**Robot controlat prin Bluetooth care detectează obstacolele și le evită în mod automat**

**Studenti: Făgădar Ionela-Cătălina**

**Hădărean Andreea-Roxana**

**Grupa: 30236**

**Profesor îndrumător: Dănescu Radu**

**Data: 16.01.2020**

**Cuprins**

[**1.** **Obiective** 2](#_Toc30002481)

[**2.** **Introducere** 2](#_Toc30002482)

[**2.1** **Componente utilizate** 2](#_Toc30002483)

[**2.2** **Schema hardware** 5](#_Toc30002484)

[**3.** **Algoritm** 5](#_Toc30002485)

[**4.** **Rezultate** 11](#_Toc30002486)

[**5.** **Concluzii** 12](#_Toc30002487)

# **Obiective**

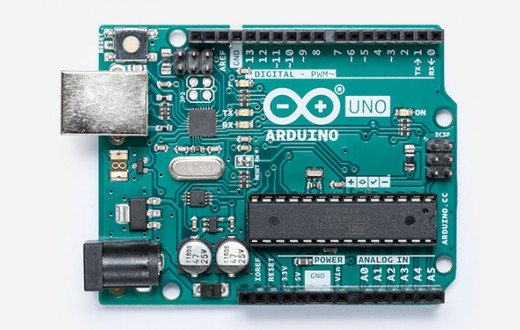
Principalul obiectiv al proiectului a fost programarea unui robot autonom, folosind cunostintele acumulate de-a lungul orelor de laborator. Acesta are capabilitatea de a se misca, de a detecta obiectele din mediul inconjurator cu ajutorul senzorului ultrasonic si de a le ocoli pentru a evita posibilele coliziuni, generand totodata un sunet de frecventa proportionala cu distanta la care se afla obiectele.

# **Introducere**

Tema proiectului consta in proiectarea si implementarea unui robot a carui actiuni pot fi controlate prin intermediul modulului Bluetooth, putand realiza miscari in 4 directii: N, S, E, V. De asemenea, daca intalneste obstacole in drum, acesta a fost proiectat sa isi analizeze mediul inconjurator, hotarand autonom ce directie sa urmeze pentru a ocoli eventualul obstacol. In momentul detectarii obstacolului, LED-ul de pe pinul 13 se aprinde, iar buzzer-ul conectat creeaza un sunet a carui frecventa este direct proportionala cu distanta la care se afla respectivul obstacol.

## **Componente utilizate**

Principala componenta utilizata in proiectarea si implementarea robotului este reprezentata de placa de dezvoltare Arduino UNO Rev3, placa care a fost programata pentru a realiza functionalitatile descrise mai sus.



Figură - Placa de dezvoltare Arduino UNO

Au fost utilizate module variate pentru a se obtine modul de functionare dorit, astfel:

* Modulul bluetooth 4.0 BLE CC2540 CC2541 – pentru conexiunea cu telefonul si receptia comenzilor de catre robot
* Senzor ultrasonic HC-SR04 – pentru detectia obiectelor
* Buzzer piezoelectric activ 3-24V HND-2312 – pentru a emite un sunet la aparitia unui obstacol



Figură - Modul Bluetooth Figură - Senzor ultrasonic Figură – Buzzer

Pentru miscarea robotului, s-au folosit 2 motoare DC impreuna cu o punte H, deoarece aceste motoare necesita o intensitate a curentului semnificativa. Viteza de rotatie a acestor motoare depinde de tensiunea de alimentare de la bornele acestuia, iar directia depinde de polaritatea acesteia. Aceste motoare actioneaza asupra a doua roti, ce ajuta la miscare, fiind prezenta si o a treia roata de suport.

