**UNIVERSITATEA GHEORGHE ASACHI DIN IAȘI**

FACULTATEA DE INGINERIE ELECTRICĂ, ENERGETICĂ ȘI INFORMATICĂ APLICATĂ

**PROIECT**

la Disciplina Senzori Inteligenți

**GHIVECI ELECTRONIC**

**Realizat de:**

Popa Mirel - Software + Lider

Oloeru Dragoș - Hardware

Cotos Ionuț - Interfață Grafică

Rusu Ana-Maria - Documentație

**Grupa:** 6407

**Coordonator:** Conf. Dr. Ing. Marius BRÂNZILĂ

**- 2018 -**

# **PREZENTAREA PROIECTULUI**

ABSTRACT - Lucrarea prezentata presupune realizarea unui sistem ce asigură condițiile optime de supraviețuire a unei plante folosind un ansamblu de senzori, microprocesoare și motoare electrice.

KEYWORDS **-** #arduinouno, #project, #gui, #hardware, #idea, #microcontroller, #software, #hcsr04, #programming, #fritzing, #innovative, #circuits, #nature&technology

# **PARTE HARDWARE**

## Piese utilizate pentru realizarea proiectului

**▫ ARDUINO UNO**



Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits)6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM2 KB (ATmega328)

EEPROM1 KB (ATmega328)

Clock Speed 16 MHz

**▫ HC-SR04**



Working Voltage DC 5 V

Working Current 15mA

Working Frequency 40Hz

Max Range 4m

Min Range 2cm

MeasuringAngle 15 degree

Trigger Input Signal 10uS TTL pulse

Echo Output Signal Input TTL lever signal and the range in proportion

Dimension 45\*20\*15mm

**▫ SERVOMOTOR TOWERPRO SG-90**



Operating Voltage is +5V typically

Torque: 2.5kg/cm

Operating speed is 0.1s/60°

Gear Type: Plastic

Rotation : 0°-180°

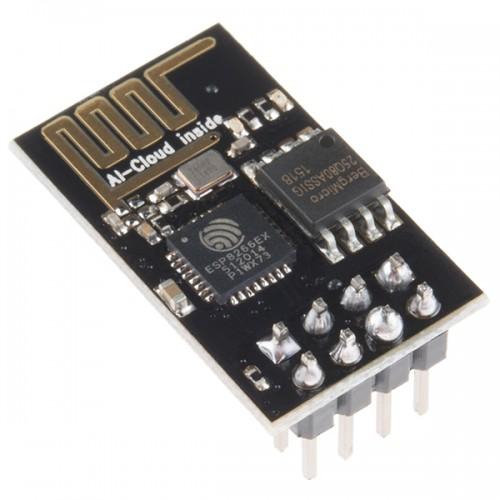
Weight of motor : 9gm

**▫ MOTOARE DE CURENT CONTINUU**



Motor Specifications  
Operating voltage**:** 3V ~ 6V DC (recommended value 5V)  
Maximum torque**:** 800g.cm  
Speed without load: 90±10rpm  
Reduction ratio: 1:48  
No Load current: 190mA(max.250mA)  
Stall Current: ~1A  
Strong anti-interference on this motor keeps it safe around micro-controllers.

**▫ ESP2866:**



802.11 b/g/n protocol

Wi-Fi Direct (P2P), soft-AP

Integrated TCP/IP protocol stack

Integrated TR switch, balun, LNA, power amplifier and matching network

Integrated PLL, regulators, and power management units

+19.5dBm output power in 802.11b mode

Integrated temperature sensor

Supports antenna diversity

Integrated low power 32-bit CPU could be used as application processor

SDIO 2.0, SPI, UART

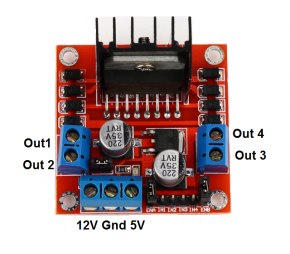
STBC, 1×1 MIMO, 2×1 MIMO

A-MPDU & A-MSDU aggregation & 0.4µs guard interval

Wake up and transmit packets in < 2ms

Standby power consumption of < 1.0mW (DTIM3

**▫ L298N**



Operating supply voltage up to 46V

Low saturation voltage

Total DC current up to 4A

Logical \”0\” input voltage up to 1.5V (high noise immunity)

