Proiect

Muntean Marisa

Cuprins

[Ce este un line-follower robot? 1](#_Toc187238530)

[Cum funcţionează? 1](#_Toc187238531)

[Aplicaţii în diverse domenii 1](#_Toc187238532)

[Hardware 2](#_Toc187238533)

[Software 4](#_Toc187238534)

[Link Video 9](#_Toc187238535)

[Concluzii 9](#_Toc187238536)

[Bibliografie 9](#_Toc187238537)

# Ce este un line-follower robot?

Line-follower robot este un tip de robot autonom programat să urmeze o linie, de obicei marcată pe o suprafață, cum ar fi o bandă neagră pe un fundal alb sau invers.

# Cum funcţionează?

Foloseşte un set de senzori şi un algoritm de control al liniei pentru a o urmări şi a ajusta mişcările (în faţă, la dreapta, la stânga, oprire).

# Aplicaţii în diverse domenii

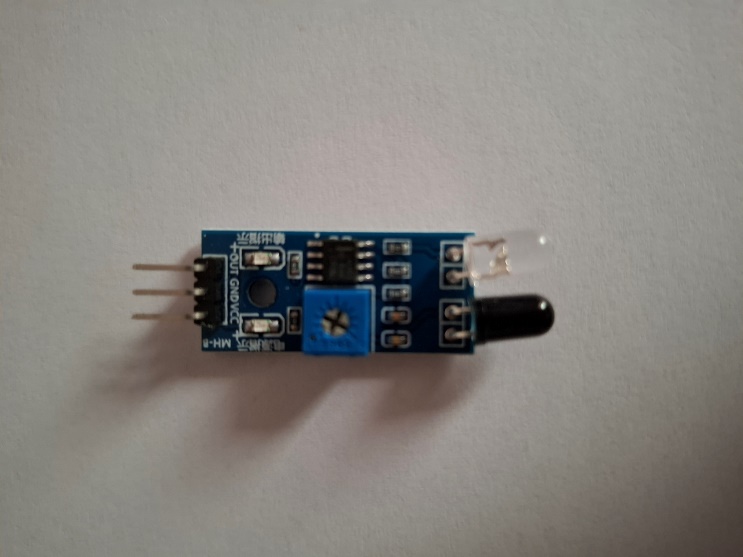
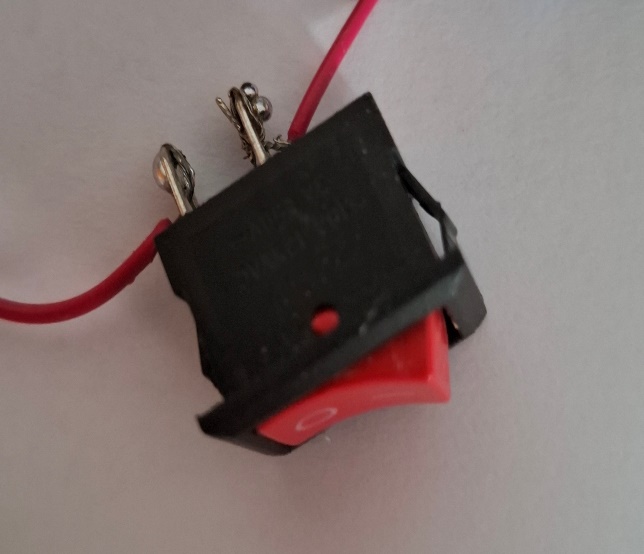
Robotul line-follower se poate aplica în diverse domenii:

