

Documentație – Proiect vehicul autonom

1. Echipa

Nume	Rol	Responsabilități
Ursu Tudor-Nicolae	Inginer hardware	Schema Tinkercad, alimentare, breadboard
Corbeanu Tudor-Nicolae	Programator	Dezvoltare cod Arduino, integrare senzori
Iustin Onosă	Designer mecanic	Construcție șasiu, montaj senzori

Achiziția pieselor a fost realizată de toți membri echipei, în funcție de necesități.

2. Prezentare generală

Proiectul constă în realizarea unui vehicul autonom, și în implementarea anumitor sisteme de siguranță de bază prezente la momentul actual pe toate autovehiculelor noi precum asistentul de coliziune frontală și asistentul de menținere a benzii. Soluția folosește componente ușor accesibile și relativ simple: Arduino UNO, driver de motoare L298N, doi senzori IR TCRT5000 și un senzor ultrasonic HC-SR04.

3. Motivație

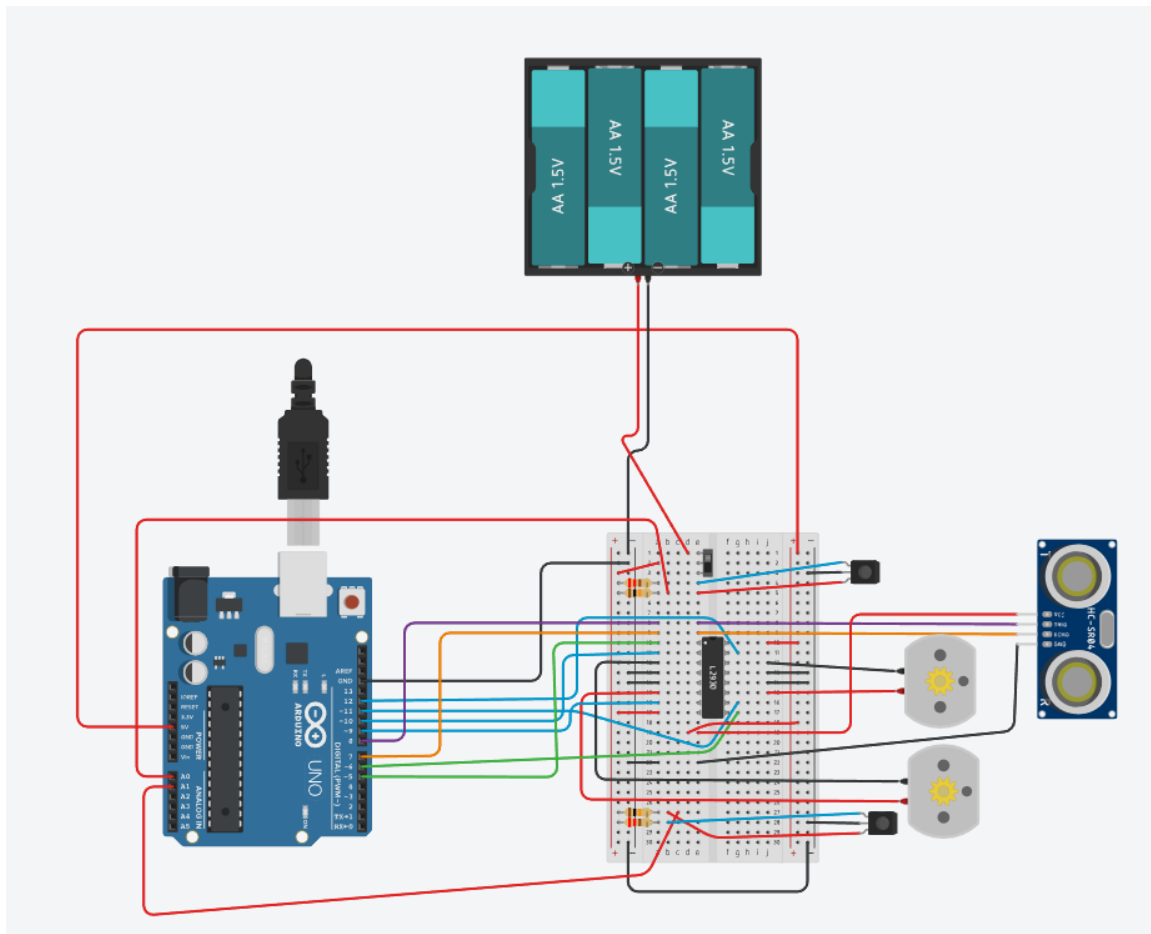
Am ales să realizăm un robot autonom folosind Arduino deoarece acest tip de proiect ne permite să aplicăm concret cunoștințele teoretice acumulate în domenii precum programarea, electronica și automatizarea. Robotul este o provocare practică ce implică lucrul în echipă, gândirea logică și capacitatea de a găsi soluții eficiente la probleme reale. Prin acest proiect ne dorim să dezvoltăm un sistem capabil să se deplaseze și să ia decizii în mod autonom, simulând comportamente utile în contexte precum evitarea obstacolelor sau navigarea într-un mediu necunoscut.

4. Arhitectura sistemului

Sistemul este împărțit în patru subsisteme principale:

- **Procesare & Control** – placa Arduino UNO, responsabilă cu citirea senzorilor și generarea semnalelor PWM pentru motoare
Tracțiune – două motoare DC 6 V cu reductor montate pe roțile din față
- **Senzori** – ansamblu compus din 2×TCRT5000 (deteecție contrast linie/asfalt) și HC-SR04 (măsurare distanță obstacol)
- **Alimentare** – 6×AA Ni-MH 2500 mAh ce oferă ~ 9 V

Schema realizată în Tinkercad:



5. Componente hardware

Cantitate	Piesa	Link
1	Arduino uno	https://ardushop.ro/ro/electronica/2271-placa-de-dezvoltare-uno-compatibila-arduino-dip-6427854022172.html
1	Breadboard	https://ardushop.ro/ro/electronica/84-breadboard-400-6427854020949.html
1	Jumper Wires	https://ardushop.ro/ro/electronica/83-65-x-fire-jumper-6427854023063.html
1	H-bridge L293D	https://www.emag.ro/circuit-integrat-l293d-ai823-s757/pd/DLBS4TMBM/?ref=history-shopping_422697024_112784_1
2	DC Motor cu reductor	https://ardushop.ro/ro/electronica/752-motor-dc-3v-6v-cu-reductor-1-48-6427854009609.html
6	Baterie AA	-
1	Suport Baterii	-
1	Intrerupator	https://ardushop.ro/ro/butoane--switch-uri/803-slider-switch-2-pozitii-6427854010391.html
4	Rezistor 220/10k	https://ardushop.ro/ro/componente-discrete/465-798-rezistor-1-4w-1-buc-alege-valoarea.html#/291-valoare_rezistenta-10r
2	Senzor IR	https://ardushop.ro/ro/electronica/1711-modul-senzor-ir-infrarosu-evita-obstacole-6427854025777.html
1	Senzor ultrasonic	https://www.emag.ro/senzor-ultrasonic-hc-sr04-3/pd/D3NL48YBM/?ref=history-shopping_422697024_157633_1
4	Roata	https://www.robofun.ro/roti-si-senile/roata-mg.html

6. Senzori

6.1. Senzori IR TCRT5000

Modulul emite lumină IR care se reflectă diferit pe culori deschise versus închise, fototranzistorul generând un curent proporțional cu cantitatea de lumină reflectată. În codul nostru, datele citite sunt LOW sau HIGH, adică 0 sau 1, LOW reprezentând culoarea neagră, iar HIGH culoarea albă.

6.2. Senzor ultrasonic HC-SR04

Modulul trimite un pachet de unde sonore la 40 kHz. Timpul dintre emitere și recepție este măsurat cu funcția `pulseIn()`, apoi convertit în distanță folosind viteza sunetului 343 m/s. Formula implementată în cod: `cm = durata / 58` (durata fiind în μ s). Pragul de oprire este setat la 30 cm pentru a asigura o distanță optimă de frânare.

7. Algoritm utilizat

Platforma utilizată pentru implementarea software-ului este Arduino IDE.

Limbajul utilizat: C++

```
int inainteStanga = 11;
int inapoiStanga = 12;
int inainteDreapta = 9;
int inapoiDreapta = 10;
int stangaViteza = 6;
int dreaptaViteza = 5;

const int triggerPin = 8;
const int echoPin = 7;
long duration;
long cm;

const int distantaPrag = 50;

int senzor1;
int senzor2;

const int senzor1Pin = A1;
const int senzor2Pin = A0;

void setup()
{
    pinMode(inainteStanga, OUTPUT);
    pinMode(inapoiStanga, OUTPUT);
    pinMode(inainteDreapta, OUTPUT);
    pinMode(inapoiDreapta, OUTPUT);
    pinMode(stangaViteza, OUTPUT);
    pinMode(dreaptaViteza, OUTPUT);

    pinMode(triggerPin, OUTPUT);
    pinMode(echoPin, INPUT);
}
```

```

pinMode(senzor1Pin, INPUT);
pinMode(senzor2Pin, INPUT);
}

void loop()
{
    //senzor ultrasonic
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    cm = duration/58;

    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    if(cm < distantaPrag)
        opreste(1000);
    else
        mergiInainte(10, 255, 255);

    //senzor IR
    senzor1 = digitalRead(senzor1Pin);
    senzor2 = digitalRead(senzor2Pin);

    if((senzor1 == LOW) && (senzor2 == LOW))
        mergiInainte(10, 255, 255);
    else if((senzor1 == HIGH) && (senzor2 == LOW))
        mergiDreapta(10, 255, 255);
    else if((senzor1 == LOW) && (senzor2 == HIGH))
        mergiStanga(10, 255, 255);
    else
        opreste(10);
}

void mergiInainte(int t, int vSt, int vDr)
{
    analogWrite(stangaViteza, vSt);

```

```

    analogWrite(dreaptaViteza, vDr);
    digitalWrite(inainteStanga, HIGH);
    digitalWrite(inapoiStanga, LOW);
    digitalWrite(inainteDreapta, LOW);
    digitalWrite(inapoiDreapta, HIGH);
    delay(t);
}

void mergiInapoi(int t, int vSt, int vDr)
{
    analogWrite(stangaViteza, vSt);
    analogWrite(dreaptaViteza, vDr);
    digitalWrite(inainteStanga, LOW);
    digitalWrite(inapoiStanga, HIGH);
    digitalWrite(inainteDreapta, HIGH);
    digitalWrite(inapoiDreapta, LOW);
    delay(t);
}

void mergiStanga(int t, int vSt, int vDr)
{
    analogWrite(stangaViteza, vSt);
    analogWrite(dreaptaViteza, vDr);
    digitalWrite(inainteStanga, LOW);
    digitalWrite(inapoiStanga, HIGH);
    digitalWrite(inainteDreapta, HIGH);
    digitalWrite(inapoiDreapta, LOW);
    delay(t);
}

void mergiDreapta(int t, int vSt, int vDr)
{
    analogWrite(stangaViteza, vSt);
    analogWrite(dreaptaViteza, vDr);
    digitalWrite(inainteStanga, HIGH);
    digitalWrite(inapoiStanga, LOW);
    digitalWrite(inainteDreapta, HIGH);
    digitalWrite(inapoiDreapta, LOW);
    delay(t);
}

void opreste(int t)
{

```

```
digitalWrite(inainteStanga, LOW);  
digitalWrite(inapoiStanga, LOW);  
digitalWrite(inainteDreapta, LOW);  
digitalWrite(inapoiDreapta, LOW);  
delay(t);  
}
```

8. Design mecanic

Șasiul robotului este construit integral din piese LEGO, oferind flexibilitate și ușurință în modificare sau construcție. Structura permite fixarea stabilă a senzorilor și componentelor electronice, fără a fi nevoie de adezivi sau imprimare 3D. Senzorii IR sunt montați frontal și complet lateral, în dreptul roților, la o distanță optimă față de sol pentru a asigura detecția eficientă a liniilor, iar senzorul HCSR04 este poziționat în partea din față a vehiculului pentru a detecta obstacole frontale. Bateria este amplasată central, între roți, pentru a menține un centru de greutate jos și pentru a oferi stabilitate.

9. Testare

Metodologie:

- Test static – citire senzor IR pe alb/negru.
- Test obstacol – plasare obiect la 40 cm și 30 cm, măsurare timp de oprire (<150 ms).
- Test traseu – linie albă vopsea pe asfalt de culoare neagră.

10. Îmbunătățiri viitoare

Pentru dezvoltarea ulterioară a proiectului, echipa noastră a identificat mai multe optimizări care ar putea crește eficiența și performanța vehiculului:

1. Integrarea unui sistem de logare Bluetooth

Pentru a analiza comportamentul robotului în timpul deplasării, planificăm adăugarea unui modul de transmisie Bluetooth conectat la un logger care salvează datele senzorilor și deciziile de navigație în timp real. Înregistrarea va permite o analiză detaliată offline, facilitând optimizarea algoritmilor.

2. Trecerea la un șasiu imprimat 3D

Deși structura actuală LEGO ne-a oferit flexibilitate și rapiditate în prototipare, planificăm realizarea unui șasiu personalizat prin imprimare 3D. Acesta va permite optimizarea greutății, a rigidității și a amplasării senzorilor, oferind o platformă mai stabilă pentru versiunea finală a robotului.