



LINE FOLLOWER

Proiect PMP



POPESC ARIADNA-IOANA
GRUPA 30238

Cuprins

1. Introducere	2
1.1. Scopul și motivația proiectului.....	2
1.2. Contribuție personală	2
2. Studiu bibliografic	2
2.1. Soluția propusă.....	2
2.2. Alte soluții	2
3. Analiză și implementare	3
3.1. Descrierea generală a soluției.....	3
3.2. Descrierea algoritmilor	3
3.3. Implementarea software a proiectului	3
3.4. Implementarea hardware a proiectului	4
4. Testare și validare.....	7
5. Concluzii	7
6. Bibliografie	8

1. Introducere

1.1.Scopul și motivația proiectului

Proiectul nostru se axează pe dezvoltarea unui robot line follower, un dispozitiv capabil să navigheze autonom urmărind o linie trasată pe o suprafață. Motivația pentru acest proiect constă în dorința de a experimenta și înțelege complexitatea tehnologiilor care stau la baza roboților line follower.

Roboții line follower au devenit din ce în ce mai relevanți, având un potențial vast de aplicare în industrii precum producție, logistică și chiar în contextul educațional. Capacitatea acestora de a urmări trasee prestabilite deschide noi orizonturi în dezvoltarea tehnologiilor autonome și eficiente.

1.2.Contribuție personală

Contribuția mea personală la proiect s-a concentrat pe implementarea algoritmului de mișcare a robotului line follower, astfel încât acesta să fie capabil să urmărească linia trasată pe suprafață. Am abordat această parte a proiectului prin proiectarea și programarea unui sistem eficient de control al motoarelor, astfel încât robotul să ajusteze direcția și viteza în funcție de detecția liniei, aceasta fiind de culoare neagră, prezentă pe o suprafață albă.

2. Studiu bibliografic

2.1.Soluția propusă

Soluția propusă constă în dezvoltarea unui robot line follower cu abilități autonome de navigație pe un traseu trasat pe o suprafață. Acest proiect integrează un set de componente hardware și software pentru a asigura funcționalități complexe și adaptabilitate în fața provocărilor mediului înconjurător.

2.2.Alte soluții

În cadrul studiului bibliografic, am examinat mai multe alternative pentru implementarea robotului line follower. Prima opțiune analizată a fost controlul manual, oferind flexibilitate în ghidarea robotului, dar cu dezavantajul pierderii autonomiei. Am investigat, de asemenea, sistemele bazate pe imagini, care promit o precizie crescută, dar cu complexitate în implementare și sensibilitate la variații de lumină. Alte alternative au inclus utilizarea altor tipuri de senzori, precum infraroșu sau presiune, aducând informații suplimentare, dar cu costuri de calibrare și posibile interferențe.

3. Analiză și implementare

3.1.Descrierea generală a soluției

Proiectul constă în dezvoltarea unui robot de tip line follower, echipat cu o placă de dezvoltare JSumo XLine 16, care utilizează un array de 16 senzori pentru a urmări o linie neagră pe o suprafață albă. Această soluție combina abilitățile de percepție vizuală a robotului, bazate pe citirea continuă a valorilor senzorilor de linie, cu acțiuni motorizate corespunzătoare pentru a menține robotul aliniat cu traseul dorit. Motoarele sunt controlate în timp real în funcție de informațiile furnizate de senzorii de linie, astfel încât robotul să poată ajusta direcția de deplasare în funcție de poziția relativă a liniei detectate. În plus, soluția integrează și funcționalități suplimentare, cum ar fi indicatori LED pentru afișarea stării robotului.

3.2.Descrierea algoritmilor

Algoritmii implementați în cadrul proiectului se concentrează pe controlul și navigarea robotului de tip line follower. În funcția `check_robot_position_and_redress()`, robotul citește informațiile de la senzorii de linie, evaluați prin intermediul funcției `read_line_sensor()`. Această funcție determină dacă robotul se află într-o poziție centrală, dacă tinde să se deplaseze spre stânga sau spre dreapta pe baza valorilor citite de la senzori. În funcție de aceste evaluări, robotul primește comenzi specifice pentru a se redresa sau a urma traseul corect.

În timpul funcționării repetitive în `loop()`, algoritmii asigură ajustarea continuă a vitezei și direcției motorului în funcție de condițiile de mediu, asigurând un comportament reactiv și adaptabil al robotului în cadrul mediului său.

3.3.Implementarea software a proiectului

Configurarea inițială și inițializarea componentelor:

- În funcția `setup()`, se inițializează porturile seriale pentru debugging și se configurează pini pentru motoare, senzori, LED-uri RGB și buzzer.
- Benzile de LED-uri RGB sunt inițializate și setate la o luminozitate specifică.
- Funcția `functie_pornire()` efectuează o secvență de iluminare a LED-urilor pentru a indica starea de pornire a robotului.
- Funcția `functie_melodie()` redă o scurtă melodie pe buzzer la pornirea robotului.

Controlul motorului și funcția de oprire cu delay:

- Funcția `StartMotor()` primește patru argumente: doi pini de motor, o direcție (0/1) și o viteză. Aceasta reglează pini pentru a controla direcția și folosește modularea lățimii de impuls (PWM) pentru a ajusta viteza motorului.
- Funcția `delayStopped(ms)` oprește ambele motoare și introduce un delay specificat în milisecunde pentru a gestiona temporizarea.

Măsurarea distanței cu senzorul ultrasonic:

- Funcția `measure_distance()` utilizează un senzor ultrasonic pentru a calcula distanța până la cel mai apropiat obstacol.
- Se folosește un semnal ultrasunete pentru a măsura timpul de călătorie, iar apoi se calculează distanța în funcție de aceasta.

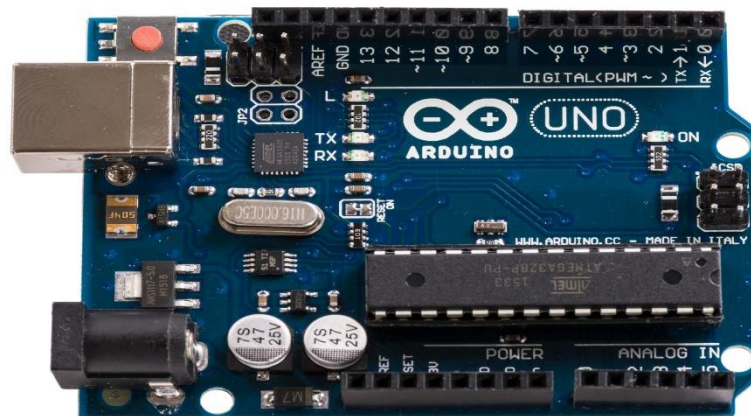
Verificarea poziției robotului și corectarea acesteia:

- Funcția `check_robot_position_and_redress()` interpretează datele de la senzorii de linie pentru a determina poziția robotului pe traseu.
- Datele de la senzori sunt citite și prelucrate în funcția `read_line_sensor()`.
- Se iau decizii privind mișcarea înainte, la stânga, la dreapta sau oprirea în funcție de datele primite de la senzorii de linie.

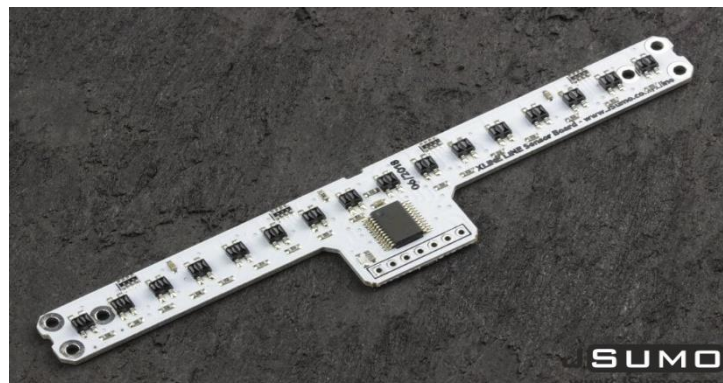
3.4.Implementarea hardware a proiectului

Componentele hardware principale utilizate pentru robot sunt:

