Obstacle Avoiding Car

Autor

Ioniță Dragoș 332CB, e-mail: dragos.ionita2303@stud.acs.upb.ro

Introducere

Proiectul constă într-o mașinuță cu două motoare, cu senzori de viteză a rotației celor două motoare și capacitatea de a urma un traseu predefinit la pornirea mașinuței (la alimentarea acesteia sau la apăsarea butonului RESET de pe Arduino).

Pe parcursul traseului, mașinuța va fi capabilă să identifice obstacolele din fața sa și să se oprească, apoi să-și schimbe direcția de mers pentru a evita coliziunea.

Descriere generală

Mașina nu va fi controlată remote / bluetooth / wi-fi, ci va porni la apăsarea unuia din butoanele corespunzătoare pentru un traseu predefinit.

Ideea de la care am pornit a fost folosirea de senzori și întreruperi hardware pentru a crea ceva ce se aseamănă cât mai mult cu o mașină robot 'inteligentă'.

Mașina va fi pre-programată software pentru a urma orice traseu, pe orice direcție, cu viteze diferite, putând astfel simula 'coregrafii', urmând diverse pattern-uri și mișcări stânga-dreapta, înainte-înapoi, inclusiv rotiri în jurul propriei axe.

Conexiunile GND și VCC pentru senzorii fotoelectrici LM393 ai motoarelor

Senzorii vor fi conectați la motoare, pe șasiul mașinuței și vor măsura viteza de rotație a roților motoare.

Schema completă fără senzorul ultrasonic de distanță HC-SR04

Prin intermediul driverului L298N, se vor putea controla motoarele pentru a putea predefini traseele de urmat, iar distanțele și vitezele vor fi calculate cu ajutorul output-ului senzorilor și trimise către motoare prin intermediul driverului L298N.

Cablajele fără senzorul ultrasonic de distanță HC-SR04

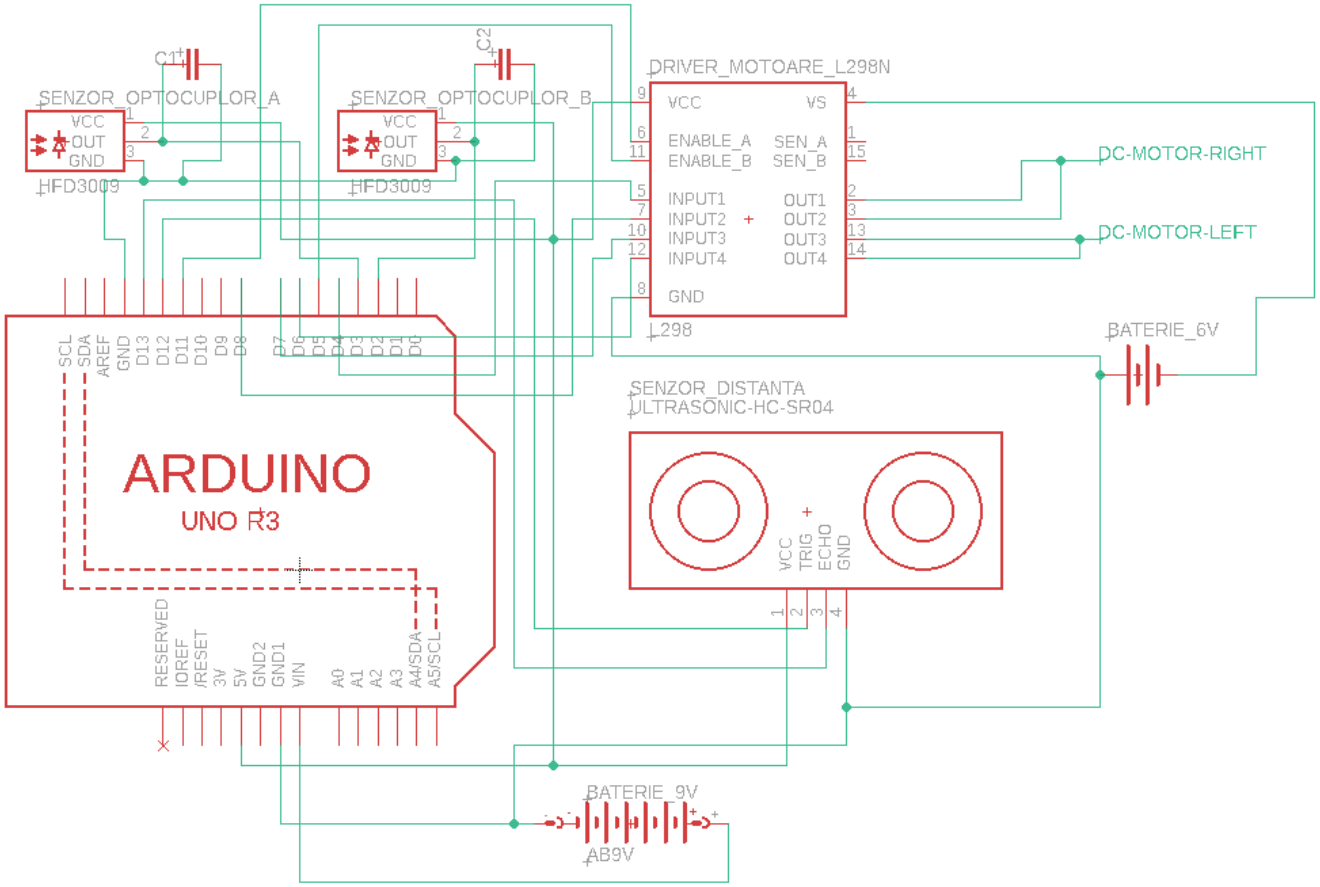
Schema bloc logică cu senzorul de distanță HC-SR04

Va exista si senzorul HC-SR04 ultrasonic pentru măsurarea distanței, care va fi montat în partea din față a mașinuței. Acesta va transmite distanța până la un eventual obstacol din față, iar mașinuța va reacționa, fie prin oprire și schimbarea direcției de mers, fie prin ocolirea acestuia și continuarea drumului.