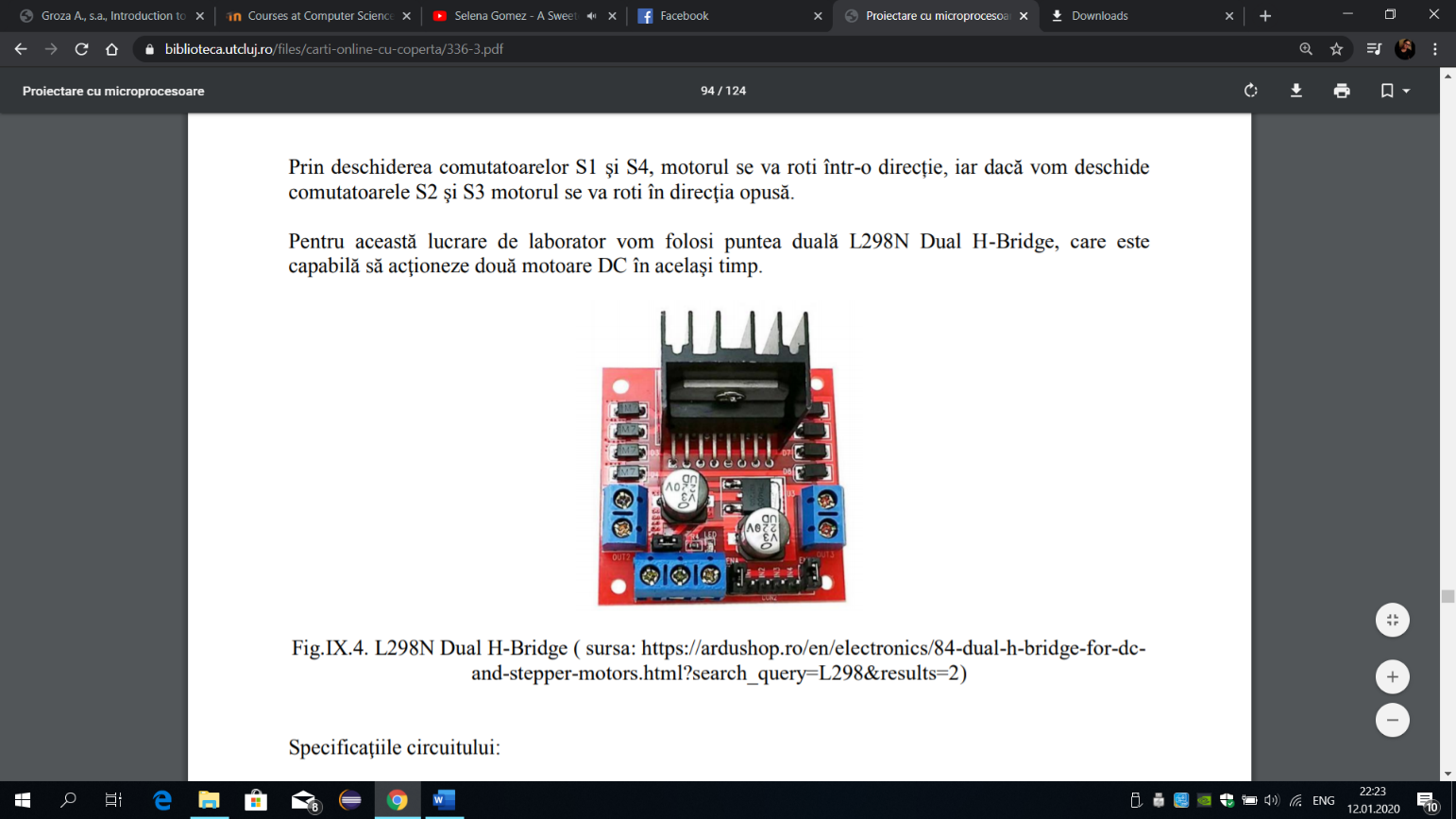
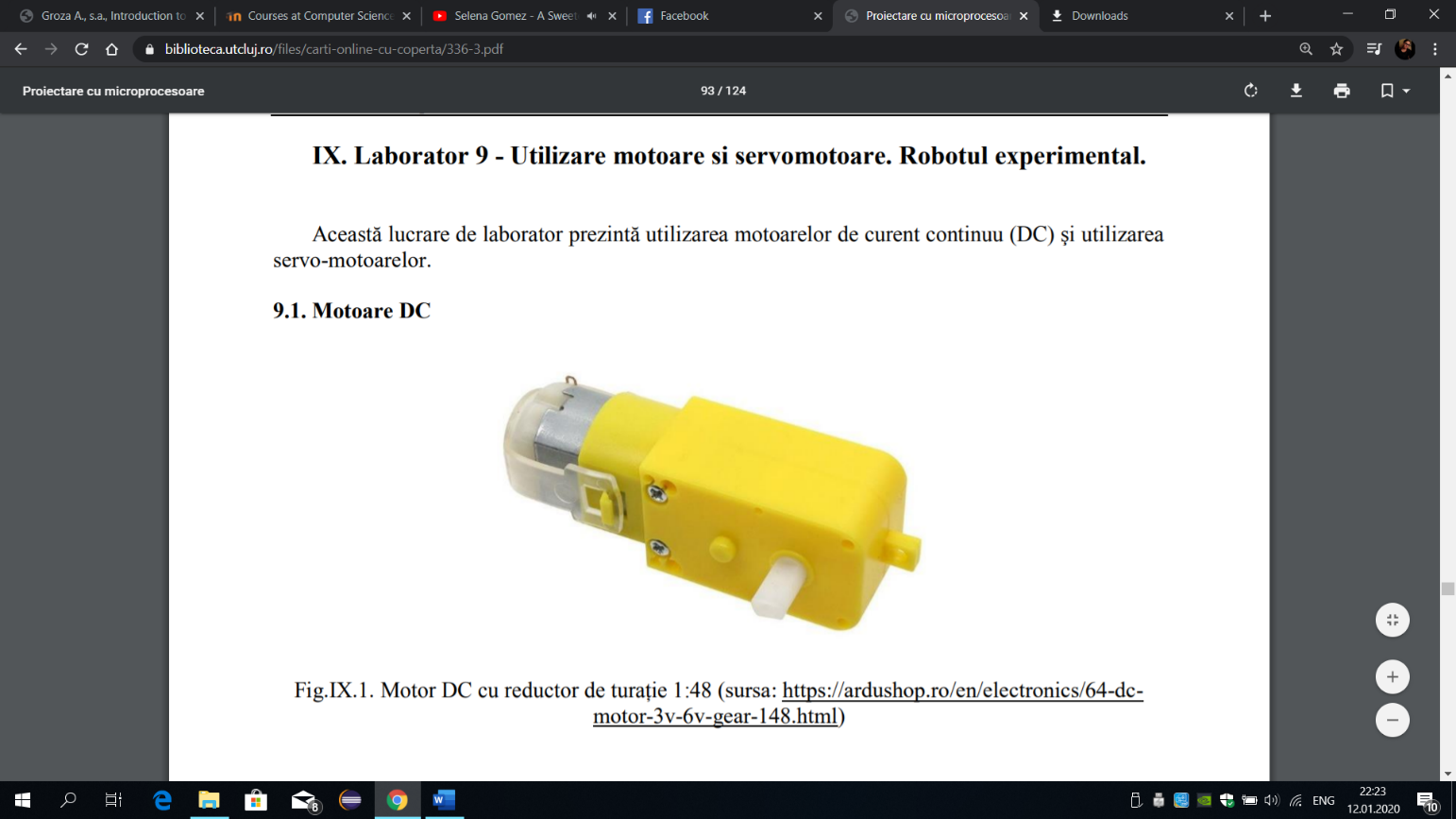


