Line Follower

- Student: **Balanica Ciprian**
- Grupa: **333CC**

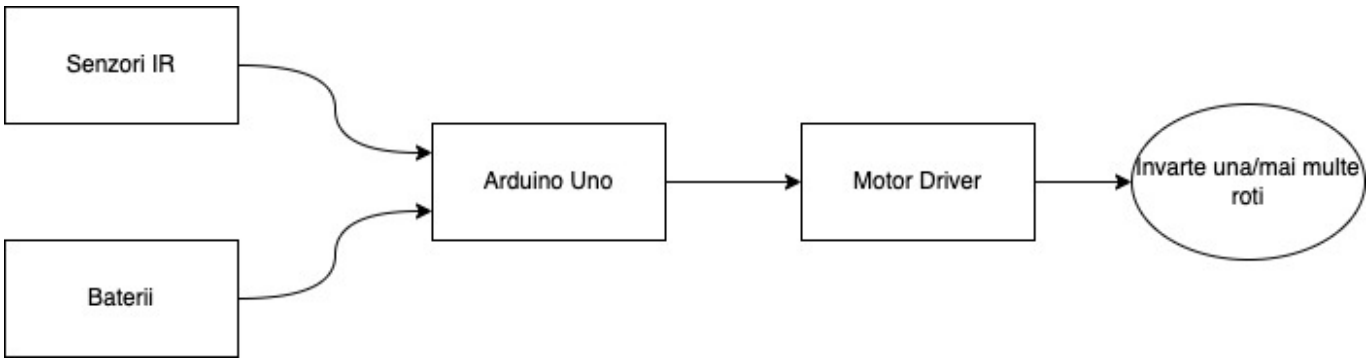
Introducere

Un Line Follower este un robot care urmareste o linie desenata pe o suprafata astfel incat sa formeze un contrast mare (de obicei linie neagra pe suprafata alba sau foarte deschisa), cu ajutorul unor senzori infrarosu. Senzorii sunt pozitionati in stanga si in dreapta liniei iar robotul merge drept pana cand unul dintre ei detecteaza linia, semnificand curbarea acesteia. Masina vireaza in functie de senzorul care s-a declansat.

Idea initiala era pentru o masina controlata prin telecomanda, dar deoarece acea idee fusese deja implementata de multe ori, am decis sa o modific facand-o cumva sa se miste automat. Consider ca acest robot ar putea fi o provocare interesanta incat desi poate un prototip de baza ar putea fi implementat rapid, ar putea fi facute in continuare foarte multe optimizari si imbunatatiri.