Hardware Design

- Plăcuță Plusivo compatibilă Arduino Uno https://ardushop.ro/ro/home/29-placa-de-dezvoltare-uno-r3.html?search_query=arduino+uno+plusivo&results=243
- Kit Șasiu Mașinuță Inteligentă cu 2 motoare cu angrenaje https://cleste.ro/sasiu-transparent-masina-inteligenta.html
  - 1 x șasiu auto
  - 2 x motoare cu angrenaje
  - 2 x roți
  - 2 x cleme fixare
  - 1 x roată universală
  - 1 x suport de baterii (bateriile nu sunt incluse)
  - Toate șuruburile și piulițele necesare
- Driver motoare (modul L298N cu punte H dublă) https://cleste.ro/modul-l298n-cu-punte-h-dubla.html
- Doi senzori fotoelectrici cu infraroșu L393M https://cleste.ro/senzor-fotoelectric-cu-infraro-u.html

- Senzor ultrasonic HC-SR04 https://cleste.ro/senzor-ultrasonic-hc-sr04.html
- 2 x condensator ceramic 10nF și 2 x condensator ceramic 3,3nF https://ardushop.ro/ro/electronica/714-condensator-ceramic-50v-2pf-220nf.html
- Breadboard mini pentru interconectare mai facilă https://cleste.ro/breadboard-mini-170-puncte.html
- 4 x baterii de 1.5V pentru motoarele mașinuței (4 x baterii clasice de 1,5 V)
- Baterie de 9V pentru alimentarea Arduino + conector pentru Arduino (baterie clasică Alkaline de 9V)
- Fire mamă-tată și tată-tată https://cleste.ro/10-x-fire-dupont-tata-tata-10cm.html și https://cleste.ro/10xfire-dupont-mama-tata-20cm.html
- Unul sau două led-uri https://cleste.ro/led-rgb-5mm-4pini.html



Componente auxiliare pentru montare și alimentare

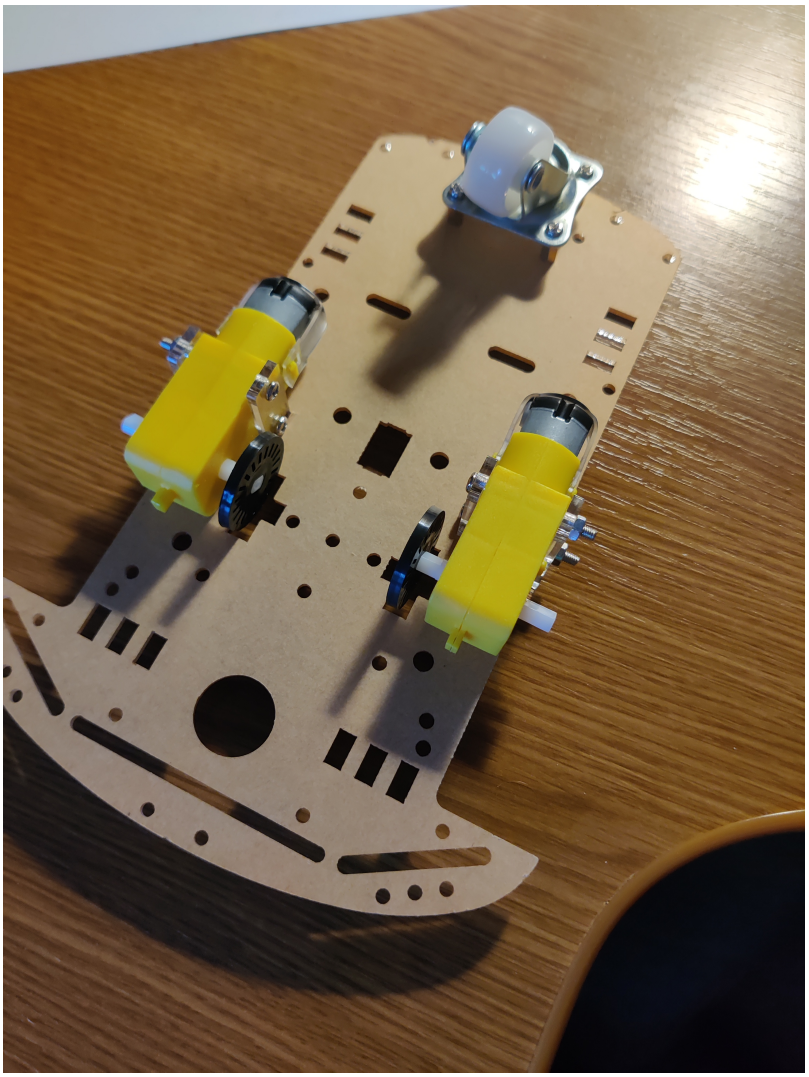
- Pistol de lipit (pentru firele alimentării motoarelor)
- Șuruburi și piulițe (1,5mm; 2mm; 2,5mm; 3mm) pentru fixarea tuturor componentelor pe șasiul de plexiglas
- Pistol cu super-glue (pentru fixarea senzorilor)
- Mini-bormașină (pentru a da găuri în șasiu, pentru o mai bună fixare a componentelor pe el)

Software Design

- Mediu de dezvoltare și testare: Arduino IDE
- Planificare pași implementare și testare software:
  - 1. Senzorii fotoelectrici LM393
  - 2. Senzorul fotoelectric HC-SR04
  - 3. Driverele motoare L298N
  - 4. Implementarea pre-programării urmăririi traseelor
  - 5. Implementarea evitării obstacolelor, fie prin oprire și schimbarea direcției, fie prin ocolire

Jurnal

- 14 Mai 2021
  - Montare motoare și roata ajutătoare pe șasiu



- 16 Mai 2021
  - Comanda 2 x condensator ceramic de 10nF și 2 x condensator ceramic de 3,3nF

- Montare senzori fotoelectrici LM393 și interfațarea acestora cu Arduino.
- În urma testării acestora după cum se vede în următoarele videoclip-uri [4], roțile encodoare ale motoarelor ce conțin 20 de fante se învârt trecând printre stâlpii optocuplatorului H2010 al senzorului fotoelectric cu comparator LM393.
- Conform specificațiilor motoarelor, la alimentarea acestora la 6V (am alimentat la 4 x baterii de 1,5V în videoclip), acestea se învârt la aproximativ 200-230 RPM.
- După cum se vede în screenshot-urile atașate, senzorii măsoară mult mai mult, undeva la 2700-2800 RPM, calculat după formula RPM = (counter / diskslots) * 60.00 , unde counter este de câte ori se întrerupe fascicolul infraroșu al optocuplatorului, diskslots sunt numărul de fante din roțile encodoare ale motoarelor (20 în cazul meu), iar * 60.00 pentru a transforma in Rotații Pe Minut (RPM).
- Diferența survine din cauza că senzorul cu comparator LM393 este foarte sensibil și, aparent, declanșează mult mai multe întreruperi pe RISING EDGE al semnalului digital OUT decât cele declanșate în mod real prin obstrucția razei infraroșii a optocuplatorului.
- Acest senzor este foarte sensibil la interferențele care pot fi introduse între pinii VCC și GND. Dacă alimentăm senzorul de la Arduino cu 5V, regulatorul de tensiune al Arduino poate introduce curenți de fugă în senzor, ceea ce duce la declanșarea mult mai multor întreruperi pe rising edge al semnalului digital OUT (sursa: [3]).
- În acest sens, am comandat condensatori ceramici de 3,3 si 10 nF, pentru a-i lipi între pinii GND și OUT ai senzorilor, pentru a netezi semnalele și a mitiga problema.