Figure 5 - Motor DC

Figure 6 - punte H

Fiecare motor are 3 pini de control. Primul motor este actionat de pinii EnA (“1”), In1 si In2, pe cand celalat de EnB (“1”), In3 si In4.

Table - Comenzii pentru motor

|  |  |  |
| --- | --- | --- |
| In1 (In3) | In2 (In4) | Efect |
| 0 | 0 | Motor oprit |
| 0 | 1 | Motor pornit – inainte |
| 1 | 0 | Motor pornit – inapoi |
| 1 | 1 | Motor oprit |

Pentru rotatia senzorului ultrasonic, s-a folosit un motor servo, deoarece acesta este un motor cu precizie ridicata. Motorul servo are 3 fire: Vcc (5V) care este de obicei de culoare rosu, GND, care are culoare neagra sau maro si un fir de comanda. De obicei, acest motor servo nu efectueaza o rotatie completa, ci doar o rotatie cu un unghi. Pentru actionarea lui, s-a folosit biblioteca Servo.

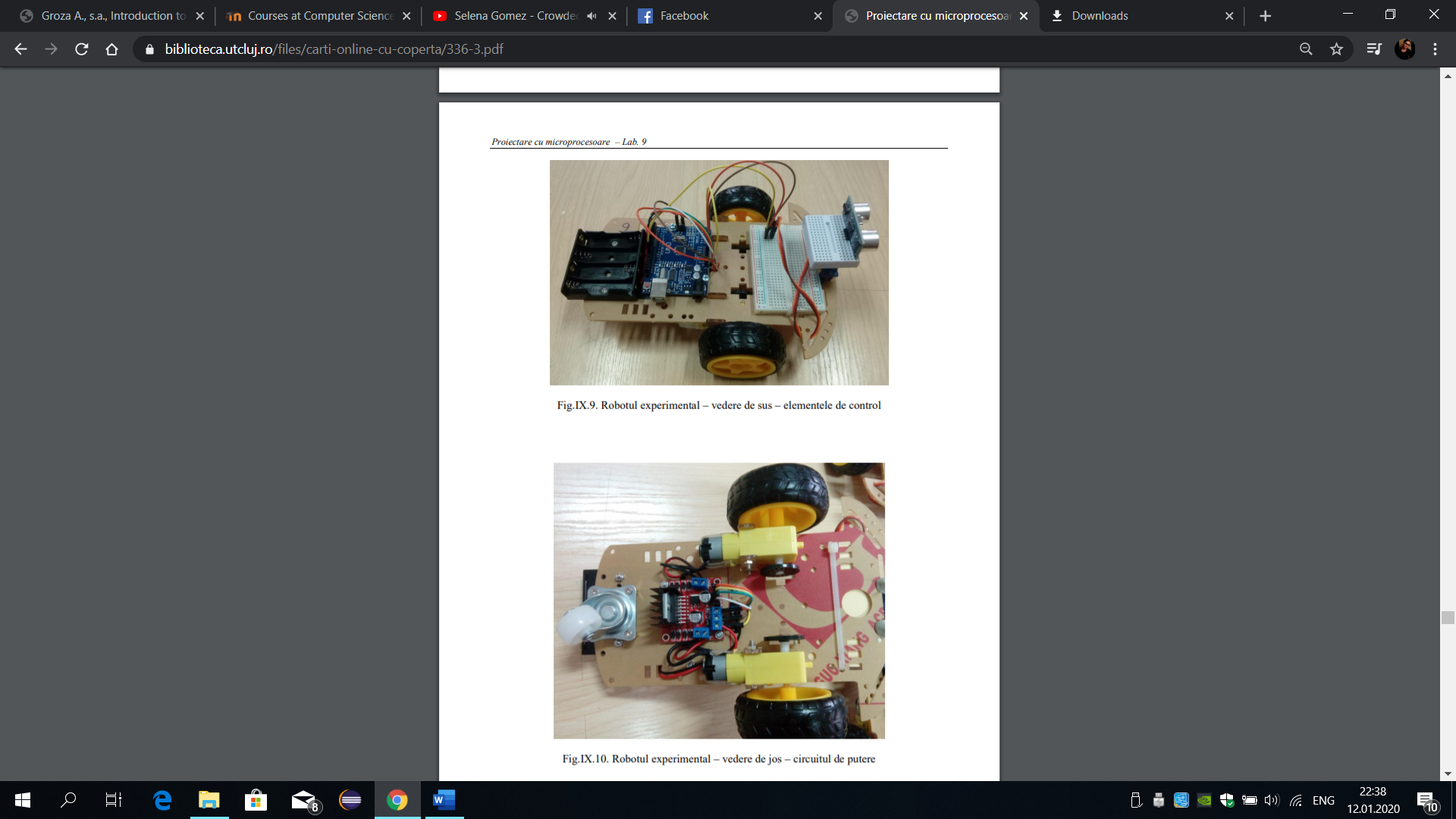


Figure 7 - robot privit de sus (fara modulul Bluetooth si buzzer)