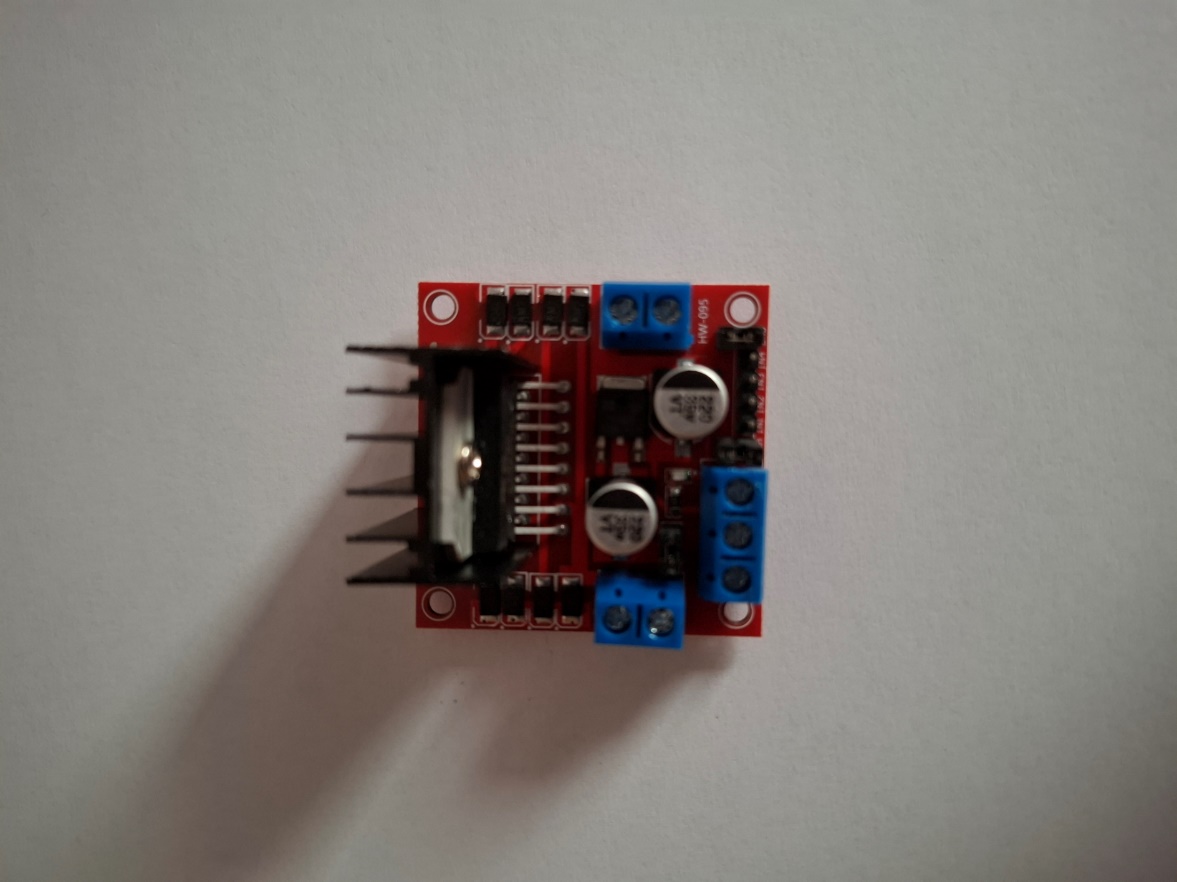
* **Automatizarea depozitelor**: Pot naviga printre rafturi pentru a transporta mărfuri, reducând necesitatea forței de muncă umane și crescând eficiența.
* **Producție**: Roboții de tip line-follower pot fi utilizați pe liniile de asamblare pentru a muta piese între diferitele etape ale producției.
* **Îngrijire medicală**: În spitale, pot transporta materiale medicale, reducând volumul de muncă al personalului și minimizând erorile.
* **Agricultură**: Acești roboți pot urma trasee predefinite pentru sarcini precum plantarea semințelor, udarea culturilor și recoltarea.
* **Produse de consum**: Unele aspiratoare și mașini de tuns iarba folosesc tehnologia de urmărire a liniei pentru a naviga în casă sau în grădină.