- 21 Mai 2021
  - Lipire condensatoare pe unul din senzori și testarea unui motor, dacă se ajunge cu măsurătorile la 200-230 RPM, valoarea optimă pentru motorul DC de 6V (video la link-ul [5]).
  - Comanda condensatoare ceramice mai mici, de 2 pF si de 10pF, în speranța că măsurătorile senzorilor vor fi cele bune, de 200-230 RPM, întrucât cu condensatoarele de 3,3 nF si de 10 nF, acestea dau aproximativ 450 RPM (cum se vede în link-ul [5]), dublu față de cât trebuiau să dea, însă un progres față de 2800 RPM, cât măsurau înainte de a folosi condensatoare (link-ul [4]).
  - Testare senzor ultrasonic de distanță HC-SR04, cu ajutorul librăriei New Ping, care ajută setup-ul software pentru măsurarea precisă a distanțelor (eroare de 1, 2 centrimetri, suprafețele folosite nu sunt perfect plane și nici senzorul nu este perfect perpendicular pe direcția normalelor acestora, însă este suficient de precis pentru scopul acestui proiect), video-ul la link-ul [6]. Codul folosit pentru calculul distanței măsurate de senzor este:

```
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
  sound_speed = (331.4 + 0.606 * TEMP + 0.0124 * HUMID) / 10000.0;
}

void loop() {
  // put your main code here, to run repeatedly:
  duration = sonar.ping(); // get the duration between sending and receiving
the signal on the sensor pins

  distance = (duration / 2) * sound_speed ; // 343 m/s, speed of light, in
transformed in cm/ms

  Serial.print("Distance = ");
  if (distance >= 400 || distance <= 2) {
```

```
    Serial.println("OUT OF RANGE!\n");
  } else {
    Serial.print(distance);
    Serial.println(" cm");
    delay(500);
  }
  delay(500);
}
```

Rezultate Obţinute

Pentru video-ul de la link-ul [4], pozele cu măsurătorile în urma demo-ului, unde se măsoară (eronat) 2800 RPM, din cauza curenților de fugă existenți în circuitul intern Arduino (nu folosisem condensatori pentru stabilizare aici):

Pentru video-ul de la link-ul [5], poza cu măsurătoarea îmbunătățită, de 450 RPM, pentru unul din motoare (dar încă eronate, față de referința de 200-230 RPM conform catalogului motoarelor). Aici am folosit condensatori de 10 nF. O posibilă îmbunătățire este folosirea unor condensatori de capacitate mai mică (pF):

COM4

```
22:49:22.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 462.00 RPM
22:49:23.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 459.00 RPM
22:49:24.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 468.00 RPM
22:49:25.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:26.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 465.00 RPM
22:49:27.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 453.00 RPM
22:49:28.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 465.00 RPM
22:49:29.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 471.00 RPM
22:49:30.336 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 459.00 RPM
22:49:31.336 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 471.00 RPM
22:49:32.299 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 471.00 RPM
22:49:33.299 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:34.302 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 462.00 RPM
22:49:35.307 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:36.308 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:37.308 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 459.00 RPM
22:49:38.309 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 462.00 RPM
22:49:39.309 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:40.321 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 450.00 RPM
22:49:41.321 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 459.00 RPM
22:49:42.321 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 459.00 RPM
22:49:43.322 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 453.00 RPM
22:49:44.322 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 456.00 RPM
22:49:45.323 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 465.00 RPM
22:49:46.323 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 228.00 RPM
22:49:47.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:48.334 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:49.302 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:50.304 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:51.304 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:52.305 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:53.315 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:54.315 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:55.314 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
22:49:56.312 -> Motor Speed 1: 0.00 RPM --- Motor Speed 2: 0.00 RPM
```

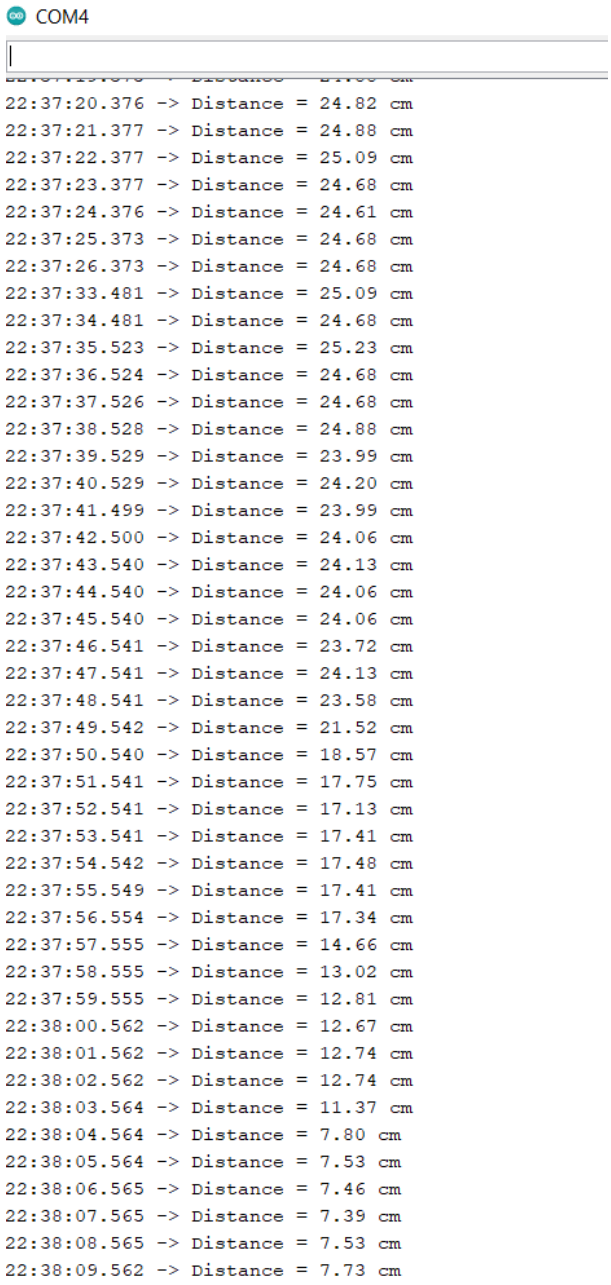
Pentru video-ul de la link-ul [6], poza cu măsurătoarea distanței cu senzorul ultrasonic HC-SR04, pentru 30cm, 25cm, 20cm, 15cm și 10cm (rigla din video pentru referință). Se observa erori de 1, 2 cm când se stabilieză poziția senzorului pentru fiecare din cele 5 valori de mai sus:

COM4

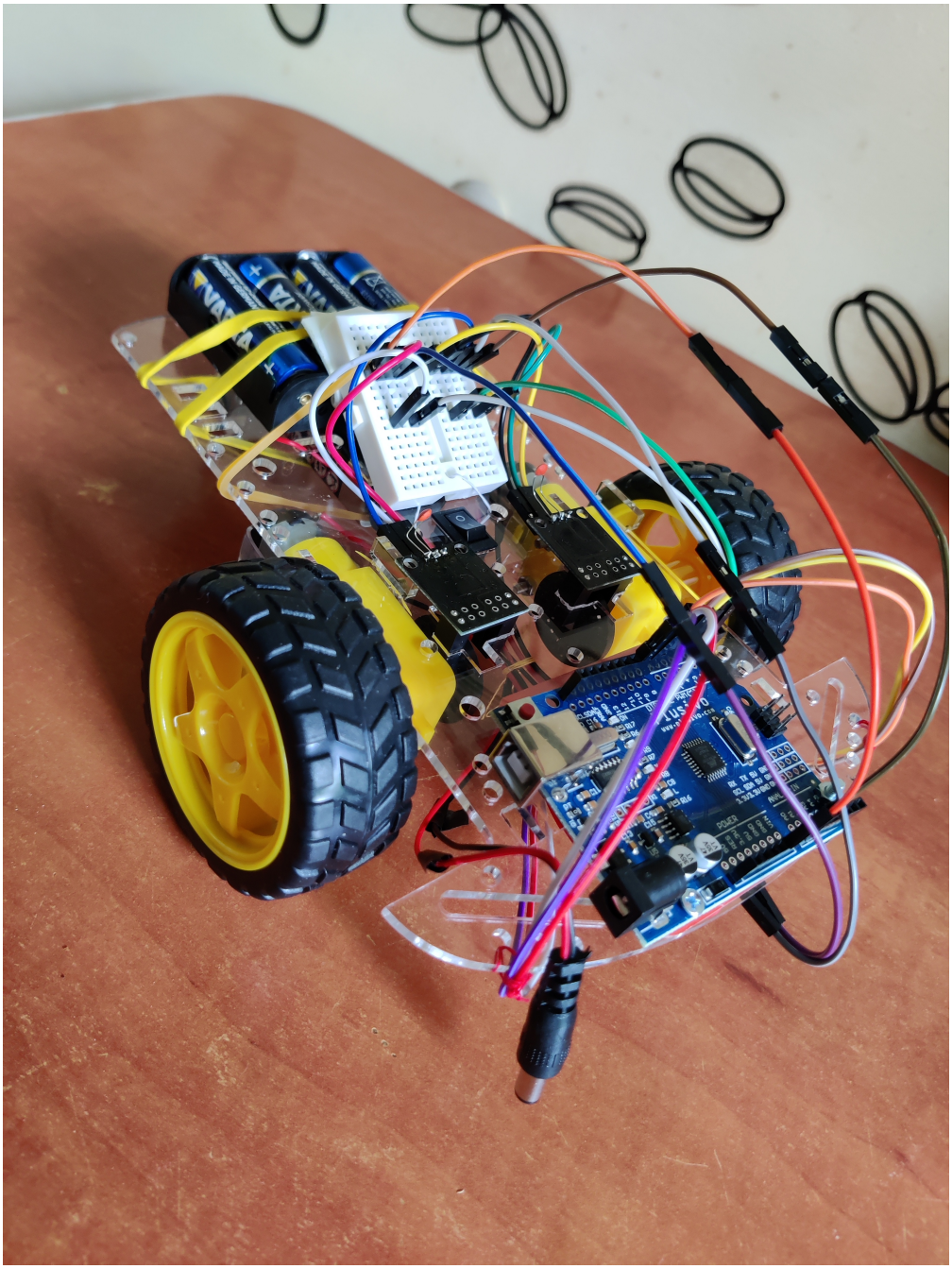
```
22:36:27.242 -> Distance = 30.24 cm
22:36:28.244 -> Distance = 30.65 cm
22:36:29.243 -> Distance = 30.17 cm
22:36:30.243 -> Distance = 30.58 cm
22:36:31.243 -> Distance = 30.65 cm
22:36:32.243 -> Distance = 30.58 cm
22:36:33.243 -> Distance = 30.58 cm
22:36:34.244 -> Distance = 30.58 cm
22:36:35.245 -> Distance = 30.10 cm
22:36:36.284 -> Distance = 30.58 cm
22:36:37.284 -> Distance = 30.24 cm
22:36:38.285 -> Distance = 30.17 cm
22:36:39.290 -> Distance = 30.24 cm
22:36:40.293 -> Distance = 30.24 cm
22:36:41.293 -> Distance = 30.17 cm
22:36:42.293 -> Distance = 30.17 cm
22:36:43.294 -> Distance = 30.17 cm
22:36:44.293 -> Distance = 30.24 cm
22:36:45.293 -> Distance = 30.65 cm
22:36:46.293 -> Distance = 30.17 cm
22:36:47.293 -> Distance = 30.58 cm
22:36:48.294 -> Distance = 30.10 cm
22:36:49.294 -> Distance = 30.58 cm
22:36:50.304 -> Distance = 30.58 cm
22:36:51.304 -> Distance = 30.10 cm
22:36:52.307 -> Distance = 29.76 cm
22:36:53.307 -> Distance = 30.24 cm
22:36:54.307 -> Distance = 30.17 cm
22:36:55.302 -> Distance = 30.10 cm
22:36:56.305 -> Distance = 30.24 cm
22:36:57.305 -> Distance = 30.24 cm
22:36:58.345 -> Distance = 29.89 cm
22:36:59.348 -> Distance = 22.62 cm
22:37:00.350 -> Distance = 23.38 cm
22:37:01.351 -> Distance = 23.17 cm
```