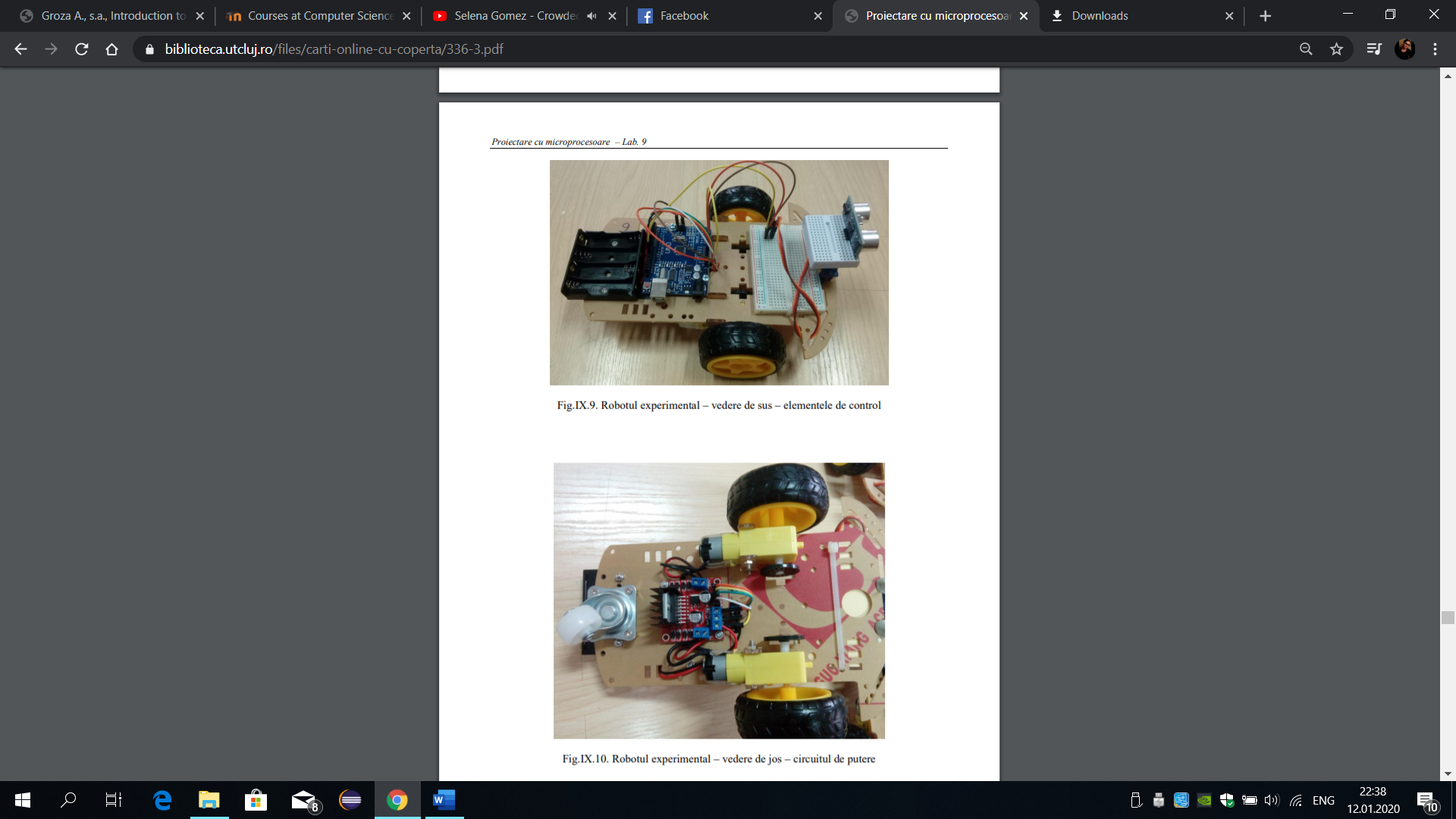
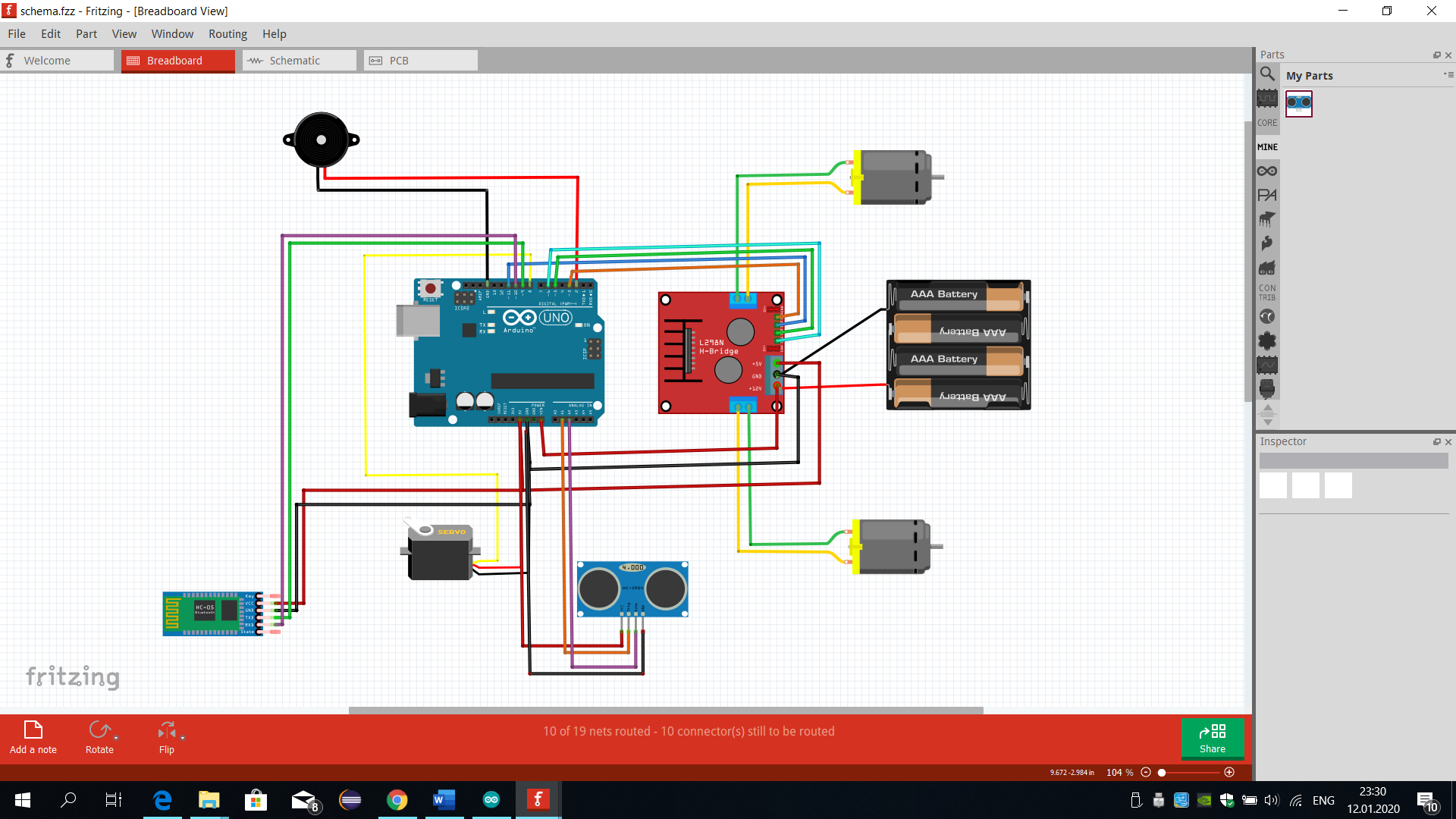


Figure 8 - robot privit de jos

## **Schema hardware**



# **Algoritm**

#include <Servo.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(9, 10); //rxPin = 9 txPin = 10;

#define mpin00 5

#define mpin01 6

#define mpin10 3

#define mpin11 11

#define echoPin A2

#define trigPin A1

int b = 0;

float durata = 0;

float distanta = 0;

char dir;

Servo srv;

void setup() {

mySerial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(13, OUTPUT);

pinMode(2, OUTPUT);

digitalWrite(mpin00, 0);

digitalWrite(mpin01, 0);

digitalWrite(mpin10, 0);

digitalWrite(mpin11, 0);

pinMode (mpin00, OUTPUT);

pinMode (mpin01, OUTPUT);

pinMode (mpin10, OUTPUT);

pinMode (mpin11, OUTPUT);

}

void StartMotor (int m1, int m2, int forward, int speed)

{

if (speed == 0) // oprire

{

digitalWrite(m1, 0);

digitalWrite(m2, 0);

}

else

{

if (forward)

{

digitalWrite(m2, 0);

analogWrite(m1, speed); // folosire PWM

}

else

{

digitalWrite(m1, 0);

analogWrite(m2, speed);

}

}

}

void delayStopped(int ms)

{

StartMotor (mpin00, mpin01, 0, 0);