Overtemperature protection

**Photorezistențe și rezistențe de 10 k:**

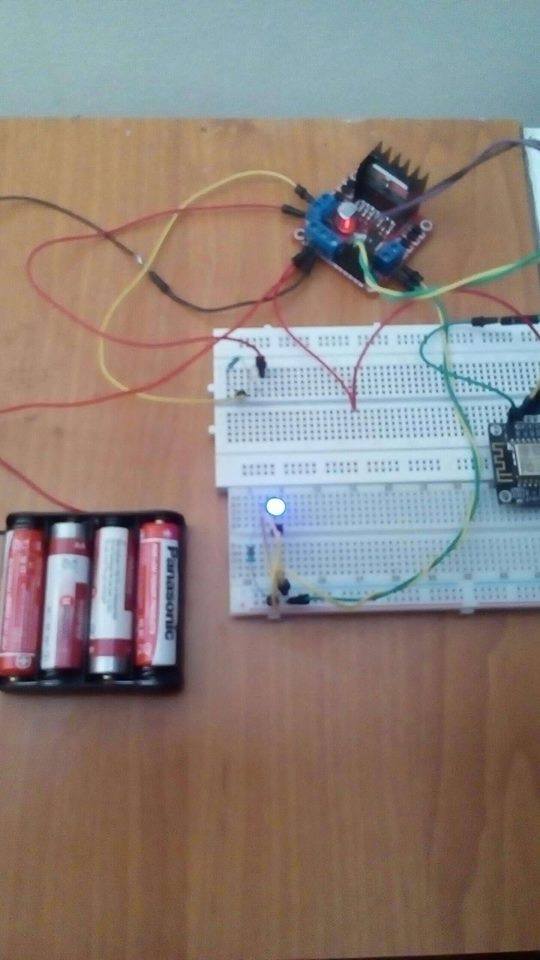


**Baterii și șasiu:**

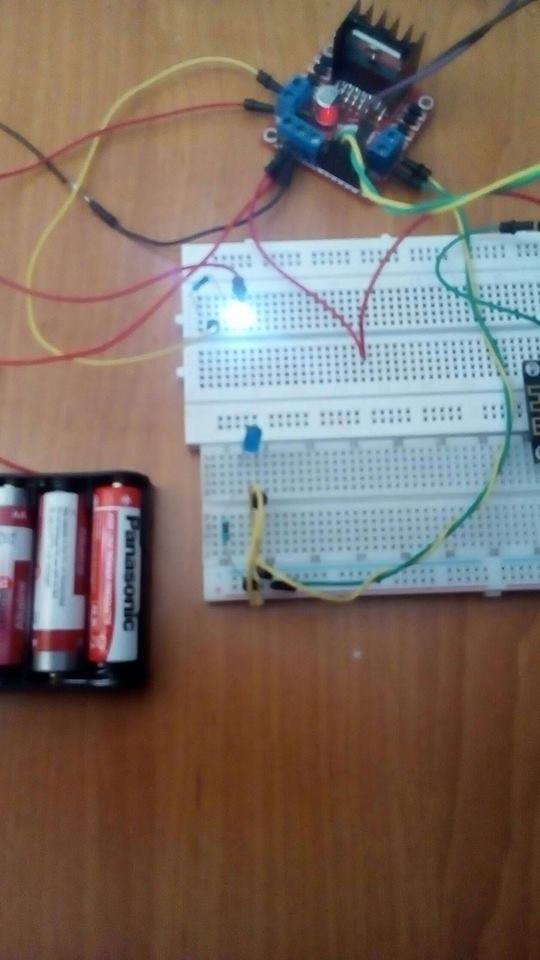


## Prezentare parte practică

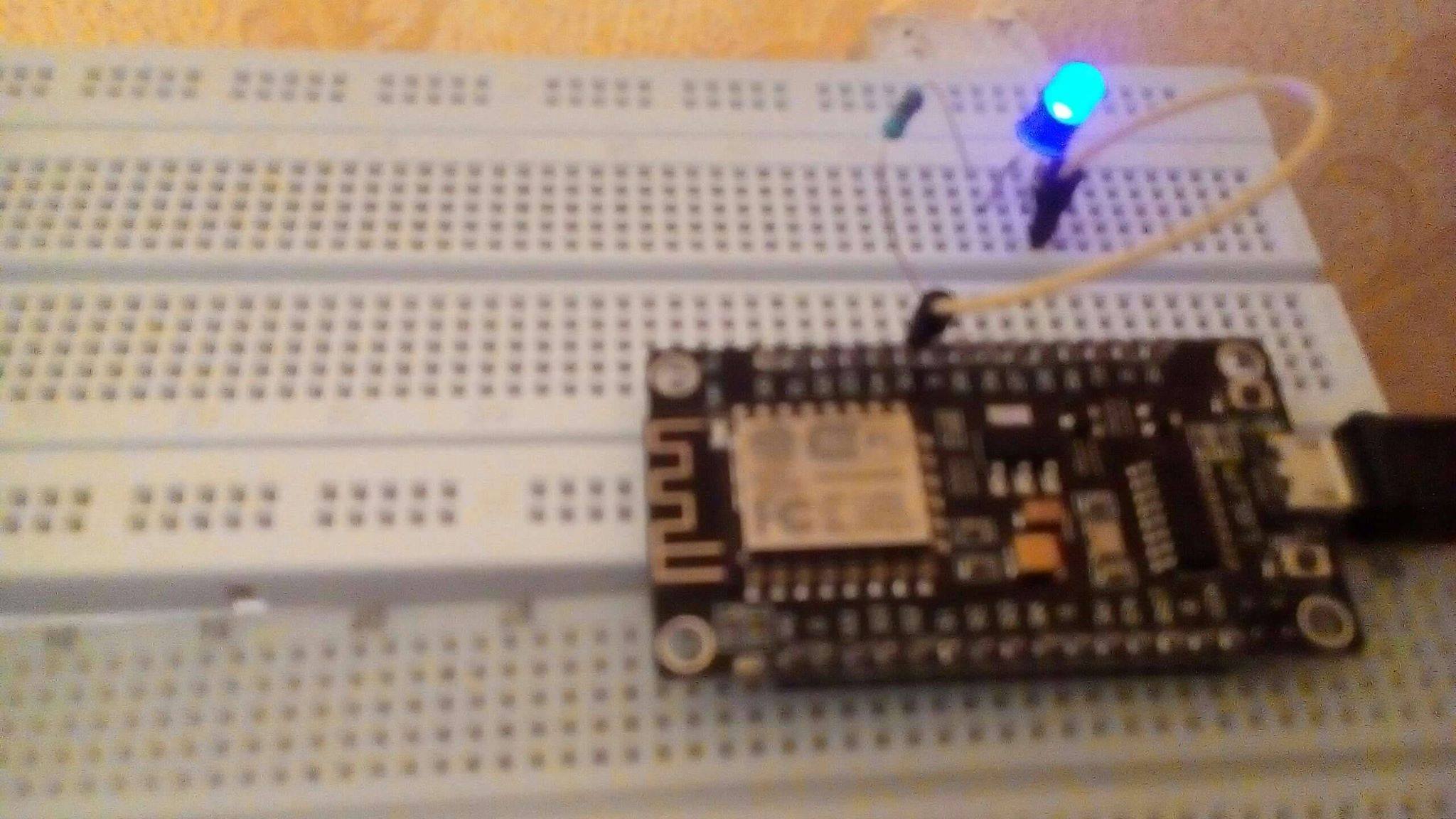
**\*DRIVER MOTOR 1\***



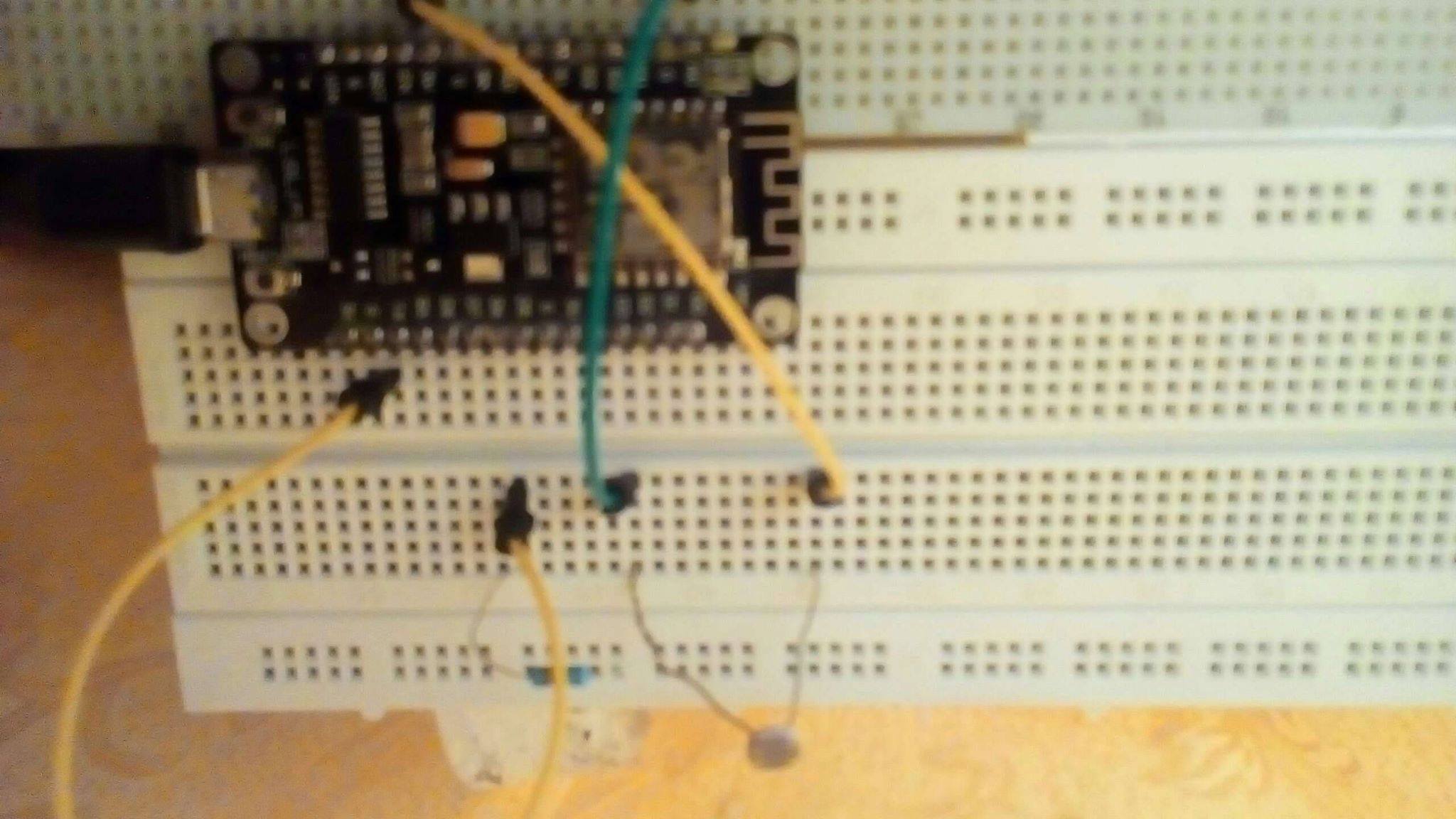
**\*TEST DRIVER 2\***



**\*TEST NOD MCU\***



**\*TEST