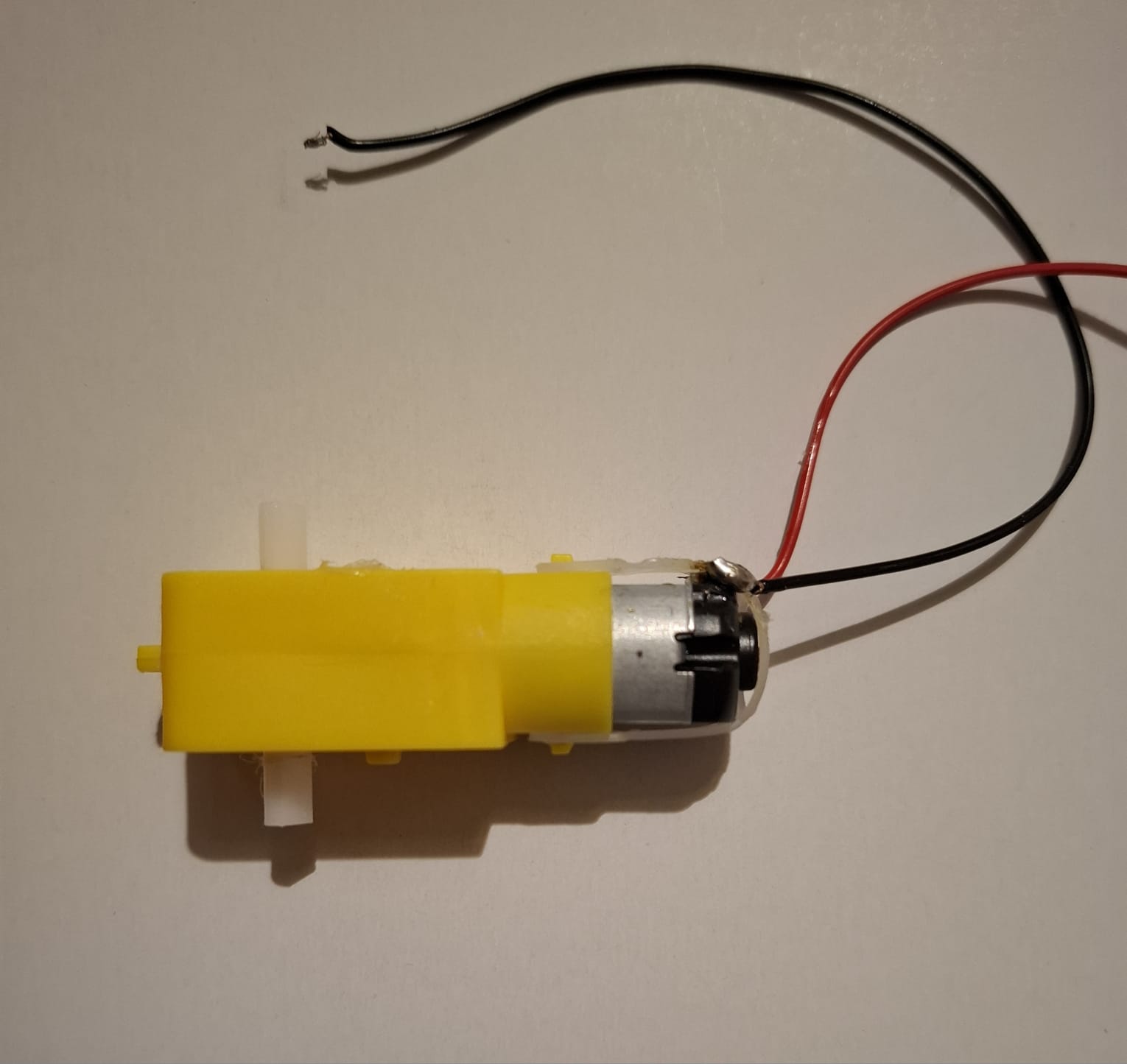
# Hardware

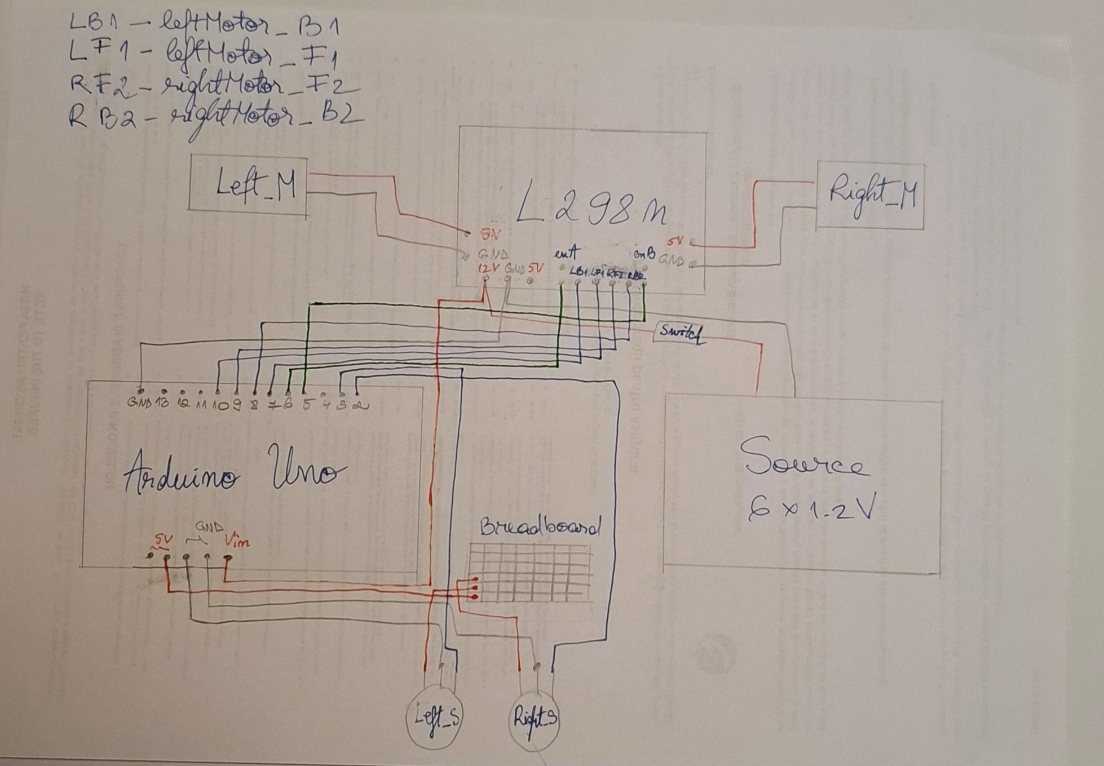
Se folosesc următoarele componente:

* 2 senzori de infraroşu
* L298n motor driver
* 2 motoare DC
* Placă Arduino UNO R3
* Placă de lemn şi şuruburi (şasiu)
* Suport 6 baterii de 1.2V
* Switch
* Fire





Schemă conexiuni: 

# Software

Cod:

#define leftMotor\_B1 7 //left part of car

#define leftMotor\_F1 10   //ok

#define rightMotor\_F2 9  //right side of car front

#define rightMotor\_B2 8

#define enB 5

#define enA 6

#define Left 3 //sensors

#define Right 2

const int speed = 180;

void setup() {

  Serial.begin(9600);

  TCCR0B = TCCR0B & B11111000 | B00000010; //a trebuit sa schimb frecventa de la PWM pentru ca la //la viteza mica motoarele nu se rotesc, iar la viteza prea mare nu pot controla masina si aceasta iese de //pe linie

  pinMode(leftMotor\_F1, OUTPUT);

  pinMode(leftMotor\_B1, OUTPUT);

  pinMode(rightMotor\_F2, OUTPUT);

  pinMode(rightMotor\_B2, OUTPUT);

  pinMode(Left, INPUT);

  pinMode(Right, INPUT);

  pinMode(enA, OUTPUT);

  pinMode(enB, OUTPUT);

  analogWrite(enA, 0);

  analogWrite(enB, 0);