- Placa Arduino Uno



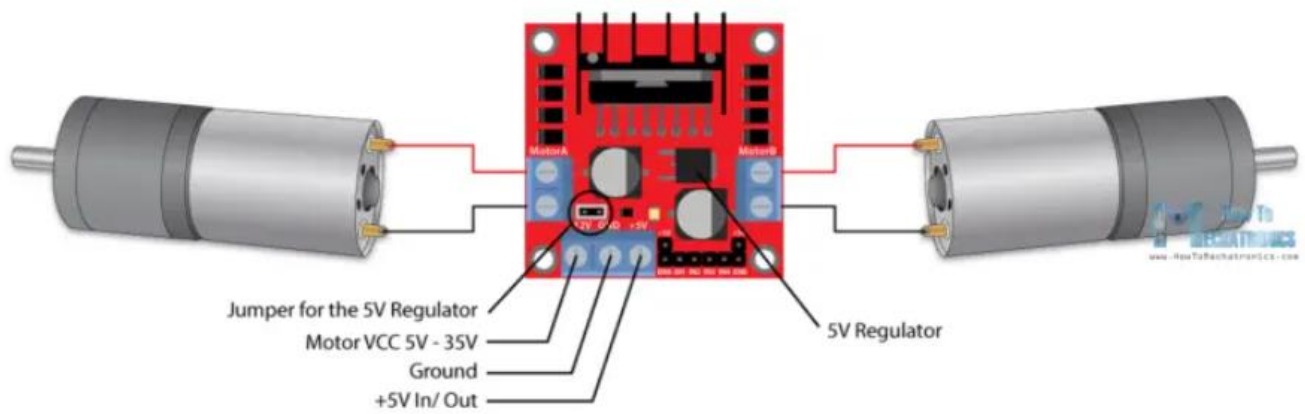
- Senzorul de linie



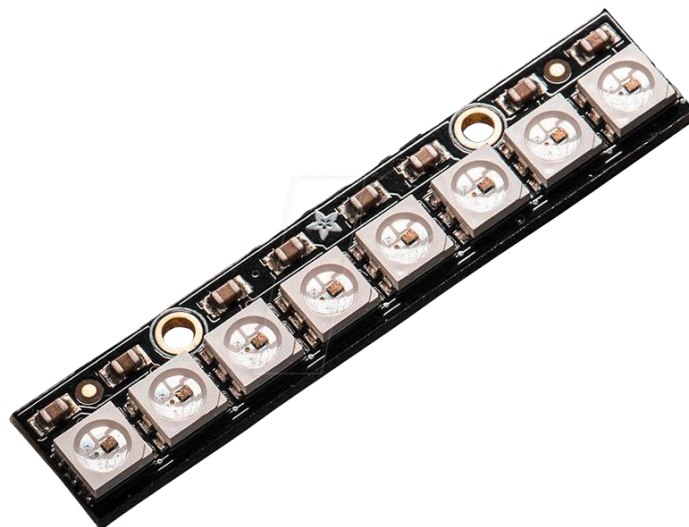
- BO motors



- L298N Motor Driver



- Neopixels



Conectarea componentelor la pinii de la Arduino:

- Pentru puntea H:
 - In1 → pin 11 (PWM)
 - In2 → pin 3 (PWM)
 - In3 → pin 6 (PWM)
 - In4 → pin 5 (PWM)
 - 5V → 5V Arduino
 - GND → GND Arduino
 - 12V → Vin Arduino
- Line sensor: (JSumo xLine 16 Sensor Array Board):
 - so → pin 7
 - s1 → pin 8
 - s2 → pin 12
 - s3 → pin 13
 - OUT → pin A0(analog)
 - + → 5V
 - - → GND
- Neopixel:
 - Di → pin10 (data IN)
 - 5V → 5V
 - GND → GND
- Potențiometrul: pin A2(analog)
- Ultrasonic sensor:
 - Vcc → 5v
 - Trig → pin4
 - Echo → pin 2
 - GND → GND

