



# DOCUMENTAȚIE

## RC-Car

Profesor coordonator : Trăsnea Bogdan

Nume : Ivascu Andrei Iulian

Programul de studii : Automatica

Grupa : 4LF412

An : 2022-2023

### Cuprins

1.Introducere.....	2
2.Arhitectură.....	2
3.Realizarea părții HW .....	5
4.Realizarea părții SW .....	7
5.Ansamblu final .....	9
6.Anexe.....	10

# 1.Introducere

Ideea a pornit de la reconfecţionarea unei maşini de tip RC, pe care să o putem controla printr-un senzor Bluetooth, cu ajutorul telefonului.

Această idee pe care am pus-o în aplicare nu este tocmai simplă, deoarece câteva dintre funcţiile maşinii au fost regândite şi am avut nevoie de următoarele componente: şasiul maşinii, Modul Bluetooth HC-05, Modul L298N cu PUNTE H DUBLĂ, Motor Servo SG90 9g, Senzor ultrasonic HC-SR04, Breadboard, DC-motor 3-6 V, Arduino UNO, o baterie de 9V şi nişte linii de cod scrise în IDE-ul de Arduino.

## 2.Arhitectură

Pentru început, aveţi nevoie de şasiul maşinii (2.1).

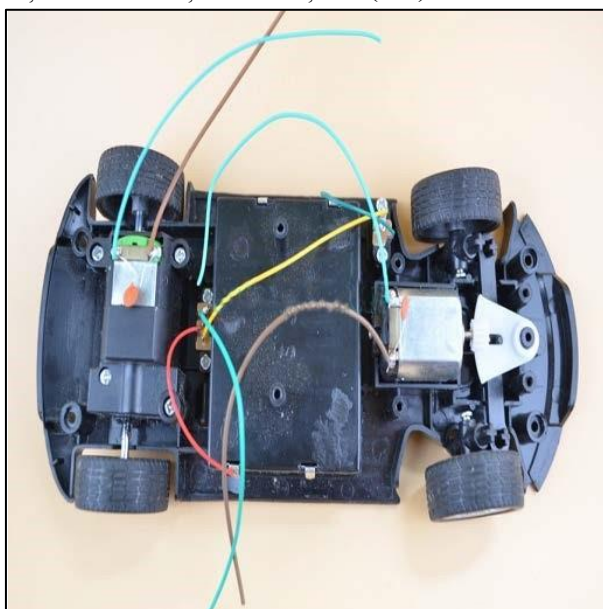


Fig.2. 1 Şasiu maşina RC

După care, aveţi nevoie de o plăcuţă Arduino (2.2), o baterie de 9V (2.3) , un Modul Bluetooth HC-05 (2.4), un Senzor Ultrasonic HC-SR04 (2.5), un Modul L298N cu PUNTE H DUBLĂ (2.6), Breadbord, Motor Servo SG90 9g şi un motor DC 3-6V (2.7).