}

void loop() {

  //citire valori senzori

  int LeftSensor = digitalRead(Left),

  RightSensor = digitalRead(Right);

 /\* Serial.print("Left IR sensor:");

  Serial.println(LeftSensor);

  Serial.print("Right IR sensor: ");

  Serial.println(RightSensor);

\*/

  if(LeftSensor == 0 && RightSensor == 0)

  {

//inainte

    rotateMotor(speed,speed);

  }

  else if(LeftSensor == 1 && RightSensor == 0)

  {

    //stanga

    rotateMotor(-speed,speed);//rotire cu spatele la motorul din dreapta

  }

  else if(LeftSensor == 0 && RightSensor == 1)

  {

//dreapta

       rotateMotor(speed,-speed); //rotire cu spatele la motorul din stanga

  }

  else  //(LeftSensor == 1 && RightSensor == 1)

  {

    //stop

    rotateMotor(0,0);

  }

}

void rotateMotor(int rightMotorSpeed,int leftMotorSpeed )

{

    if (rightMotorSpeed <0)

    {

      digitalWrite(rightMotor\_F2,LOW);

      digitalWrite(rightMotor\_B2,HIGH);

    }

    else   if (rightMotorSpeed >0)

    {

      digitalWrite(rightMotor\_F2,HIGH);

      digitalWrite(rightMotor\_B2,LOW);

    }

    else{

      digitalWrite(rightMotor\_F2,LOW);

      digitalWrite(rightMotor\_B2,LOW);

    }

    if (leftMotorSpeed <0)

    {

      digitalWrite(leftMotor\_F1,LOW);

      digitalWrite(leftMotor\_B1,HIGH);

    }

    else   if (leftMotorSpeed >0)

    {

      digitalWrite(leftMotor\_F1,HIGH);

      digitalWrite(leftMotor\_B1,LOW);

    }

    else{

      digitalWrite(leftMotor\_F1,LOW);

      digitalWrite(leftMotor\_B1,LOW);

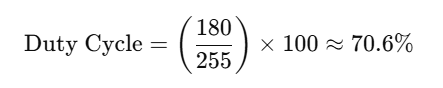
    }

    analogWrite(enB, abs(rightMotorSpeed));

  analogWrite(enA, abs(leftMotorSpeed));

/\*

Motorul functioneaza la 71% din puterea maxima



\*/

}

# Link Video

<https://youtu.be/xY4Rc8MPFNA>

# Concluzii

* **Un motor driver pentru 2 motoare este într-adevăr pentru 2 motoare, nu pentru 4!** În urma acestui proiect am învăţat ca datasheet-ul trebuie citit înainte(nu este destul să vezi doar tutoriale), altfel motoarele şi driver-ul nu functioneaza conform asteptarilor(sunete ciudate produse de motoare, dar si de driver; masina se bloca la curbe cu toate ca senzorii detectau ce trebuie)
* **În momentul în care Arduino primeşte curent doar de la baterii, trebuie folosit pinul de Voltage In(Vin).**
* **Modificarea vitezei se face cu ajutorul lui PWM.** În loc de a trimite un curent constant (analogic), PWM folosește un semnal digital (care poate fi doar ON sau OFF) pentru a "simula" diferite niveluri de putere. Prin schimbarea duratei în care semnalul este ON față de timpul total al unui ciclu, poți controla cât de multă energie ajunge la dispozitiv. De asemenea, cu un Duty Cycle mai mare, motorul primește mai multă putere și se învârte mai repede.
* **Am învăţat să utilizez pistolul de lipit!**

# Bibliografie

<https://www.youtube.com/watch?v=5jh-5HGvC-I&t=355s&ab_channel=hashincludeelectronics>

<https://circuitdigest.com/microcontroller-projects/arduino-uno-line-follower-robot>

<https://youtu.be/jvKanmkshrs?si=f24J__We7iqiuFCM>