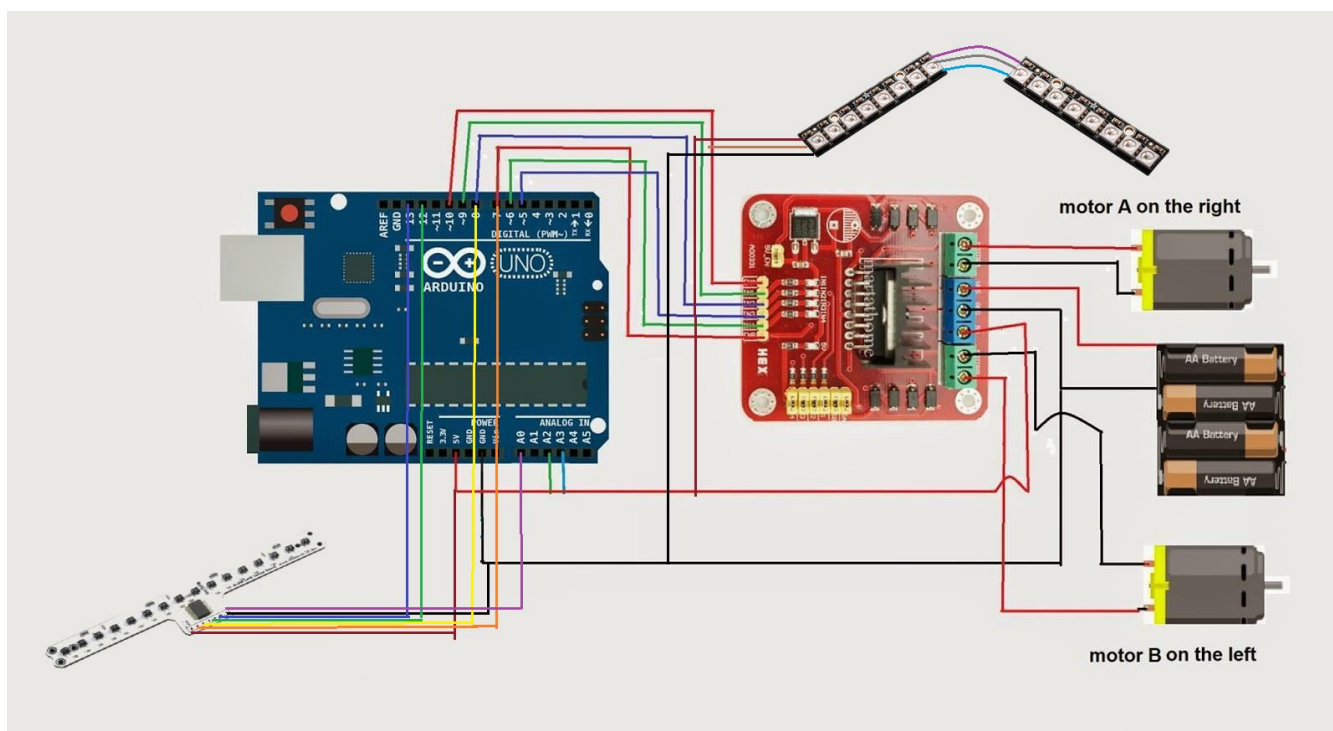
Shield-ul de pe placă are rolul de a extinde pinii și de a adăuga două rânduri de pini de alimentare(5V, GND). Dacă apar probleme la încărcarea codului, se pot foarte simplu deconecta componentele prin deconectarea shield-ului.

Pentru folosirea LED-urilor RGB care să reflecte acțiunile robotului trebuie instalată librăria “Adafruit Neopixel”.

Scopul potențiometrului era initial de a controla viteza motoarelor cu care să învârtă roțile.

Senzorul de linie este tratat ca un vector de 16 senzori. Accesăm fiecare senzor prin alternarea celor 4 switch-uri (s0, s1, s2, s3) și citim o valoare analogică de pe fiecare senzor. Valoarea citită reprezintă intensitatea luminii reflectate pe suprafață: 0 → alb, 1023 → negru, deci va putea detecta linie albă pe fundal închis sau linie neagră pe fundal deschis.

Diagrama circuitului:



4. Testare și validare

Pentru partea de testare, am început prin a lua valori de la senzorul de linie pentru a ști între ce valori se situează în funcție de suprafață. Suprafața de testare disponibilă a fost parchetul, de la care s-au luat valori între 100 și 200, în unele părți trecând chiar și de 250. Pe lângă partea de parchet citită de către senzori, s-au înregistrat și crăpăturile care aveau culori închise, înregistrându-se valori peste 500, în contextul în care 0 este considerat alb iar peste 750 va fi negru. I-am realizat un traseu pe care să-l urmăresc, însă de cele mai multe ori la întâlnirea unei crăpături ieșea de pe linia neagră (realizată dintr-o bandă de izolat neagră). Apoi, legat tot de partea de suprafață, am realizat un traseu din mai multe foi albe sub banda neagră de izolat. Valorile senzorului au scăzut, iar performanța robotului a crescut, pe aceasta suprafață urmând linia cea neagră.

5. Concluzii

În concluzie, robotul line follower nu doar oferă beneficii educaționale semnificative, ci și deschide perspective practice în diverse domenii. Un bun exemplu ar putea fi adaptarea sa pentru transportul eficient de materiale într-un mediu industrial poate contribui la îmbunătățirea eficienței operațiunilor de producție, dar și la reducerea costurilor acestora. Această tehnologie se remarcă prin potențialul său de a reduce costurile de muncă, de a se adapta la diferite medii și de a contribui la dezvoltarea continuă a soluțiilor tehnologice în automatizare.

6. Bibliografie

<https://robu.in/how-to-make-a-line-follower-robot-using-arduino-connection-code/>

<https://www.jsumo.com/xline-16-sensor-array-board>