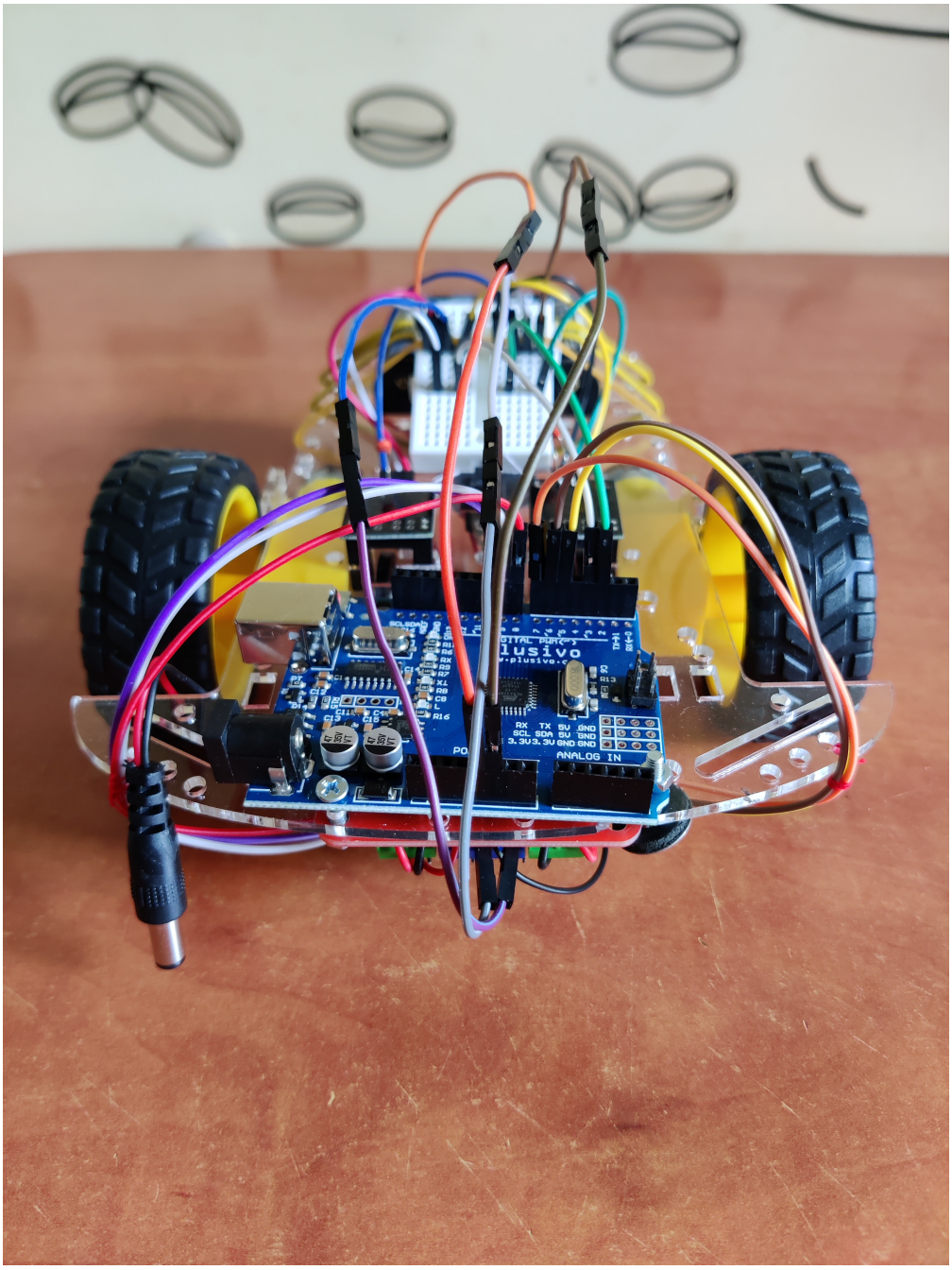
COM4

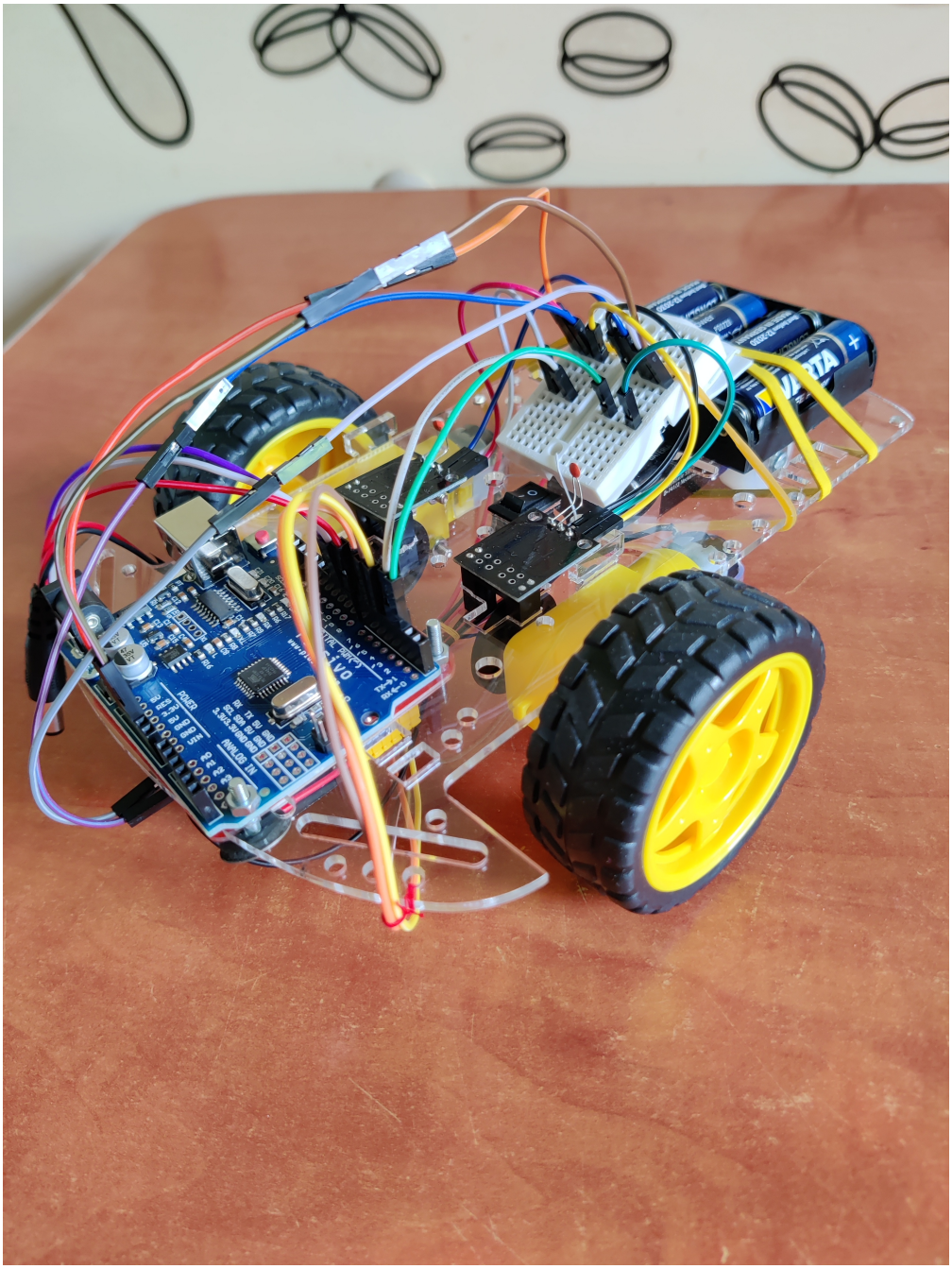
```
22:37:09.354 -> Distance = 23.17 cm
22:37:10.363 -> Distance = 23.51 cm
22:37:11.363 -> Distance = 23.99 cm
22:37:12.363 -> Distance = 23.38 cm
22:37:13.363 -> Distance = 23.38 cm
22:37:14.364 -> Distance = 23.38 cm
22:37:15.364 -> Distance = 24.34 cm
22:37:16.374 -> Distance = 24.75 cm
22:37:17.374 -> Distance = 24.88 cm
22:37:18.375 -> Distance = 25.30 cm
22:37:19.375 -> Distance = 24.88 cm
22:37:20.376 -> Distance = 24.82 cm
22:37:21.377 -> Distance = 24.88 cm
22:37:22.377 -> Distance = 25.09 cm
22:37:23.377 -> Distance = 24.68 cm
22:37:24.376 -> Distance = 24.61 cm
22:37:25.373 -> Distance = 24.68 cm
22:37:26.373 -> Distance = 24.68 cm
22:37:33.481 -> Distance = 25.09 cm
22:37:34.481 -> Distance = 24.68 cm
22:37:35.523 -> Distance = 25.23 cm
22:37:36.524 -> Distance = 24.68 cm
22:37:37.526 -> Distance = 24.68 cm
22:37:38.528 -> Distance = 24.88 cm
22:37:39.529 -> Distance = 23.99 cm
22:37:40.529 -> Distance = 24.20 cm
22:37:41.499 -> Distance = 23.99 cm
22:37:42.500 -> Distance = 24.06 cm
22:37:43.540 -> Distance = 24.13 cm
22:37:44.540 -> Distance = 24.06 cm
22:37:45.540 -> Distance = 24.06 cm
22:37:46.541 -> Distance = 23.72 cm
22:37:47.541 -> Distance = 24.13 cm
22:37:48.541 -> Distance = 23.58 cm
22:37:49.542 -> Distance = 21.52 cm
```

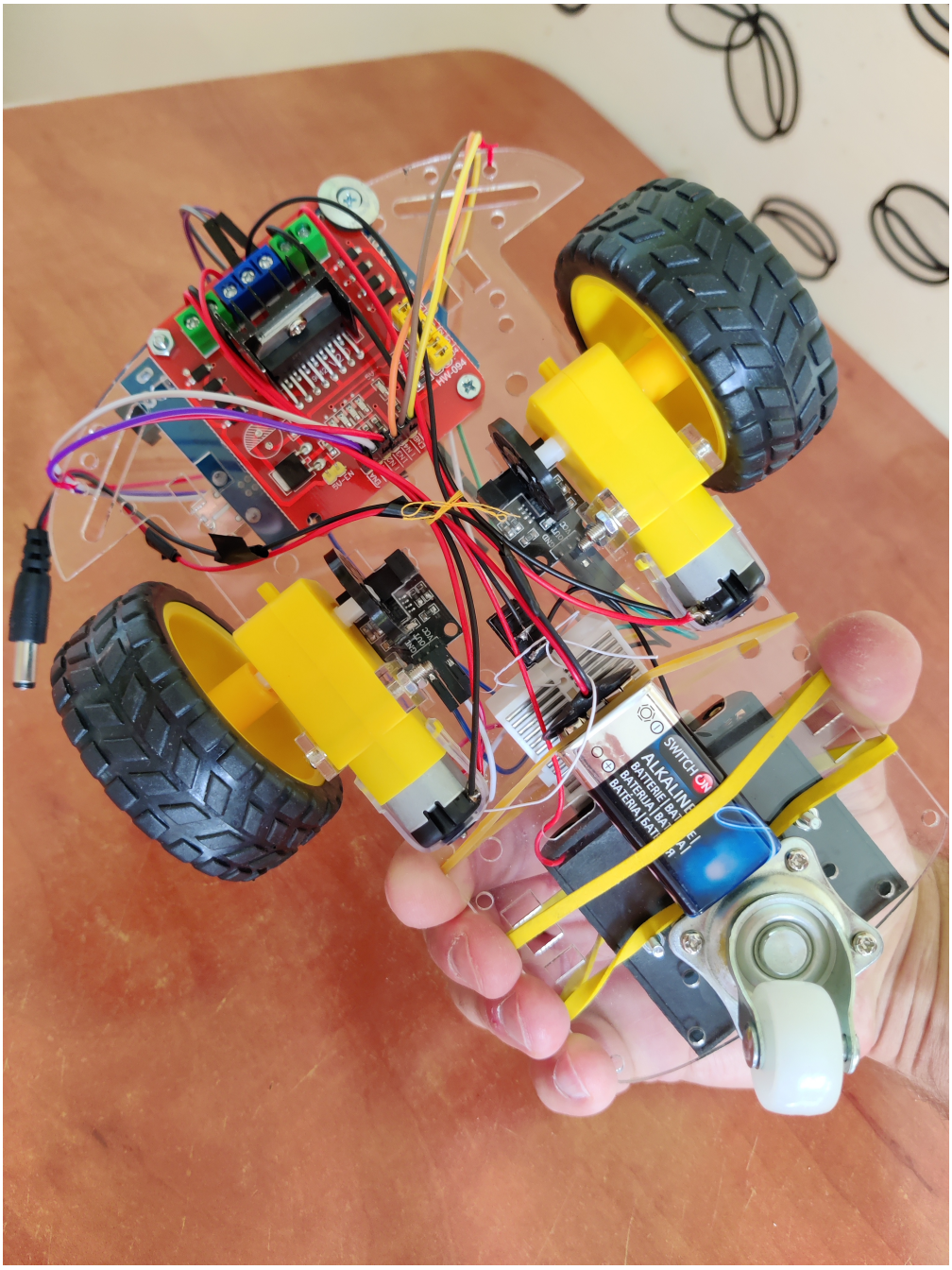


- 23 Mai 2021
  - Asamblare completă robot (hardware), conform schemelor bloc și imaginilor prezentate, fără senzorul de distanță HC-SR04 montat. Cuprinde plăcuța Arduino, driverul L298N H-Bridge, senzorii optocuplori H2010 cu microprocesor LM393, mini-breadboard-ul, bateriile 4 x 1,5V, bateria de 9V pentru Arduino, motoarele DC și roțile șasiului. În plus față de schema bloc, am implementat și un întrerupător pentru alimentarea motoarelor prin driverul L298N de la cele 4 baterii de 1,5V. Se vede în pozele următoare, switch-ul 0/1 dintre senzori, langa breadboard:




ANALOG IN





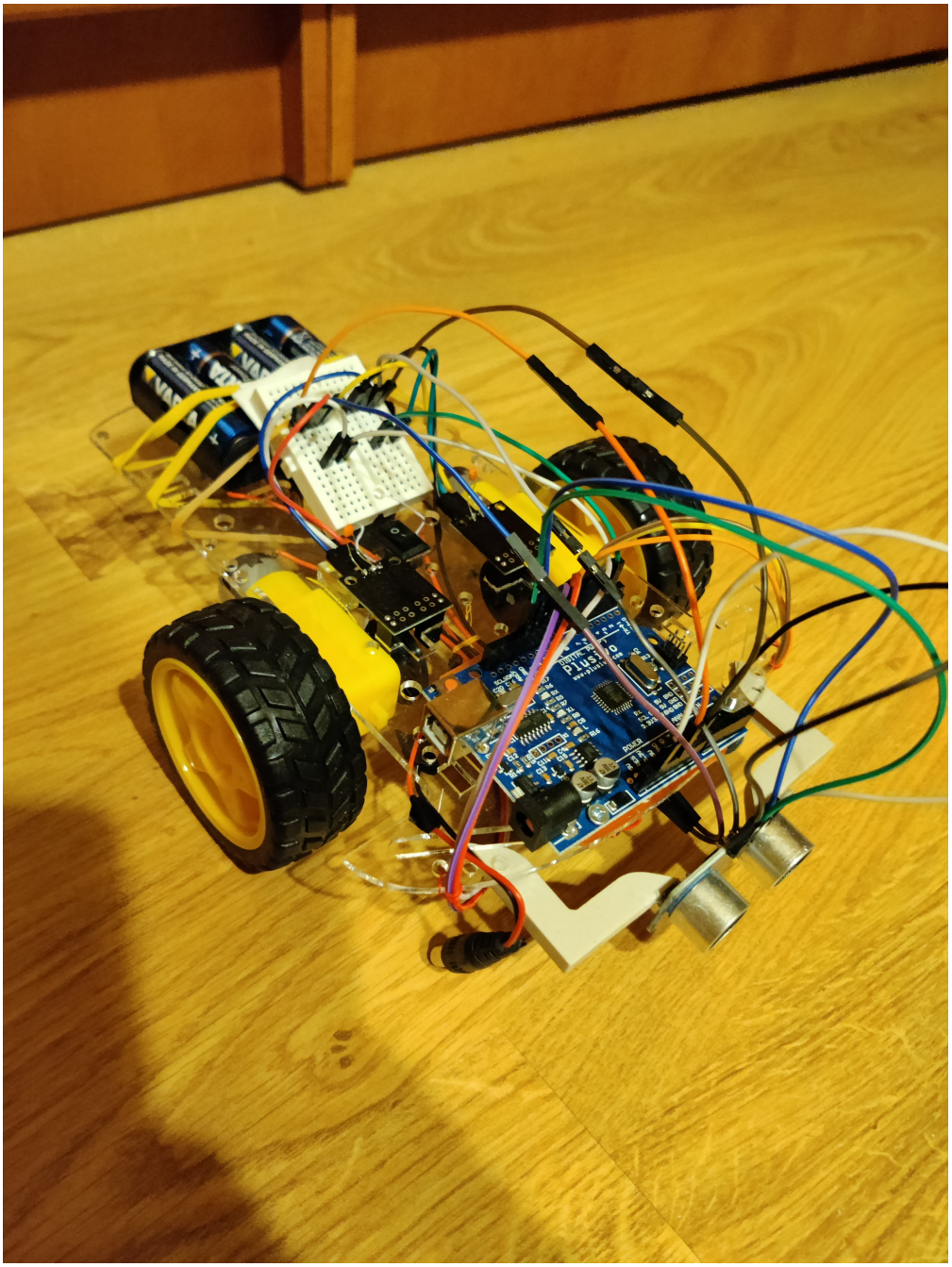
- Videoclip cu o rutină scurtă de testare a driver-ului L298N este la link-ul [7], robotul merge înainte la o viteză de 75% pentru 4 secunde. Codul sursa pentru aceasta rutină este:

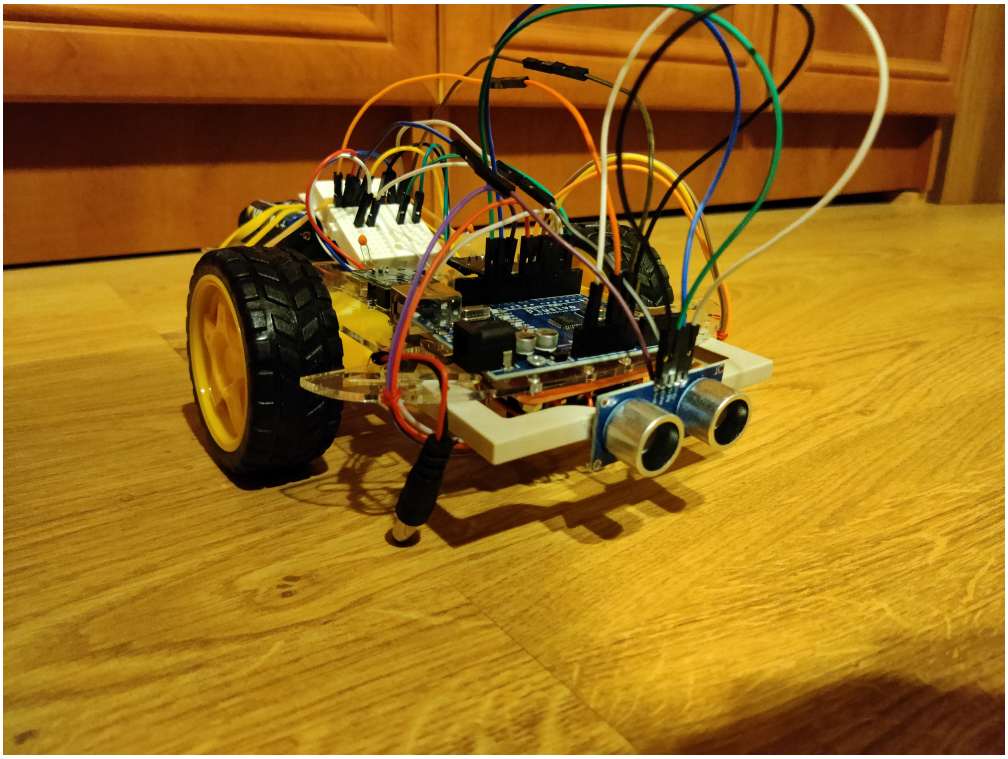
```
// Pins for Motor A
int en_A = 10;
int in_1 = 9;
int in_2 = 8;

// Pins Motor B

int en_B = 5;
int in_3 = 7;
int in_4 = 6;

void demoOne()
{
```

```
  // Turn on Motor A FORWARD
  digitalWrite(in_1, HIGH);
  digitalWrite(in_2, LOW);

  // Turn on Motor B FORWARD
  digitalWrite(in_3, HIGH);
  digitalWrite(in_4, LOW);

  delay(500);

  // Set Speed for Motor A at 200/255
  analogWrite(en_A, 200);

  // Set Speed for Motor B at 200/255
  analogWrite(en_B, 200);

  // Wait 4 seconds
  delay(4000);

  // Turn off both Motors A and B
  digitalWrite(in_1, LOW);
  digitalWrite(in_2, LOW);
  digitalWrite(in_3, LOW);
  digitalWrite(in_4, LOW);
}

void setup() {
  // put your setup code here, to run once:

  pinMode(en_A, OUTPUT);
  pinMode(en_B, OUTPUT);
  pinMode(in_1, OUTPUT);
  pinMode(in_2, OUTPUT);
  pinMode(in_3, OUTPUT);
  pinMode(in_4, OUTPUT);

  delay(2000);

  demoOne();
  delay(1000);
}

void loop() {
  // put your main code here, to run repeatedly
}
```

- 3 Iunie 2021
  - Am montat pe șasiu, în partea din față a robotului, senzorul de distanță HC-SR04 și l-am configurat software să măsoare distanța la fiecare 0.3 secunde, iar dacă aceasta este mai mică decât un prag stabilit (18 centimetri, de exemplu), să detecteze și să semnaleze acest lucru,

pentru ca să se ruleze rutina de evitare a obstacolului. (În cazul demo-ului meu, am ales ceva simplu: la fiecare detecție a unui obstacol, robotul să se întoarcă spre dreapta în jurul axei sale, cu 90 de grade, sau mai mult, de multe ori chiar și la 180 de grade (am încercat cu mai mulți pași ai trecerii fantelor encodoare prin dreptul senzorilor, 8, 10, și 15 (o rotiță encodoare are 20 de fante))). Mai jos atașez 2 poze cu robotul în varianta finală, cu senzorul de distanță atașat.





- La link-ul [8] este demo-ul, format dintr-o compilație de mai multe secvențe de rulare a robotului, în care acesta detectează și evită (în cele mai multe cazuri) obstacolele de pe 'traseu'.

Concluzii

Proiectul încă poate fi îmbunătățit din punct de vedere software, putând fi configurat să ruleze orice traseu și să reacționeze la obstacole în moduri diferite. Marea problemă în menținerea direcției robotului o reprezintă roata directoare de pe rulmentul din spatele șasiului (cum se poate observa în demo-ul de la link-ul [8]), care nu stă mereu dreaptă și astfel traiectoria nu este dreaptă atunci când ar trebui să fie în mod normal (deoarece motoarele au aceeași turație). Mai mult, senzorul nu detectează foarte precis dacă obstacolul nu este perpendicular pe direcția de emitere a acestuia, deoarece undele nu revin pe aceeași direcție înapoi către senzor, iar asta poate duce la opriri prea târzii, după cum se poate observa în unele situații din demo-ul de la link-ul [8].

Download

Arhiva cu codul folosit la filmarea videoclipurilor componente din compilația din demo-ul de la linkul [8]: ionita_dragos_332cb_demo_robot_v1.0.zip

https://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/obstacleavoidingcar?do=export_pdf

Bibliografie/Resurse

[1] https://dbot.ws/rbtcar

[2] https://dronebotworkshop.com

[3] http://androminarobot-english.blogspot.com/2017/03/encoder-and-arduinotutorial-about-ir.html

[4] https://youtu.be/5BZTMBUdEPs

[5] https://youtu.be/IS9QvN5y_8c

[6] https://youtu.be/H9Kglekphdg

[7] https://youtu.be/CIiOOT6_2Y0

[8] https://www.youtube.com/watch?v=EcM2bb0ISI8