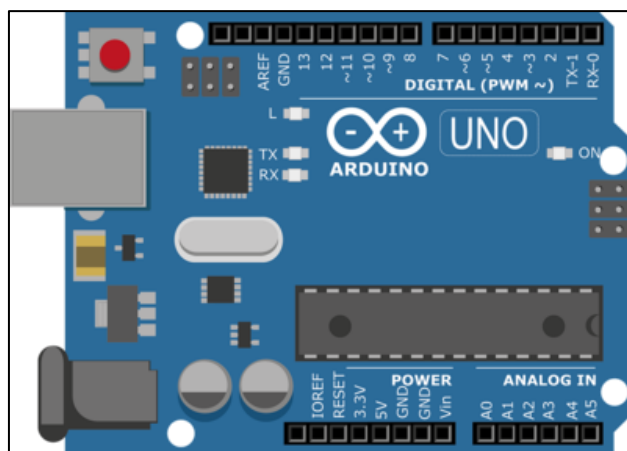


Fig.2. 2 Plăcuță Arduino



Fig.2. 3 Baterie 9V

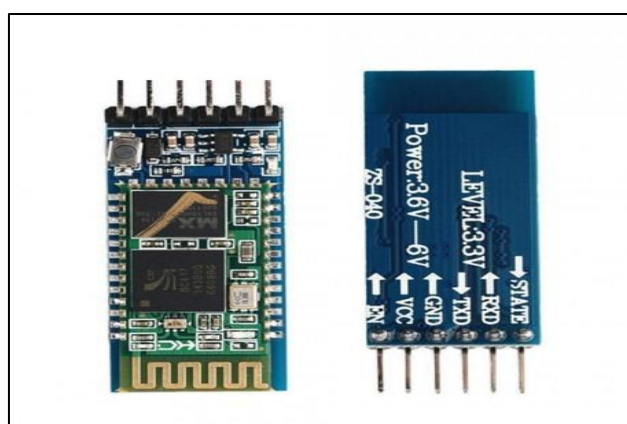


Fig.2. 4 Modul Bluetooth HC-05



Fig.2. 5 Senzor Ultrasonic HC-SR04

Pentru a controla motoarele este nevoie de un Modul de tip L298N în punte H, dublă (2.6).

Fig.2. 6 Modul L298N cu punte H dublă

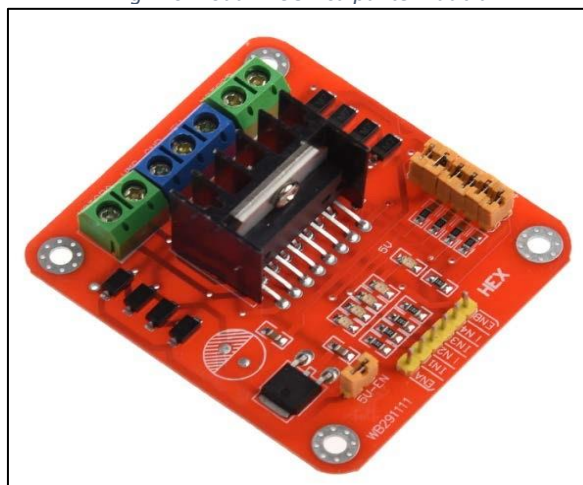


Fig.2. 7 DC-motor 3-6 V



### 3. Realizarea părţii HW

Ca prim pas, am avut nevoie de o maşină RC, căreia i-am scos caroseria. Apoi, am început să lucrăm pe şasiul maşinii, la care am regândit direcţia şi tracţiunea. Următorul pas pe care l-am făcut a fost montarea unui modul de tip L298N pentru a controla motorul. Ulterior, modulul a fost conectat la plăcuţa de Arduino.

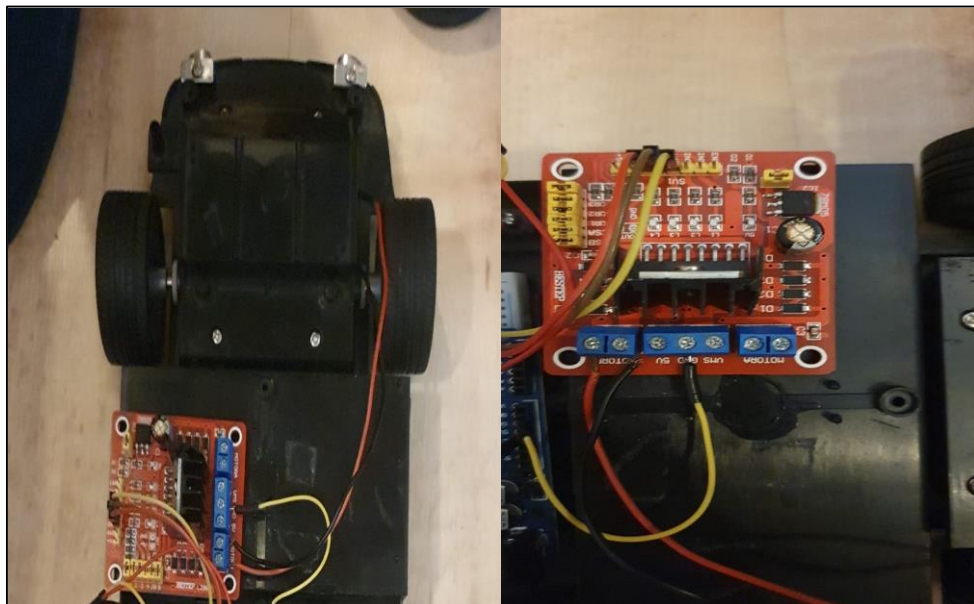


Fig3. 1

În figura (3.2) se pot vedea toate porturile folosite de pe plăcuţa Arduino.