FOTOREZISTENȚĂ\***



**PROCESUL DE LIPIRE**



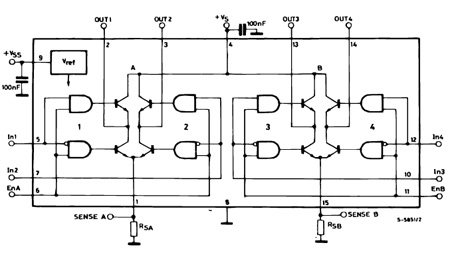
**ASAMBLARE**

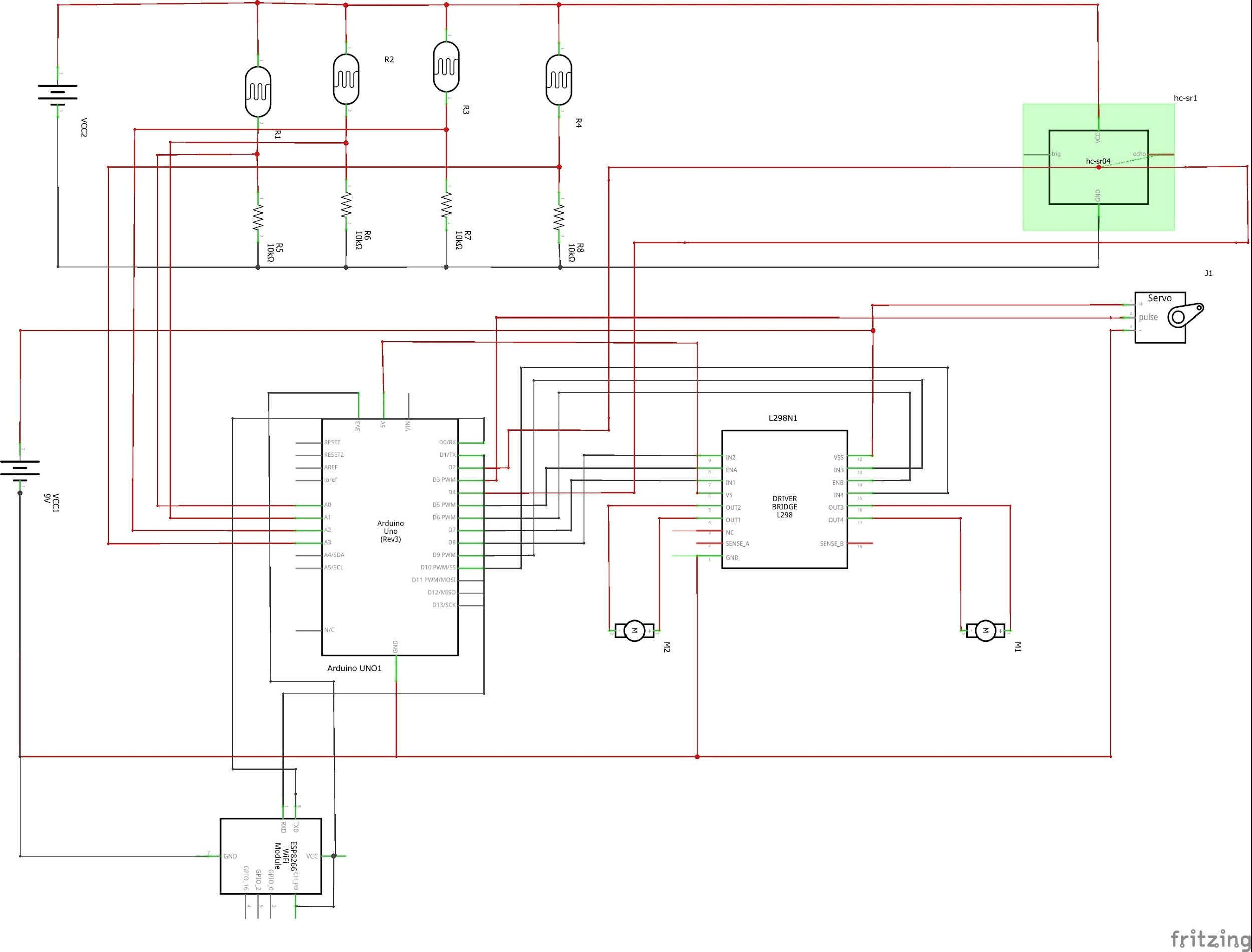




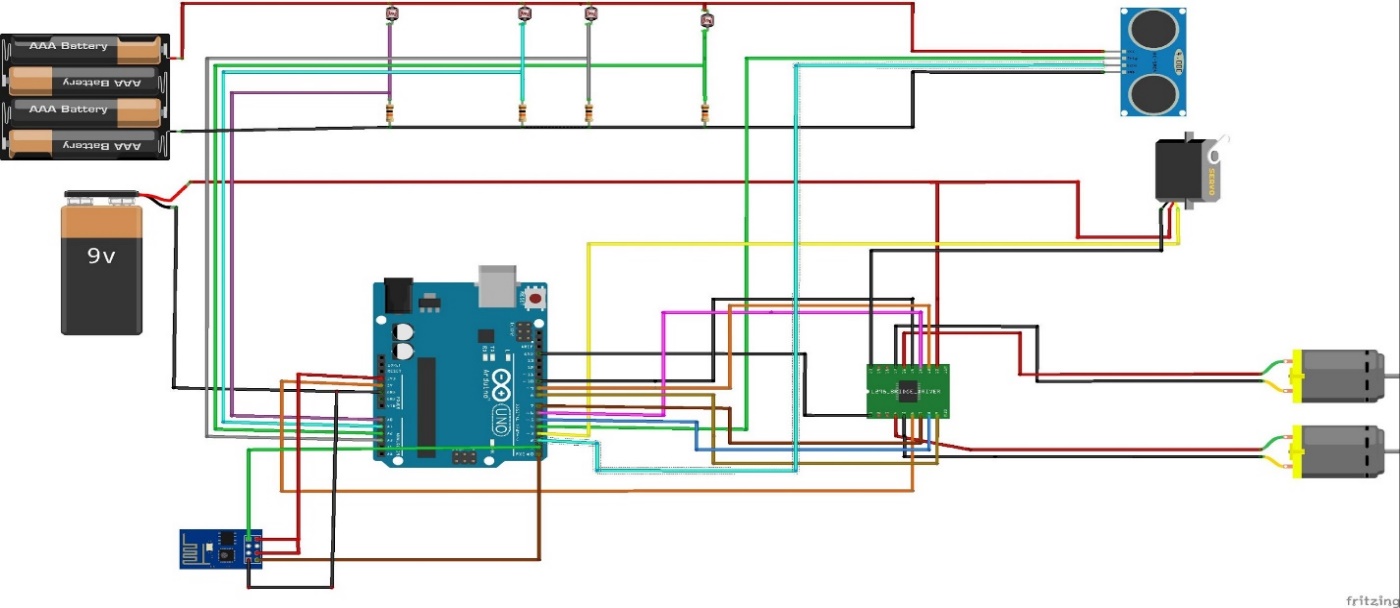
# **PARTE SOFTWARE**

## Schema bloc

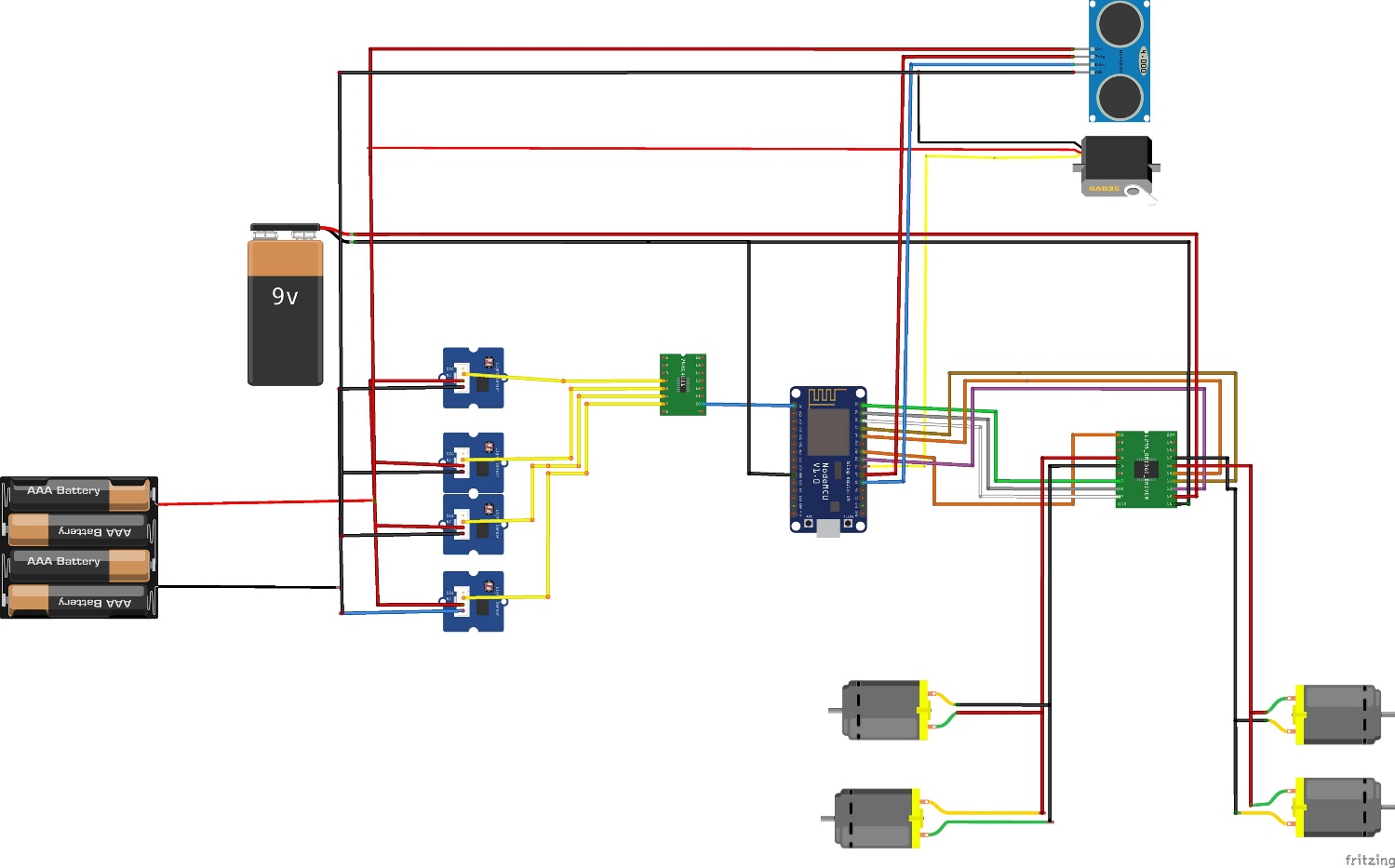




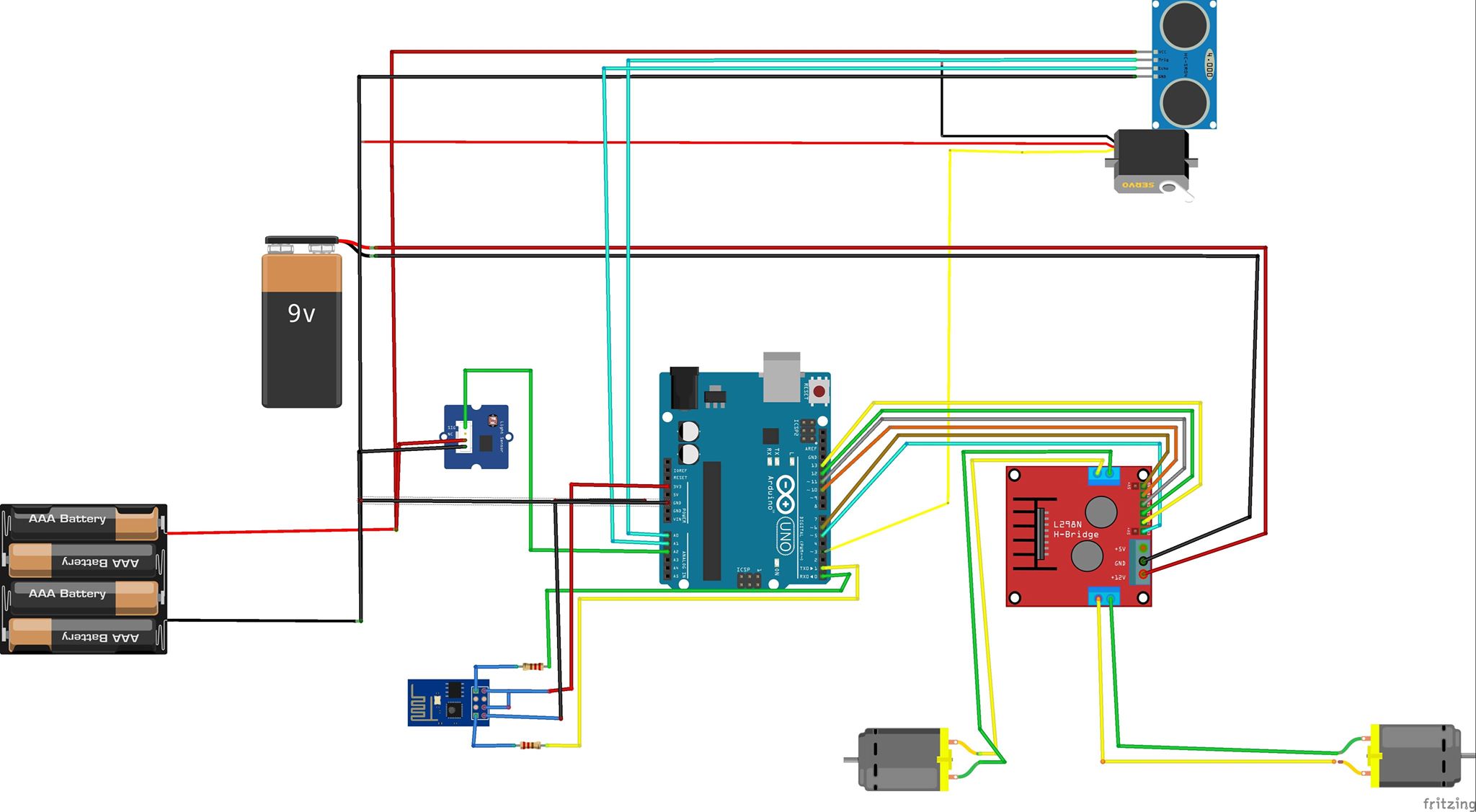
\*pentru realizarea schemelor s-a folosit softul Fritzing\*

