

**Robot 4WD autonom ocolire obstacole cu senzor cu ultrasunete**

**Studente : Jula Mădălina-Veronica și Lungu Andrada-Cristiana**

**Grupa: 30233**

**Îndrumator: Răzvan Itu**

**Data: 11 Ianuarie 2021**

**Enunțul problemei**

Acest proiect pune problema miscării unui robot care evita obstacolele implementat pe placa Arduino UNO, cu ajutorul shield-ului Motor Control L293D sau a punții H.

În implementarea proiectului am folosit un kit special pentru mașină. Acesta conține:

Placă de dezvoltare UNO R3   
Șasiu robot 4WD(mașina).  
Roți 4 bucăţi.  
Motor 5v reductor robot (DC Gear Motor) 4 bucăţi.  
Organe de asamblare (șuruburi, piuliţe, plăcuţe)  
Modul driver motoare L293D Compatibil Arduino  
Suport baterii Compatibil Arduino  
Senzor cu ultrasunete HC-SR04 Compatibil Arduino  
Cabluri Dupont (mama-tata)  
Cabluri Dupont

Am adăugat în plus un servomotor și bineînteles sursele de alimentare.

Mașinile au fost asamblate de la 0. A fost nevoie de câteva material auxiliare pentru a cositori firele motoarelor la modulul de interfață și pentru a stabiliza servomotorul pe șasiul mașinii.

**Soluția**

Pentru a rezolva acest proiect a trebuit să ne documentat atât despre motoarele DC, servomotorul și senzorul de ultrasunete, cât și despre posibila funcționare a mașinii și stările acesteia.

Am ales în a scrie un cod simplu și destul de ușor de înțeles. Ce a fost mai dificil e că una dintre noi a primit un shield, iar cealalta o punte H, pentru a controla motoarele și sursa de alimentare, așa că vor fi puțin distincte codurile de la cele 2 mașinuțe.

În cod am inițializat motoarele, distantele necesare, pinii senzorului (Trigger si Echo). În functia de setup am substras distanța până la următorul obstacol ( sau pana unde a putut sa calculeze senzorul) și în funcția de loop am întrebat printr-un if daca această distanță este mai mică decât 20. Dacă propoziția este adevărată atunci mașina se va opri din mers, se va duce înapoi și va verifica în stânga și în dreapta dacă există un obiect. Dacă distanța din stânga este mai mica de 20, mașina o va lua în dreapta, dacă distanța din dreapta e mai mica, ea va merge în stânga. Dacă mașina nu are vreun obstacol, aceasta va merge în continuare în față.

#include <AFMotor.h>

#include <NewPing.h>

#include <Servo.h>

#define TRIG\_PIN A0

#define ECHO\_PIN A1

#define MAX\_DISTANCE 200

#define MAX\_SPEED 190 // sets speed of DC motors

#define MAX\_SPEED\_OFFSET 20

NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE);

AF\_DCMotor motor1(1, MOTOR12\_1KHZ);

AF\_DCMotor motor2(2, MOTOR12\_1KHZ);

AF\_DCMotor motor3(3, MOTOR34\_1KHZ);

AF\_DCMotor motor4(4, MOTOR34\_1KHZ);

Servo myservo;

boolean goesForward;

int distance = 200;

int speedSet = 0;

void setup() {

myservo.attach(10);

myservo.write(115);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop() {

int distanceR = 0;

int distanceL = 0;

delay(40);

if(distance<=20)

{

moveStop();

delay(50);

moveBackward();

delay(500);

moveStop();

delay(50);

distanceR = lookRight();

delay(200);

distanceL = lookLeft();

delay(200);

if(distanceL<=20)

{

turnRight();

moveForward();

}else

if (distanceR<=20)

{

turnLeft();

moveForward();

}

}else

{

moveForward();

}

distance = readPing();

}

int lookRight()

{

myservo.write(50);

delay(500);

int distance = readPing();

delay(100);

myservo.write(115);

return distance;

}

int lookLeft()

{

myservo.write(170);

delay(500);

int distance = readPing();

delay(100);

myservo.write(115);

return distance;

delay(100);

}

int readPing() {

delay(70);

int cm = sonar.ping\_cm();

if(cm==0)

{

cm = 250;

}

return cm;

}

void moveStop() {

motor1.run(RELEASE);

motor2.run(RELEASE);

motor3.run(RELEASE);

motor4.run(RELEASE);

}

void moveForward() {

goesForward=true;

motor1.run(FORWARD);

motor2.run(FORWARD);

motor3.run(FORWARD);

motor4.run(FORWARD);

for (speedSet = 0; speedSet < MAX\_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly

{

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(5);

}

}

void moveBackward() {

goesForward=false;

motor1.run(BACKWARD);

motor2.run(BACKWARD);

motor3.run(BACKWARD);

motor4.run(BACKWARD);

for (speedSet = 0; speedSet < MAX\_SPEED; speedSet +=2) // slowly bring the speed up to avoid loading down the batteries too quickly

{

motor1.setSpeed(speedSet);

motor2.setSpeed(speedSet);

motor3.setSpeed(speedSet);

motor4.setSpeed(speedSet);

delay(5);

}

}

void turnRight() {

motor1.run(FORWARD);

motor1.setSpeed(speedSet);

motor2.run(FORWARD);

motor2.setSpeed(speedSet);

motor3.run(RELEASE);

motor4.run(RELEASE);

}

void turnLeft() {

motor1.run(RELEASE);

motor2.run(RELEASE);

motor3.run(FORWARD);

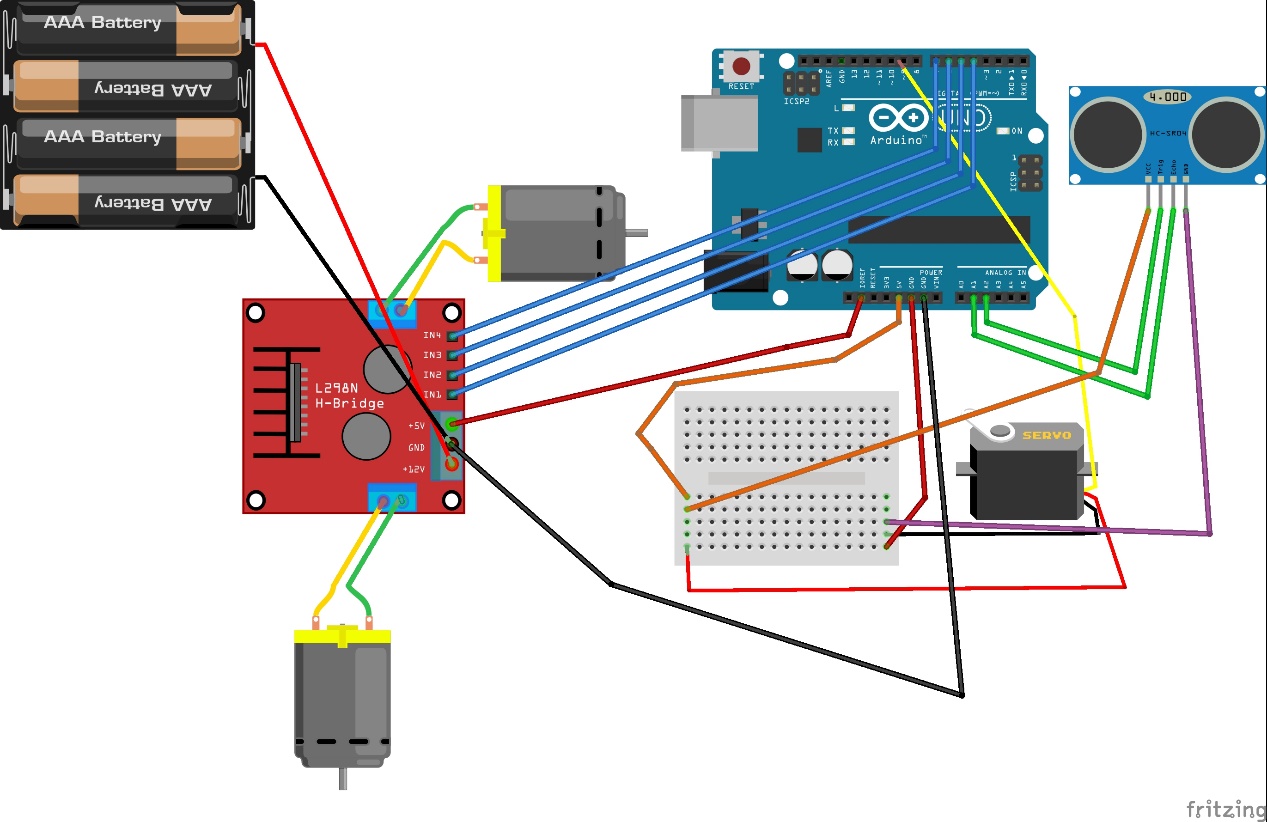
motor3.setSpeed(speedSet);