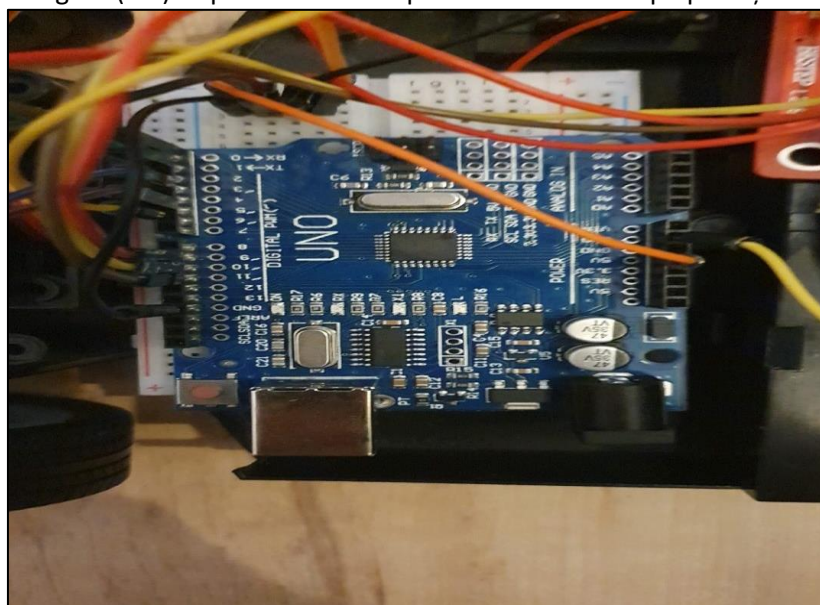


Fig3. 2

Pentru direcţia maşinii am montat un Motor Servo SG90 9g, care se află pe puntea faţă. De asemenea, pentru controlul maşinii am montat un modul Bluetooth HC-05 prin care putem da comenzi



maşinii. Ultimul pas pe care l-am făcut a fost montarea Senzorului Ultrasonic HC-SR04 pentru evitarea ciocnirilor.



Fig3. 3

Pinii IN3, IN4 şi ENB de pe modulul L298N trebuie conectaţi la pinul 10,11 şi 9 GND de pe plăcuţa de Arduino şi modulul 298N este alimentat de o baterie de 9V prin porturile VMS şi GND. GNDul de pe modulul L298N trebuie conectat la GND-ul de pe Arduino. Motorul este conectat la poarta B pe modulul L298N. Servo motorul SG90 9g care controlează direcţia este legat la pinul 6, alimentat cu 5V la GND pe plăcuţa Arduino. Modulul Bluetooth HC-05, RX trebuie conectat la pinul 0 (TX) de pe Arduino şi TX-ul la pinul 1 (RX) de pe Arduino, alimentat cu 5V şi legat la GND.

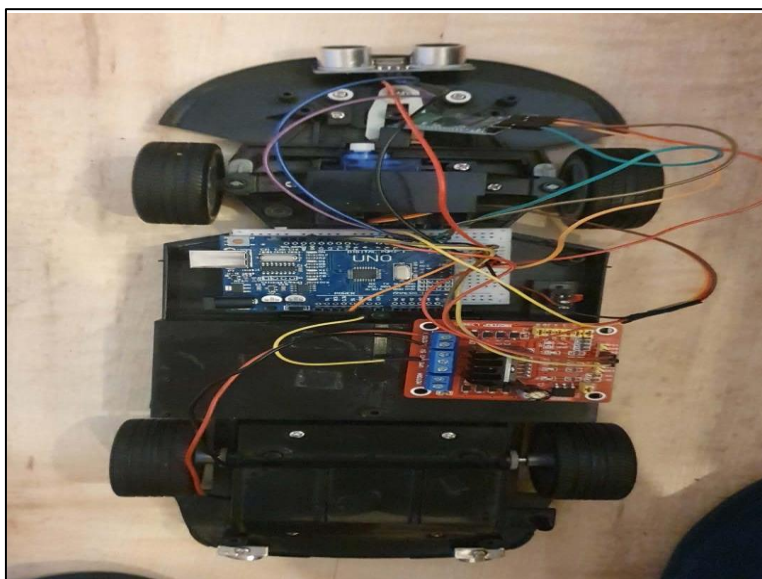


Fig3. 4

Trigger-ul Senzorului Ultrasonic HC-SR04 trebuie conectat la pinul 3 de pe Arduino, iar Echo-ul va fi conectat la pinul 5 de pe Arduino, alimentat cu 5V la GND pe plăcuţa de Arduino.