 **\*SCHEMA ELECTRICĂ INIȚIALĂ\***

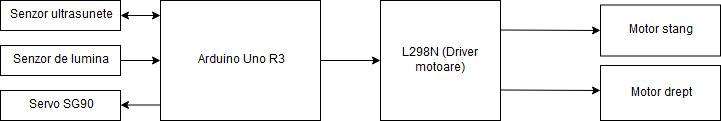
## **\*SCHEMA ELECTRICĂ FINALĂ\***



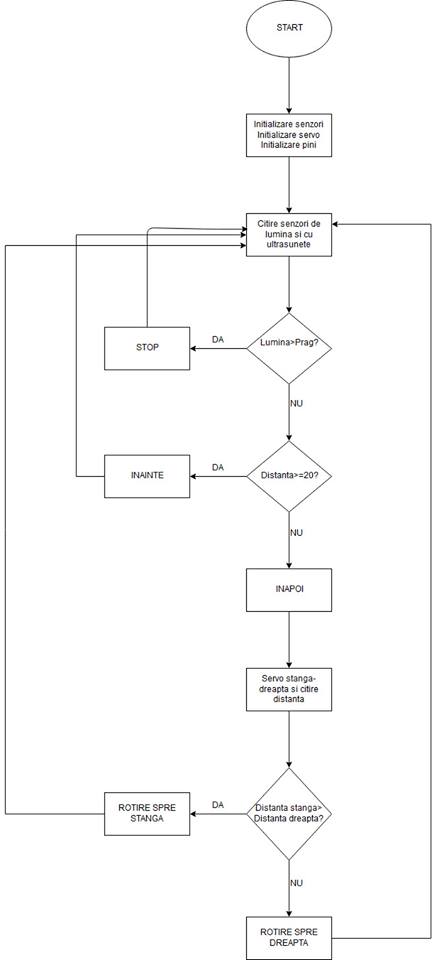
Schema-finala



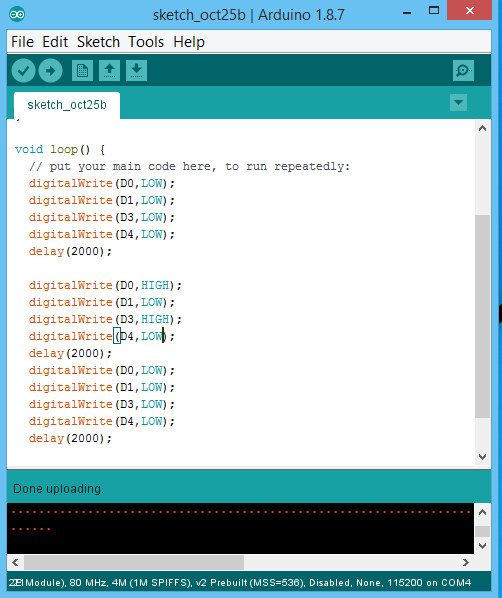
**\*SCHEMA BLOC\***



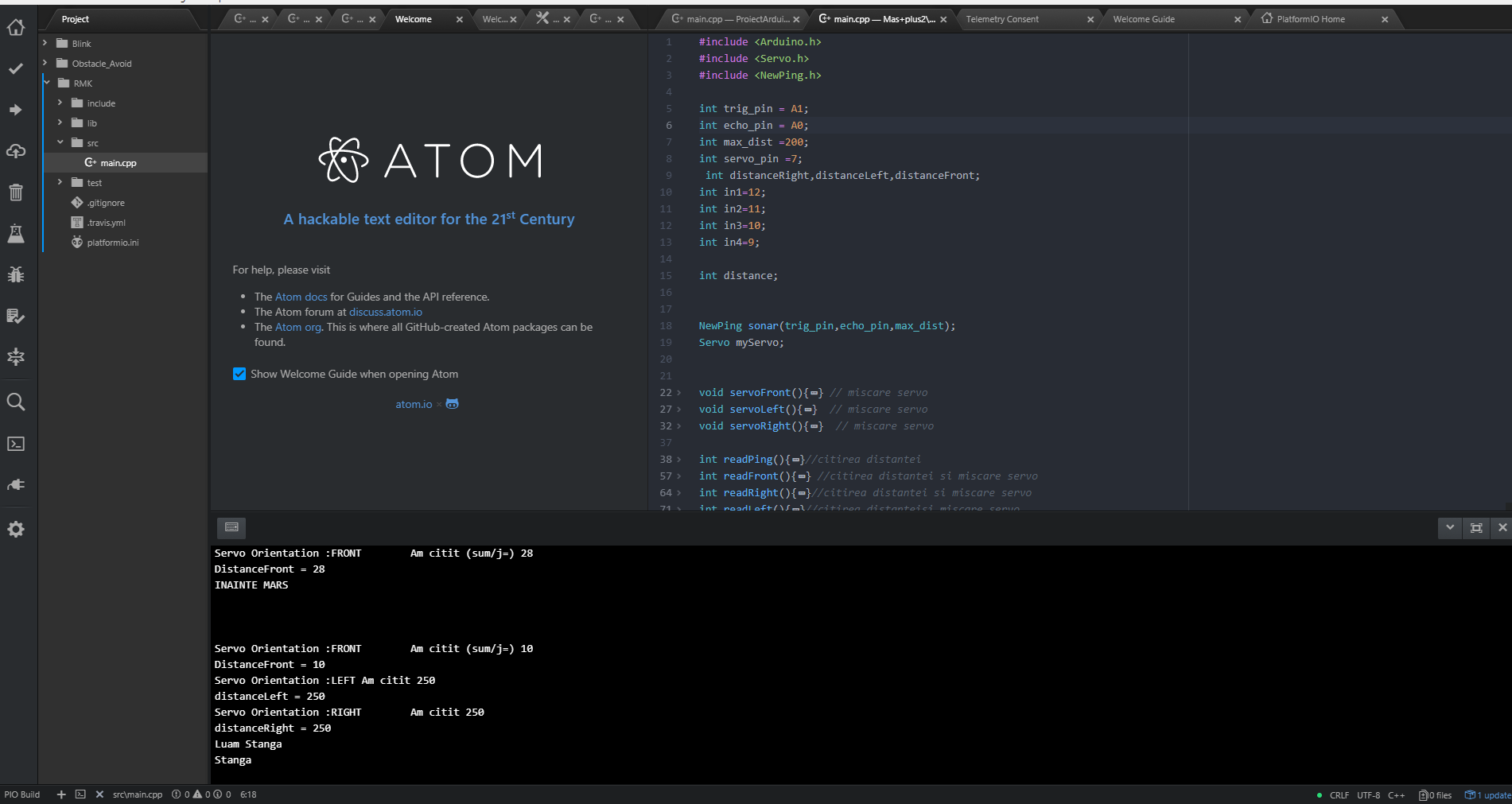
**\*ORGANIGRAMA COD\***



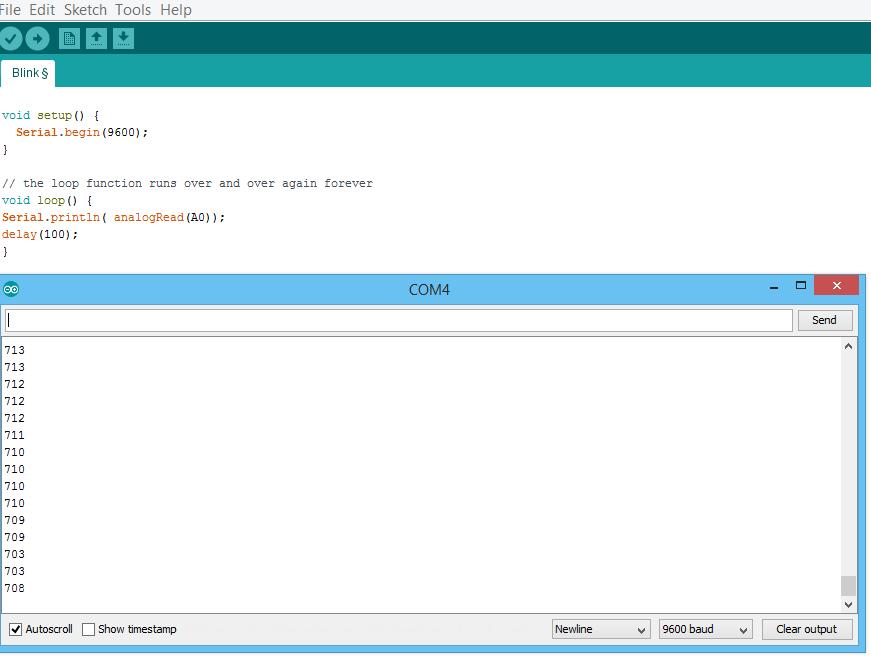
**\*SCREENSHOT COD TEST DRIVER MOTOR\***



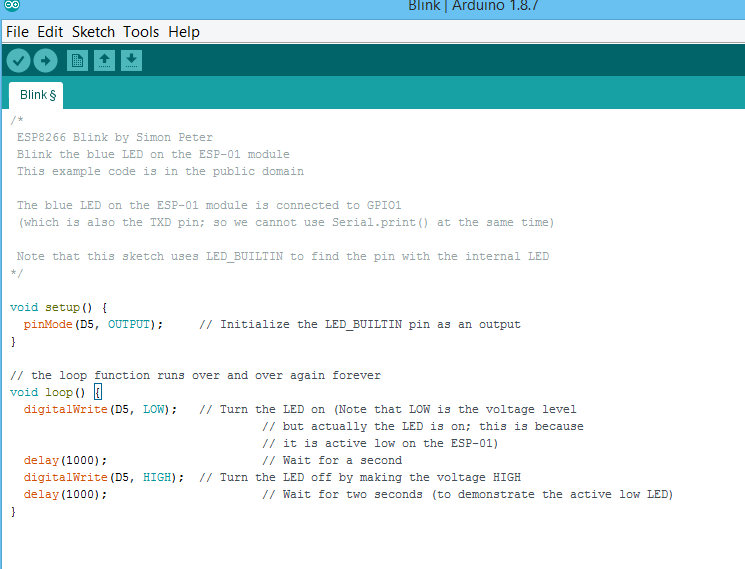
**\*TESTAREA CODULUI ÎN ATOM\***



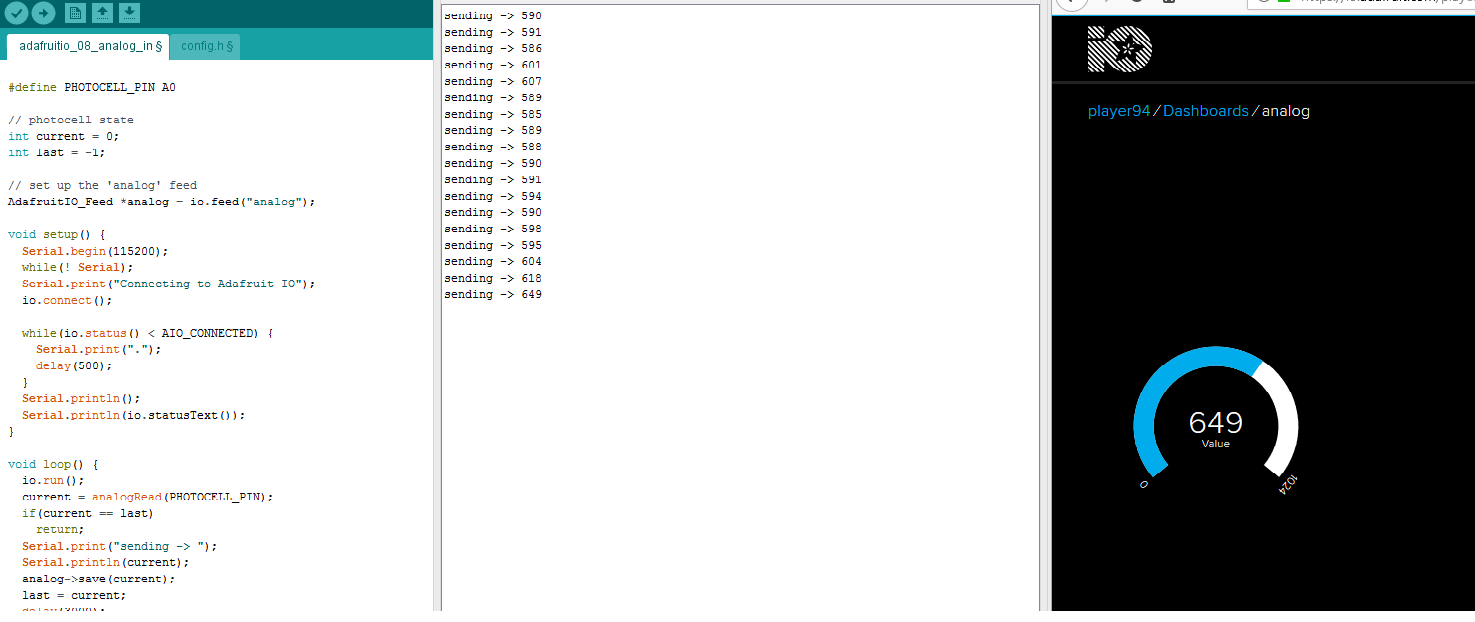
**\*SCREENSHOT COD FOTOREZISTENȚĂ\***



**\*SCREENSHOT COD TEST NOD MCU\***



**\*SCREENSHOT WI FI\***



## Cod sursă

#include <Arduino.h>

#include <Servo.h>

#include <NewPing.h>

#define trig\_pin A1

#define echo\_pin A0

#define light\_senzor\_1 A2

#define light\_senzor\_2 A3

#define servo\_pin 3

#define in1 12 **//stanga**

#define in2 13 **//stanga**

#define in3 11 **//dreapta**

#define in4 10 **//dreapta**

#define enA 6 **//motorul stang**

#define enB 5 **//motorul drept**

#define max\_dist 200

int distanceRight,distanceLeft,distanceFront,distance;