StartMotor (mpin10, mpin11, 0, 0);

delay(ms);

}

void playWithServo(char dir, int pin)

{

srv.attach(pin);

float poz = srv.read();

if (dir == 'N') {

srv.write(90);

}

else if (dir == 'V') {

srv.write(180);

}

else if (dir == 'E') {

srv.write(0);

}

delay(1000);

srv.detach();

}

float calculeazaDistanta() {

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

float durata\_l = pulseIn(echoPin, HIGH);

digitalWrite(2, b);

b = 1 - b;

float distanta\_l = durata\_l / 58.2;

return distanta\_l;

}

void pornesteN() {

StartMotor (mpin00, mpin01, 1, 128);

StartMotor (mpin10, mpin11, 0, 128);

delay (500);

delayStopped(500);

}

void pornesteS() {

StartMotor (mpin00, mpin01, 0, 128);

StartMotor (mpin10, mpin11, 1, 128);

delay (500);

delayStopped(500);

}

void pornesteV() {

StartMotor (mpin00, mpin01, 1, 128);

StartMotor (mpin10, mpin11, 0, 0);

delay (500);

delayStopped(500);

}

void pornesteE() {

StartMotor (mpin00, mpin01, 1, 0);

StartMotor (mpin10, mpin11, 0, 128);

delay (500);

delayStopped(500);

}

void gasesteCale(char dir) {

int merge = 0;

float d = 0;

if (merge == 0){

if (dir != 'N') playWithServo('N', 8);

d = calculeazaDistanta();

if (d > 20) {

pornesteN();

merge = 1;

}

}

if (merge == 0){

if (dir != 'V') playWithServo('V', 8);

d = calculeazaDistanta();

if (d > 20) {

pornesteV();

merge = 1;

}

}

if (merge == 0){

if (dir != 'E' ) playWithServo('E', 8);

d = calculeazaDistanta();

if (d > 20) {

pornesteE();

merge = 1;

}}

if (merge==0){

pornesteS();

merge=1;

}

}

void loop() {

if (mySerial.available()) {

dir = mySerial.read();

if (dir != 'S') {

playWithServo(dir, 8);

distanta = calculeazaDistanta();

}

if ((distanta <= 20 && distanta >= 0) && dir != 'S') {

digitalWrite(13, HIGH);

gasesteCale(dir);

}

else {

digitalWrite(13, LOW);

if (dir == 'N') {

pornesteN();

}

else if (dir == 'S') {

pornesteS();

}

else if (dir == 'V') {

pornesteV();

}

else if (dir == 'E') {

pornesteE();

}

}

}

else {

distanta = calculeazaDistanta();

if (distanta <= 20 && distanta >= 0) {

digitalWrite(13, HIGH);

}

else {

digitalWrite(13, LOW);

}

}

}

# **Rezultate**

Atata timp cat robotul este controlat prin intermediul modului Bluetooth, el se va deplasa, daca nu exista obstacole, in directia aleasa dintre cele patru directii posibile: N, S, V, E. Daca se alege o directie ce nu poate fi executata (exista obstacol pe acea directie), robotul cauta cu ajutorul senzorului cu ultrasunete o posibila directie de deplasare. Daca acesta are obstacole pe directiile N, E si V, el se va deplasa spre S. In momentul in care se detecteaza un obstacol la o distanta mai mica de 20 de cm, LED-ul conectat la pinul 13 al placii de dezvoltare Arduino UNO se va aprinde si va ramane aprins atata timp cat obsatcolul ramane la o distanta mai mica decat 20 de cm. De asemenea, buzzer-ul conectat va incepe sa genereze sunete de frecventa redusa cand obstacolul este mai departe, frecventa acestora crescand cu apropierea obstacolului.

# **Concluzii**

In concluzie, robotul controlat cu ajutorul modulelor prezentate mai sus are doua moduri posibile de deplasare: fie controlat cu ajutorul telefonului mobil sau un laptop care are functia de Bluetooth activata, fie autonom in momentul in care intalneste obstacole. Tehnologia ce sta la baza robotului poate fi folosita pentru proiectarea unui aspirator inteligent sau a unei masinute de jucarie. Algoritmul proiectului poate fi extins pentru a fi posibila desenarea traseului urmat de robot, astfel putand fi utilizat pentru maparea si masurarea diferitelor terenuri.