## 4.Realizarea părţii SW

În următoarele figuri se vor putea vedea fragmente din cod.

```
1
2  #include<SoftwareSerial.h>
3  #include<Servo.h>
4  Servo motor;
5
6
7  SoftwareSerial BlueTooth(0,1);
8  char BT_input;
9  int motorPin1=10;
10 int motorPin2=11;
11 int trigPin = 3;
12 int echoPin = 5;
13 int led = 4;
14
15
16 void setup()
17 {
18   BlueTooth.begin(9600);
19   Serial.begin(9600);
20   pinMode(motorPin1,OUTPUT);
21   pinMode(motorPin2,OUTPUT);
22   pinMode(9, OUTPUT);
23   pinMode(trigPin, OUTPUT);
24   pinMode(echoPin, INPUT);
25   pinMode(led, OUTPUT);
26   motor.attach(6);
27   motor.write(100);
28
29 }
30 void loop()
31 {
32   long duration, distance, verf=0; //Calculul distantei cu senzorul ultrasonic
33   digitalWrite(trigPin, LOW);
```

Fig. 4. 1

```

34 delayMicroseconds(20);
35 digitalWrite(trigPin,HIGH);
36 delayMicroseconds(100);
37 digitalWrite(trigPin, LOW);
38 duration = pulseIn(echoPin, HIGH);
39 distance = duration/29.387/2; //calculul distantei in cm
40 if(Bluetooth.available(>)0) //Testarea existentei conexiunii bluetooth
41 {
42     while(distance){ //Folosim while pentru ca recalcularea distantei sa fie facuta astfel incat
43         BT_input=Bluetooth.read(); //Daca nu este respectata conditia de distanta data de if-ul de mai jos tot procesul functiei
44         digitalWrite(trigPin, LOW); //void loop sa fie reluat
45         delayMicroseconds(20);
46         digitalWrite(trigPin,HIGH);
47         delayMicroseconds(100);
48         digitalWrite(trigPin, LOW);
49         duration = pulseIn(echoPin, HIGH);
50         distance = duration/29.387/2;
51         if(distance <= 50 && verf==1) //daca distanta inregistrata de senzor este mai mica de 50 cm
52         { //masina franeaza
53             digitalWrite(motorPin1,LOW);
54             digitalWrite(motorPin2,HIGH);
55             analogWrite(9, 255);
56             delay(200);
57             verf=0;
58             digitalWrite(motorPin1,LOW);
59             digitalWrite(motorPin2,LOW);
60             analogWrite(9, 255);
61         }
62     }
63     else if(BT_input=='f') //Comanda deplasare inainte
64     {
65         digitalWrite(motorPin1,HIGH);
66         digitalWrite(motorPin2,LOW);
67         analogWrite(9, 255);
68         verf=1;
69     }
70     else if(BT_input=='b') //Comanda deplasare inapoi
71     {
72         digitalWrite(motorPin1,LOW);
73         digitalWrite(motorPin2,HIGH);
74         analogWrite(9, 255);
75         verf=0;
76     }
77 }
78 else if(BT_input=='s') //Comanda franare
79 {
80     digitalWrite(motorPin1,LOW);
81     digitalWrite(motorPin2,HIGH);
82     analogWrite(9, 255);
83     delay(200);
84     verf=0;
85     digitalWrite(motorPin1,LOW);
86     digitalWrite(motorPin2,LOW);
87     analogWrite(9, 255);
88 }
89 else if(BT_input=='i') //Comanda directiei centru
90 {
91     motor.write(100);
92 }
93 else if(BT_input=='l') //Comanda directiei stanga
94 {
95     motor.write(145);
96 }
97 else if(BT_input=='r') //Comanda directiei dreapta
98 {
99     motor.write(55);
100 }

```

Fig. 4. 2



```
101     else if(BT_input=='n')
102     {
103         digitalWrite(led, HIGH); //aprinde faruri
104     }
105     else if(BT_input=='o')
106     {
107         digitalWrite(led, LOW); //stinge faruri
108     }
109 }
110 }
111
112
113
114
115 }
```

Fig. 4. 3

Împreună cu codul de pe Arduino, am folosit și aplicația de pe telefon, Arduino Bluetooth Controller pe care o puteți găsi doar pe PlayStore.

## 5. Ansamblu final

În imaginile de mai jos, se poate vedea asamblarea finală a proiectului.



Fig.5. 1



Fig.5. 2

## 6. Anexe

1. <https://overhelmed-arduino-coding.blogspot.com/2020/10/ultrasonic-sensor-ftdc-motor-in.html>
2. <https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor>