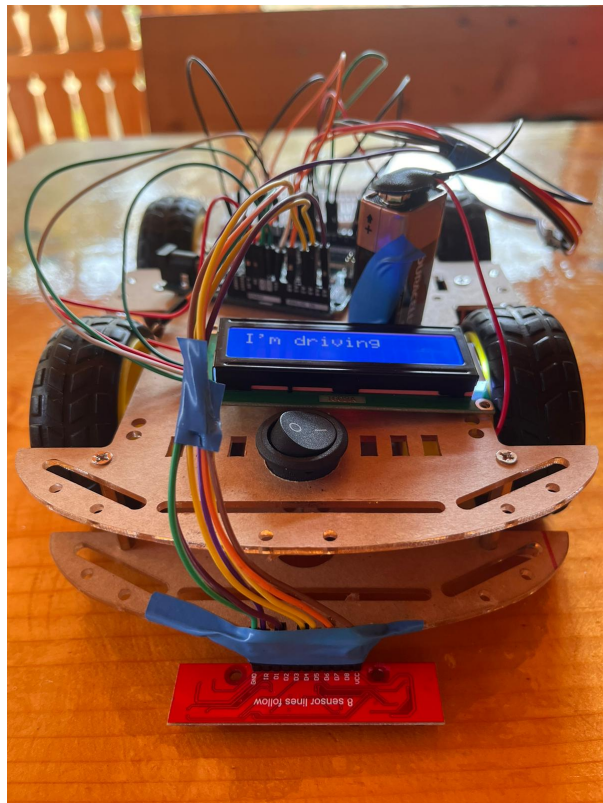
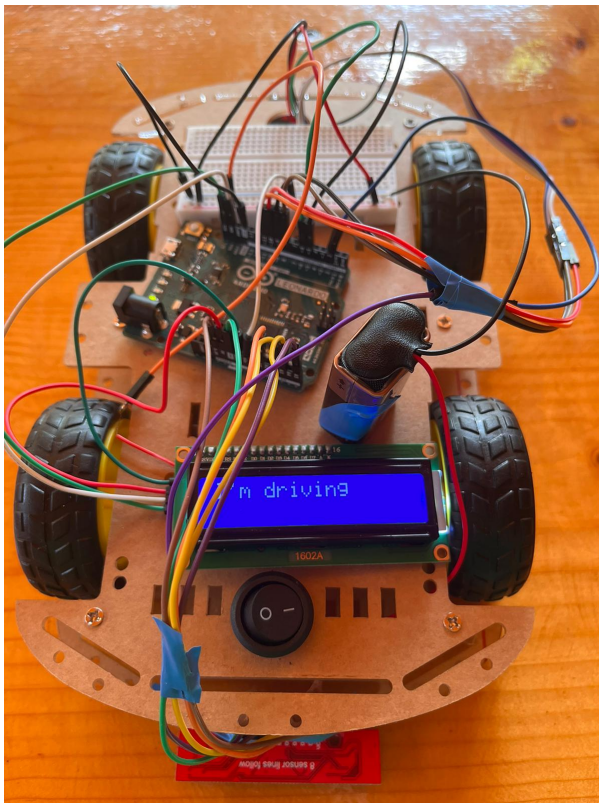
Piesele utilizate:

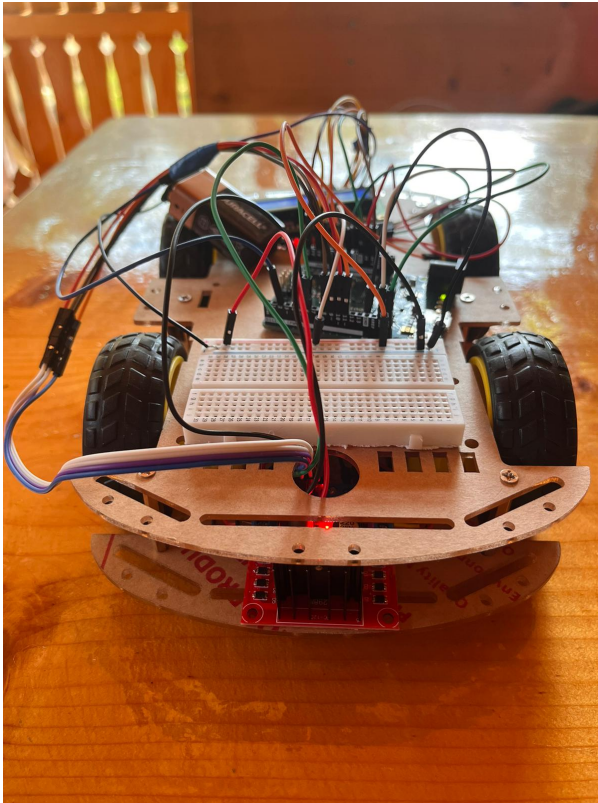
- Driver motor L298N
- Motoare
- Roti
- Senzori analogici
- Arduino Leonardo
- Conectori
- Baterie 9V
- Breadboard
- LCD

Schema electrică:



Software Design





Se vede pe filmare faptul că la final, datorită podelei inegale, mașinii îi este dificil să ia ultimul viraj, însă reușește în final. De asemenea, am filmat roboțelul fără ecran, deoarece s-ar fi consumat și mai multă baterie (Duracell sponsor me)

Link demo: https://youtube.com/shorts/i08KIP_Tbvl?feature=share

Mediu de dezvoltare: Arduino IDE

Biblioteci utilizate: Wire.h, LiquidCrystal_I2C.h

Programul este gândit să detecteze când ar trebui ca mașina să facă viraje folosind regula mâinii stângi: întâi încearcă să vireze la stânga, dacă nu reușește încearcă să meargă înainte, apoi în dreapta și în final dacă niciunul din aceste cazuri nu sunt îndeplinite, se întoarce, virând la stânga până când senzorul din mijloc vede iar linia. Ecranul afișează un mesaj corespunzător dacă acesta merge sau stă pe loc. Cu ajutorul unui întrerupător alimentez motoarele și plăcuța arduino.

Funcția unde se află toată logica implementării este loop(), unde se și realizează regula mâinii stângi și tot acolo se realizează citirea de pe senzori cu ajutorul analogRead() ca după să se decidă ce viraje are de realizat robotul.

Motoarele sunt grupate câte două (cele din stânga și cele din dreapta), întrucât pe fiecare parte comenzile de deplasare sunt la fel pentru ambele roți și sunt controlate prin PWM, utilizând funcția analogWrite() din framework.

Arhiva codului : [mazesolver_stan_andreea.zip](#)

Concluzii

A fost un proiect destul de challenging, a fost foarte greu să fac mașina să meargă mai ales pentru că bateriile se duceau foarte repede, iar podeaua nu era perfect dreaptă astfel că virajele erau luate destul de greu(cum se vede și în filmare la finalul traseului), nici aderența roților pe hârtia albă nu a fost așa cum ar fi trebuit, ceea ce a îngreunat foarte mult realizarea proiectului.

A fost prima oară când am realizat și partea hardware pentru un proiect, însă am învățat lucruri noi care cu siguranță vor fi utile.

Jurnal

1. 25 aprilie - achiziționare piese
2. 30 aprilie - pagină documentație
3. 1-15 mai - realizare hardware
4. 15-26 mai - realizare software]]

Bibliografie

1. Laboratorul 3 PWM : <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023>
2. Laborator 4 ADC : <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2022>
3. Laborator 6 I2C : <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2022>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/adarmaz/maze-solver>



Last update: **2023/05/28 20:57**