



DOCUMENTAȚIE ROBOTICĂ

Profesor coordonator : Trăsnea Bogdan

Student : Coroiu Dana- Cornelia

Grupa : 4LF412



Robot ocolitor de obstacole

Profesor coordonator : Trăsnea Bogdan

Student : Coroiu Dana- Cornelia

Grupa : 4LF412



Cuprins

1. Introducere.....	pag.4
2. Componente(Aritectura).....	pag.4
3. Asamblare.....	pag.6
4. Schema electrica.....	pag.7
5. Cod.....	pag.8
6. Probleme intampinate.....	pag.12
7. Ansamblu final.....	pag.13
8. Bibliografie.....	pag.14

1. Introducere

Proiectul presupune asamblarea și programarea unui robot mobil care să evite obstacole cu ajutorul unui senzor cu ultrasunete .

2. Componente (Arhitectura)

Pentru asamblarea robotului am folosit un kit de sasiu cu 4 roți (fig. 2.1)



(fig. 2.1- Kit sasiu robot cu 4 roți)

-Placa dezvoltare compatibila Arduini UNO 328 (fig. 2.2)



(fig. 2.2- Placa dezvoltare compatibila Arduini UNO 328)

-Baterie 9V DURACELL (fig. 2.3)



(fig. 2.3 - Baterie 9V)

- L298N Driver cu punte H pentru a controla motoarele (fig.2.4)



(fig. 2.4- L298N Driver cu punte H)

-Senzor ultrasonic(fig. 2.5)



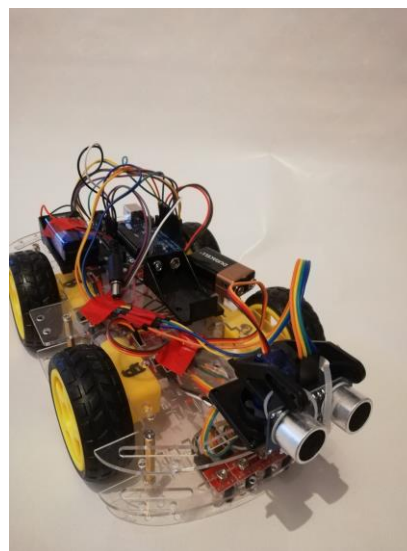
(fig. 2.5- senzor ultrasonic)

3. Asamblare

Am asamblat sasiul astfel:

- am dezlipit folia de protecție de pe placile acrilice ce compun carcasa;
- am fixat motoarele cu ajutorul encoderelor și al suruburilor cu piulita;
- am folosit distantieri pentru a prinde cele două plăci astfel încât să aibă loc motoarele și celelalte piese necesare funcționării robotului.[3]

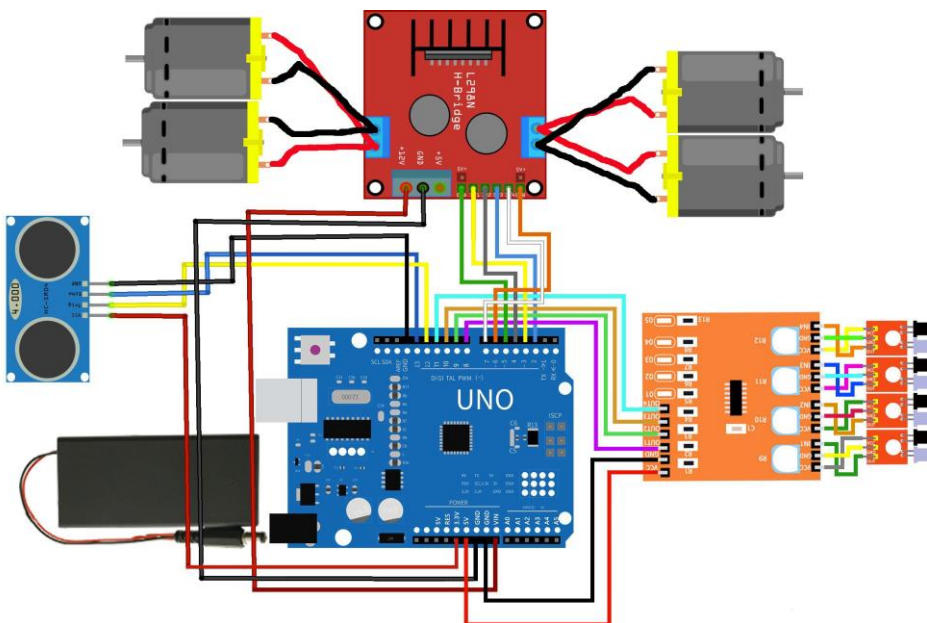
În interiorul sasiului se află driver-ul dublu cu punte H, care este conectat la motoare, iar deasupra se regăsește placa Arduino, suportul de baterie și senzorul ultrasonic.



3.Schema electrica

Am conectat firele conform schemei electrice din fig. 3.1 .

Obs. : Schema electrica corespunde si modulului de urmarire al liniei.[1]



(fig. 3.1- Schema electrica)

Obs. : Robotul porneste la introducerea bateriei de 9v.