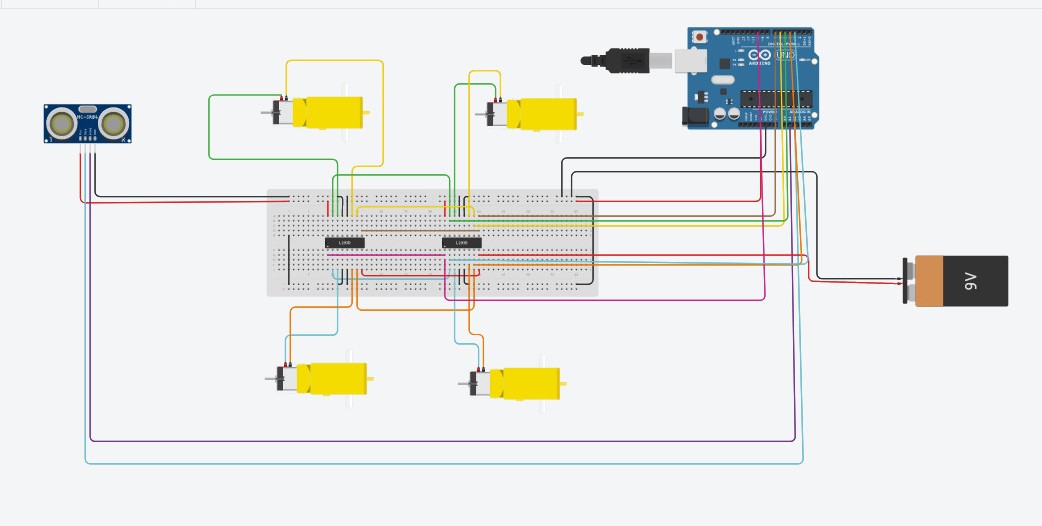
motor4.run(FORWARD);

motor4.setSpeed(speedSet);

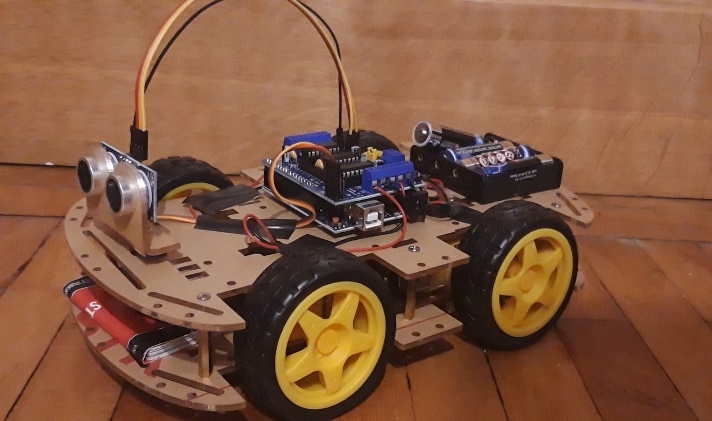
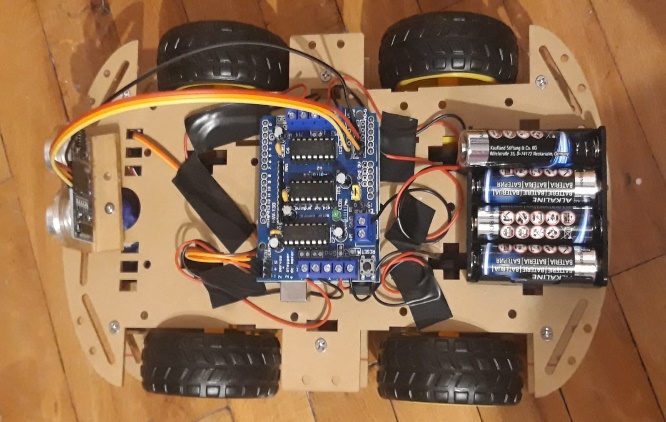
}

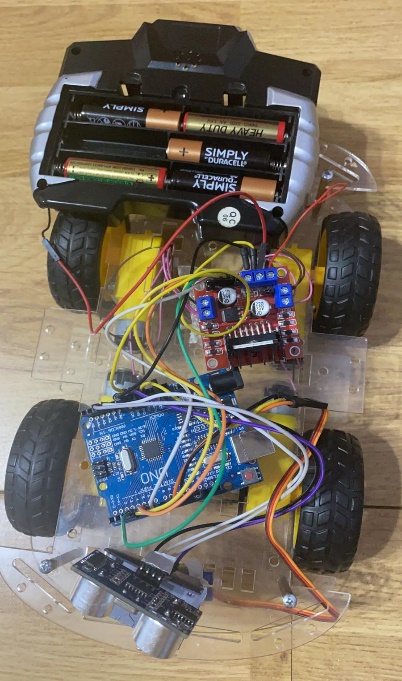
**Diagrama Circuitului (schema)**





**Poze cu proiectul**



****

**Bibliografie**

1. Lucrările de laborator: <https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf>
2. <https://roboromania.ro>
3. <https://create.arduino.cc/projecthub>
4. <https://www.youtube.com/watch?v=1n_KjpMfVT0&t=31s>
5. <https://github.com>