

Line-Follower Robot - Popescu Silviu 333CA Demo Proiect

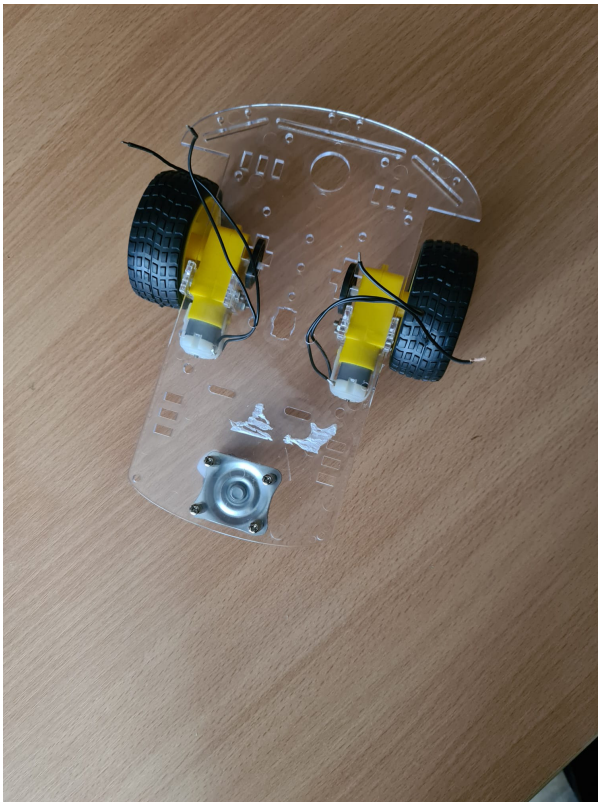
Hardware Design

Lista componente:

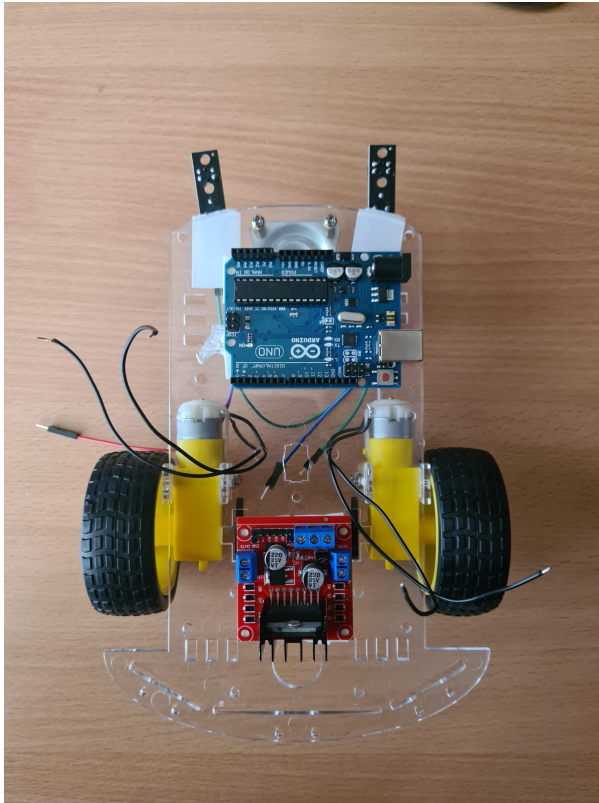
- Arduino Uno R3 ATmega328P x 1
- Modul Senzor Infrarosu TCT5000 x 2
- Driver Motoare L298N x 1
- Motor cu Angrenaj x 2
- Baterie 9V x 1
- Sasiu x 1

Etapele dezvoltarii hardware:

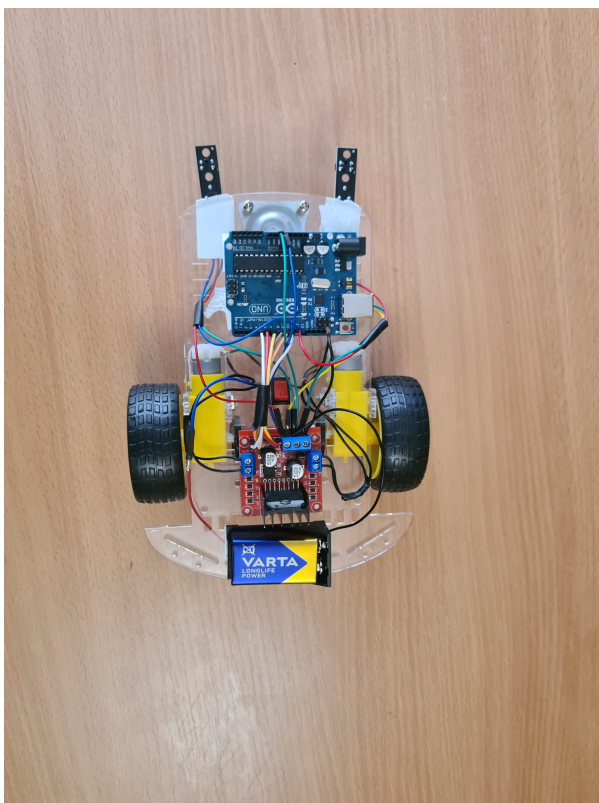
1.Asamblare Sasiu Robot



2.Montare Componente Elcetronice



3.Realizare Cablaj



Software Design

Tehnologii Utilizate

- Mediu de dezvoltare: Arduino IDE 2.1.0
- Proiectul NU foloseste librarii auxiliare

Principiul de Functionare al Senzorilor

Se monteaza doi senzori infrarosii pe extremitatile robotului.

Fiecare senzor are doua LED-uri:emitor si receptor.

Emitorul emite o raza de lumina infrarosie.

Daca raza ajunge pe o suprafata alba, care reflecta lumina, aceasta este captata de receptor.

Daca raza ajunge pe o suprafata neagra, care absoarbe lumina, aceasta nu mai este captata de receptor.

Astfel, senzorul poate determina tipul suprafetei pe care se afla.

Interactiunea cu Motoarele

Fiecare motor are asociati 3 pini: 1 pin pentru controlul vitezei si 2 pini pentru controlul directiei.

Scrierea pe pinul de viteza se face folosind **PWM**.

Interactiunea cu Senzorii

Senzorii sunt conectati la pinii analogici A2 si A4 ai placutei Arduino.

Citirea acestora se face folosind **ADC cu prescaler 128** pentru a obtine valori discrete.

Pe baza valorilor citite constant de la senzori, se iau urmatoarele decizii:

- Ambii senzori sunt pe suprafata alba $\Rightarrow$ Robotul se afla pe linie $\Rightarrow$ Se deplaseaza inainte
- Senzorul dreapta se afla pe linia neagra $\Rightarrow$ Vireaza la dreapta
- Senzorul stanga se afla pe linia neagra $\Rightarrow$ Vireaza la stanga
- Ambii senzori se afla pe suprafata neagra $\Rightarrow$ Stop

Controlul Sistemului

- Sistemul este controlat printr-un algoritim **PID:Proportional, Integrative, Derivative**. Mecanismul asigura deplasarea controlata a robotelului in timpul virajelor, lucru ce duce la un timp mai bun de parcurgere al circuitului.
- Acesta calculeaza starea curenta a robotului si pe baza acesteia calculeaza o eroare. Modul de calcul pentru eroarea curenta este:
 - daca senzor_dreapta = negru && senzor_stanga = alb $\rightarrow err = 1000$
 - daca senzor_dreapta = alb && senzor_stanga = negru $\rightarrow err = -1000$
 - daca senzor_dreapta = alb && senzor_stanga = alb $\rightarrow err = 0$
- In functie de eroare, se modifica viteza de rotatie a motorului:
 - $err = -1000 \rightarrow$ viraj stanga $\rightarrow$ cresc viteza motorului drept, scad viteza motorului stang(eventual rotesc in sens invers)
 - $err = 1000 \rightarrow$ viraj dreapta $\rightarrow$ cresc viteza motorului stang, scad viteza motorului drept(eventual rotesc in sens invers)

Rezultate Obținute

Datorita proiectului am implementat primul meu robotel. Acesta foloseste un algoritm de control PID, pentru a obtine stabilitate si o viteza mai performanta.

Proiectul poate fi imbunatatit:

- **Hardware:**
 - Proiectarea propriului sasiu; mai stabil in viraje
 - Organizarea mai buna a cablajului prin proiectarea propriilor placute integrate
- **Software:**
 - Adaugarea mai multor senzori infrarosii(8) pentru ca PID sa fie cat mai precis

- Utilizarea unor valori diferite pentru constantele K_p , K_i , K_d
- Utilizarea unor biblioteci speciale: QTRSensors

Concluzii

Mi-a placut sa implementez acest proiect, chiar daca m-am chinuit :)

Download

Arhiva cu codul Arduino al proiectului, alaturi de pozele din timpul dezvoltarii hardware se gaseste aici:[pm-popescu-silviu-333ca-2023.zip](#)

Bibliografie/Resurse

Resurse Hardware

- [Utilizare pistol de lipit](#)
- [Asamblare sasiu](#)

Resurse Software

- [Datasheet ATmega328P](#)
- [Configurare PWM](#)
- [Configurare ADC](#)
- [Algoritmul PID](#)
- [Mediu dezvoltare schema bloc](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/adarmaz/line-follower>



Last update: **2023/05/30 07:06**