4.Cod

Am descărcat biblioteca specială AFmotor.zip . [2]

```
#include <AFMotor.h>

//initializarea și definire pinilor

#define Trig 12

#define Echo 13

#define ENA 5

#define ENB 6

#define IN1 3

#define IN2 4

#define IN3 2

#define IN4 7

// variabila de distanță

float cm;

float temp;

void setup() {

    Serial.begin(9600);

    pinMode(Trig, OUTPUT);

    pinMode(Echo, INPUT);

    pinMode(IN1, OUTPUT);
```




```
pinMode(IN2, OUTPUT);  
pinMode(IN3, OUTPUT);  
pinMode(IN4, OUTPUT);  
  
pinMode(ENA, OUTPUT);  
pinMode(ENB, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(Trig, LOW);  
    delayMicroseconds(2);  
    digitalWrite(Trig, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(Trig, LOW);  
  
    temp = float(pulseIn(Echo, HIGH));  
    cm = (temp * 17 ) / 1000;  
    if (cm < 30 && cm > 10) //daca distanta este mai mica de 30 de cm ,dar mai mare de 10 cm  
    {  
        back();  
        delay(500);  
        Left();  
        delay(200);  
    }  
    if (cm >= 30) //daca distanta este mai mare sau egala cu 30 de cm atunci robotul se intoarce  
    {  
        forward();  
        delay(100);  
    }
```



```
}
```

```
if (cm < 10) // daca distanta este mai mica de 10 cm atunci robotul se opreste
```

```
{
```

```
    STOP();
```

```
}
```

```
Serial.print("Echo =");
```

```
Serial.print(temp);
```

```
Serial.print(" | | Distance = ");
```

```
Serial.print(cm);
```

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

```
delay(100);
```

```
}
```

```
// definirea directiilor de miscare
```

```
void forward() {
```

```
    analogWrite(ENA, 220);
```

```
    analogWrite(ENB, 220);
```

```
    digitalWrite(IN1, HIGH);
```

```
    digitalWrite(IN2, LOW);
```

```
    digitalWrite(IN3, LOW);
```

```
    digitalWrite(IN4, HIGH);
```

```
    Serial.println("Forward");
```

```
}
```

```
void back() {
```

```
    analogWrite(ENA, 220);
```



```
analogWrite(ENB, 220);  
digitalWrite(IN1, LOW);  
digitalWrite(IN2, HIGH);  
digitalWrite(IN3, HIGH);  
digitalWrite(IN4, LOW);  
Serial.println("Back");  
}
```

```
void Left() {  
    analogWrite(ENA, 220);  
    analogWrite(ENB, 220);  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    Serial.println("Left");  
}
```

```
void Right() {  
    analogWrite(ENA, 220);  
    analogWrite(ENB, 220);  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    Serial.println("Right");  
}
```

```
void STOP() {
    digitalWrite(ENA, LOW);
    digitalWrite(ENB, LOW);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    Serial.println("STOP");
}
```

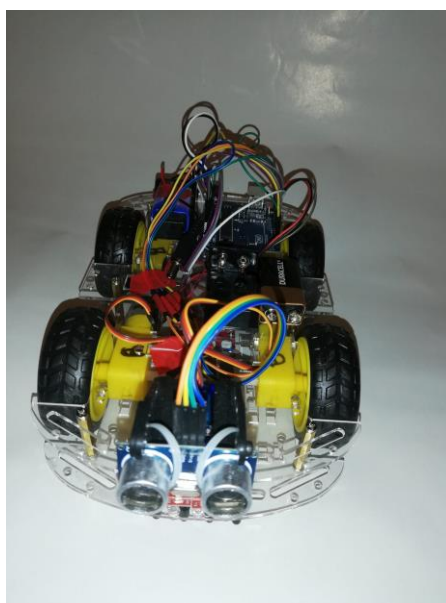
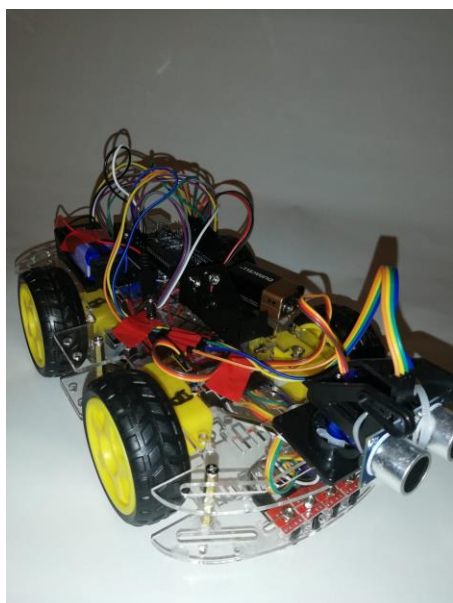
5. Probleme intampinate

Calitatea bateriei -> am atasat prima data 2 baterii de 9V(BattExtra), dar mai puțin calitative. Astfel, am înlocuit cele 2 baterii cu una (tot de 9V, Duracell), dar de o calitate mai bună.



5. Ansamblu final

În urma celor menționate mai sus, am reușit să realizez un robot ce evită obstacole cu ajutorul unei plăcuțe Arduino, sasiului, un driver cu punte H, un senzor ultrasonic și o baterie de 9V. Alte piese de care am avut nevoie au fost: fire de legătură mama-tata, suruburi, piulițe și suport de baterie de 9V. Rezultatul final este prezentat în imaginile de mai jos.





7. Bibliografie

1. [\(7884\) Arduino 4WD Obstacle Avoidance Car - YouTube](#)
2. [Obstacle Avoidance Mode - Arduino SA](#)
3. [Kit Smart Car 4WD, Bluetooth, IR, Ultrasunete \(sigmanortec.ro\)](#)