NewPing sonar(trig\_pin,echo\_pin,max\_dist);

Servo myServo;

void servoFront(){

myServo.write(80);

Serial.print("Servo Orientation :FRONT\t");

delay(500);

} **// miscare servo**

void servoLeft(){

Serial.print("Servo Orientation :LEFT\t");

myServo.write(180);

delay(500);

} **// miscare servo**

void servoRight(){

Serial.print("Servo Orientation :RIGHT\t");

myServo.write(10);

delay(500);

} **// miscare servo**

int readPing(){

int sum=0;

int j=0;

int distance[10];

for(int i=0;i<5;i++){

distance[i]=sonar.ping\_cm();

sum=sum+distance[i];

delay(50);

if(distance[i]>0) j++;}

if((sum/j)>250 || (sum==0)){

//Serial.println("Am citit 250");

return 250;

}

else{

//Serial.print("Am citit (sum/j=) ");

//Serial.println(sum/j);

return sum/j;

}

}**//citirea distantei**

int readFront(){

servoFront();

distanceFront=readPing();

Serial.print("DistanceFront = ");

Serial.println(distanceFront);

return distanceFront;

} **//citirea distantei si miscare servo**

int readRight(){

servoRight();

delay(100);

distanceRight=readPing();

Serial.print("distanceRight = ");

Serial.println(distanceRight);

return distanceRight;

}**//citirea distantei si miscare servo**

int readLeft(){

servoLeft();

delay(100);

distanceLeft=readPing();

Serial.print("distanceLeft = ");

Serial.println(distanceLeft);

return distanceLeft;

}**//citirea distantei si miscare servo**

void move\_front(){

digitalWrite(in1,HIGH);

digitalWrite(in2,LOW);

digitalWrite(in3,HIGH);

digitalWrite(in4,LOW);

}

**//functia pentru deplasarea masinii inainte**

void move\_right(){

digitalWrite(in1,HIGH);

digitalWrite(in2,LOW);

digitalWrite(in3,LOW);

digitalWrite(in4,HIGH);

}

**//functia pentru orientarea masinii in partea dreapta**

void move\_left(){

digitalWrite(in1,LOW);

digitalWrite(in2,HIGH);

digitalWrite(in3,HIGH);

digitalWrite(in4,LOW);

}

**//functia pentru orientarea masinii in partea stanga**

void move\_back(){

digitalWrite(in1,LOW);

digitalWrite(in2,HIGH);

digitalWrite(in3,LOW);

digitalWrite(in4,HIGH);

}

**//functia pentru deplasarea masinii**

void move\_stop(){

digitalWrite(in1,LOW);

digitalWrite(in2,LOW);

digitalWrite(in3,LOW);

digitalWrite(in4,LOW);

}

**//funcite pentru oprirea masinii**

void setup() {

analogWrite(enA,255);

analogWrite(enB,255);

Serial.begin(9600);

myServo.attach(servo\_pin);

servoFront();

distanceFront=readFront();

}

void loop() {

while(readFront()>=25){

move\_front();

Serial.println("merge inainte");

}

move\_stop();

distanceLeft=readLeft();

delay(200);

distanceRight=readRight();

delay(200);

if(distanceLeft>=distanceRight && distanceLeft>=25 )

{

Serial.println("face stanga");

move\_left();

delay(400); **//delay pentru intoarcerea masinii la 90 de grade la stanga**

}else if(distanceLeft<distanceRight && distanceRight>=25)

{

move\_right();

Serial.println("face dreapta");

delay(400); **//delay pentru intoarcerea masinii la 90 de grade la dreapta**

} else {

Serial.println("se intoarce la 180 de grade");

if(distanceRight>=5) {move\_right(); delay(800); Serial.println("Dreapta imprejur");}

else

if(distanceLeft>=5) {move\_left(); delay(800); Serial.println("Stanga imprejur");}

}

Serial.println("\n\n");

}

# 

# **CONCLUZII**

Sperăm că acest proiect va inspira pe cât mai multe persoane să își creieze un mic colț verde cu ajutorul tehnologiei. Orice device care poate ține în viață o plantă are mari șanse să contribuie atât la dezvoltarea inteligenței artificiale cât și la îmbunătațirea relației natură-tehnologie. Acest concept poate fi cu atât mai mult exploatat cu cât oamenii devin din ce in ce mai interesați de ideea de a-și cultiva singuri hrana pentru consumul propriu. Asemenea device-uri pot reduce semnificativ timpul destinat îngrijirii unei plante, dezvoltând ideea de a face un lucru într-un mod inteligent, dar și într-un timp cât mai scurt.

# **BIBLIOGRAFIE**

# **CUPRINS**

[**PREZENTAREA PROIECTULUI** 2](#_Toc531043498)

[ABSTRACT 2](#_Toc531043499)

[KEYWORDS 2](#_Toc531043500)

[**PARTE HARDWARE** 3](#_Toc531043501)

[Piese utilizate pentru realizarea proiectului 3](#_Toc531043502)

[Prezentare parte practică 10](#_Toc531043503)

[**PARTE SOFTWARE** 15](#_Toc531043504)

[Schema bloc 15](#_Toc531043505)

[Schema electrică finală 17](#_Toc531043506)

[Cod sursă 23](#_Toc531043507)

[**CONCLUZII** 27](#_Toc531043508)

[**BIBLIOGRAFIE** 28](#_Toc531043509)

[**CUPRINS** 29](#_Toc531043510)