Descriere generală

Schema Bloc



Hardware Design

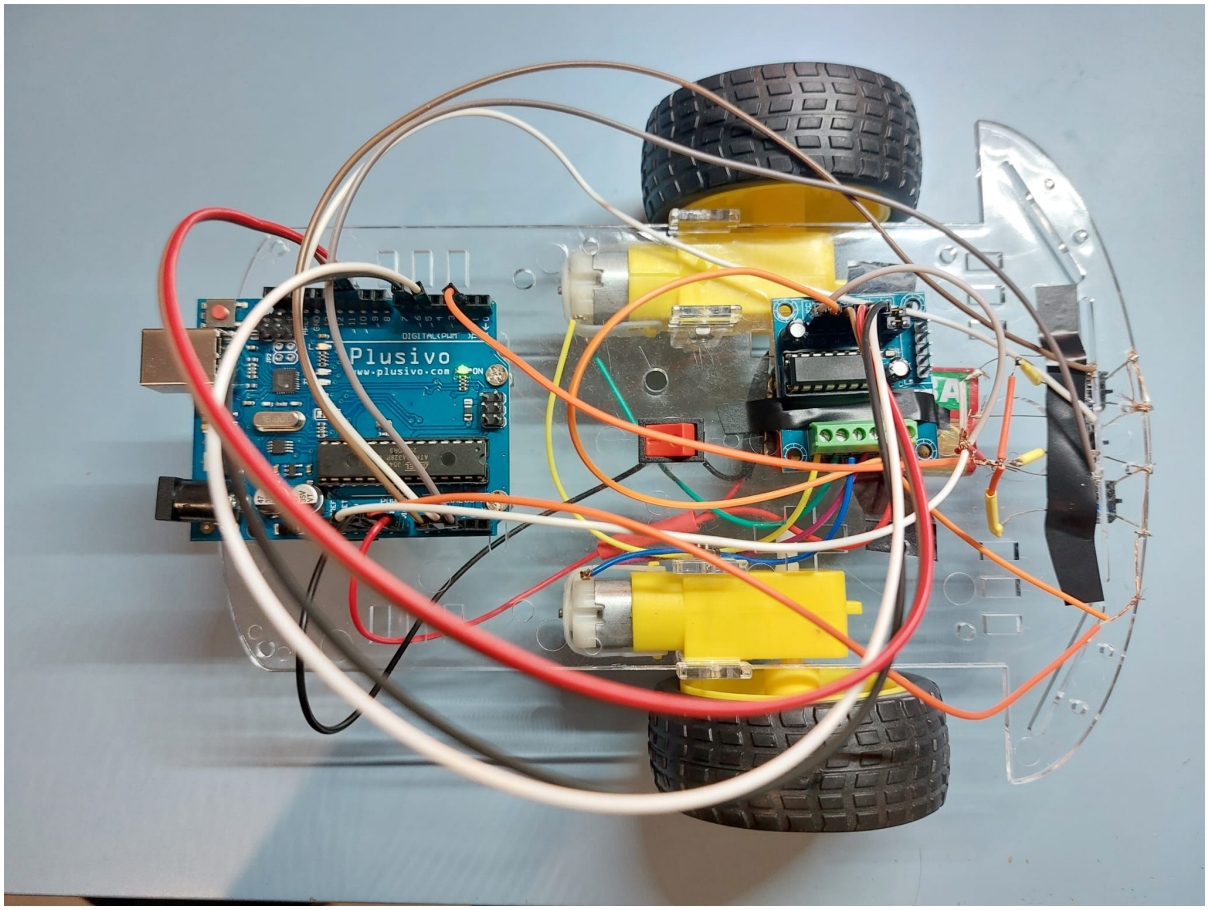
Piese folosite

- Arduino Uno
- 3 senzori IR TCRT5000
- L293D motor driver
- 2 motoare
- 2 roti + roata de suport
- Baterie 9V
- Cadru masina
- Fire
- Rezistente
- Switch

Schema Electrica

(LED-urile si senzorii sunt aratati separat dar functioneaza ca o singura componenta. LED urile emit lumina IR, iar senzorii o receptioneaza)

Circuit



Software Design

Softul line followerului urmareste un algorimt clasic pentru acest tip de robot, numit PID (Proportional-Integral-Derivative). Acest program implementeaza doar partea de PD, partea Integrala fiind folosita in general la roboti mai avansati pentru precizie.

In functia de setup initializam toate varibilele si pinii. In loop avem doua etape, cea de calibrare si cea de cursa. Etapa de calibrare este rulata o singura data, dupa care se sare peste ea si se apeleaza direct partea de cursa.

In etapa de calibrare robotul trece cu toti cei 3 senzori peste linie si peste spatiul alb, pentru a afla valorile lor minime si maxime in mediul curent. Putem seta directia in care se va invarti robotul, asa ca il punem sa se invarte la stanga si la dreapta de mai multe ori pentru a fi siguri ca a acoperit tot terenul.

```
if (!calibrated) {
  delay(1000);
  Serial.println("Calibrating...");
  calibrate(0);
  calibrate(1);
  calibrate(0);
  calibrate(1);

  calibrated = 1;
```

```
  Serial.println("Calibration done!");

  delay(4000);  // Delay to allow the user to place the robot on the line

  Serial.println("Race...");
}
...
void calibrate(int dir) {
  ms = millis();

  if (dir == 0) {
    motorSpeed(-500, 500);
  } else {
    motorSpeed(500, -500);
  }

  while ((ms + CALIBRATION_TIME) > millis()) {
    int sens_value[N_SENS];
    for (int x = 0; x < N_SENS; x++) {
      sens_value[x] = analogRead(SENS[x]);
      sens_min[x] = (sens_value[x] < sens_min[x]) ? sens_value[x] : sens_min
[x];
      sens_max[x] = (sens_value[x] > sens_max[x]) ? sens_value[x] : sens_max
[x];
    }
  }
  motorSpeed(0, 0);
  delay(100);
}
```

In partea de cursa, se verifica pozitia actuala a liniei, si impreuna cu pozitia anterioara se calculeaza corectia PD:

```
float get_PD_correction(float line, float last_line, float kp, float kd) {
  float proportional = line;
  float derivative = line - last_line;
  float correction = (kp * proportional + kd * derivative);

  return correction;
}
```

Linia este calculata simplu prin atribuirea unor greutati fiecarui senzor, care mai apoi sunt insumati. In functie de aceasta corectie, decidem daca masina va vira la stanga sau la dreapta, setand viteze motoarelor.

```
void motorSpeed(int m1, int m2) {  // From -1000 to 1000
  int pwm1 = map(abs(m1), 0, 1000, 0, 255);
  int pwm2 = map(abs(m2), 0, 1000, 0, 255);
  pwm1 = (m1 > 0) ? 255 - pwm1 : pwm1;
  pwm2 = (m2 >= 0) ? pwm2 : 255 - pwm2;
```

```
  analogWrite(MA2, pwm1);
  analogWrite(MB2, pwm2);
  digitalWrite(MA1, (m1 > 0) ? HIGH : LOW);
  digitalWrite(MB1, (m2 >= 0) ? LOW : HIGH);
}
```

Concluzii

Desi la inceput parea ca ar fi un proiect rapid de implementat, am realizat ca pot aparea multe probleme neasteptate. Deoarece un breadboard ar fi fost prea greoi si ar fi ocupat prea mult spatiu, am fost nevoit sa leg firele intre ele si sa le lipesc prin anumite locuri. De asemenea, alimentarea a fost o problema majora. Majoritatea bateriilor erau prea mici pentru a alimenta driverul si motoarele, dar cele de 9V erau consumate foarte repede. In cod am fost nevoit sa modific anumite valori pentru a calibra robotul. In final consider ca am avut multe de invatat din acest proiect si ca debuggingul de hardware a reprezentat o provocare interesanta, diferita de uzualul debugging de software.

Download

line-follower-archive.zip

Bibliografie/Resurse

- laboratoare ocw
- forum arduino (pentru probleme tehnice)
- datasheeturi
- https://www.instructables.com/Line-Follower-Robot-With-Arduino-Really-Fast-and-R/
- https://circuitdigest.com/microcontroller-projects/arduino-uno-line-follower-robot

Export to PDF