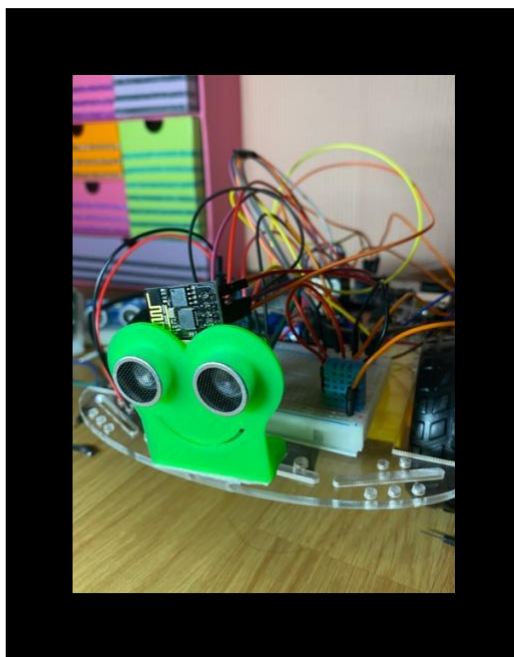




# IOT - ROBOT TRANSMITATOR DE TEMPERATURA CONTROLAT REMOTE



Autori: Muresan Daiana Emanuela

Materia: Retele de Calculatoare

Grupa: 30145

Universitatea Tehnica Cluj Napoca

Facultatea de Automatica si Calculatoare

**2020**

# Cuprins

<b>1</b>	<b>INTRODUCERE.....</b>	<b>2</b>
1.1	CONTEXT GENERAL .....	2
1.2	OBIECTIVE.....	2
1.3	SPECIFICAȚII.....	2
<b>2</b>	<b>ANALIZĂ, PROIECTARE, IMPLEMENTARE.....</b>	<b>3</b>
2.1	ACHIZITIONAREA COMPONENTELOR NECESARE SI FAMILIARIZAREA CU MEDIUL DE DEZVOLTARE ARDUINO IDE SI APLICATIA BLYNK.....	3
2.2	TESTAREA SI ANALIZAREA FIECAREI COMPONENTE IN PARTE .....	4
2.3	ASAMBLAREA TUTUROR COMPONENTELOR SI TESTAREA.....	11
<b>3</b>	<b>CONCLUZII .....</b>	<b>15</b>
3.1	REZULTATE OBȚINUTE .....	15
3.2	DIRECȚII DE DEZVOLTARE .....	15
<b>4</b>	<b>BIBLIOGRAFIE.....</b>	<b>16</b>

# 1 Introducere

## 1.1 Context general

Acest proiect reprezinta un sistem inteligent controlat prin comanda virtuala si care monitorizeaza si afiseaza in timp real temperatura, in cadrul unei aplicatii mobile. Pentru aceasta am ales sa utilizam un ansamblu de tip robot cu 2 roti si un senzor ultrasonic, un senzor de umiditate si temperatura si un modul Wi-Fi, implementate cu ajutorul placii de dezvoltare MEGA2560.

Robotul reprezinta o combinatie intre inteligenta computationala si un ansamblu de componente fizice, avand diferite utilitati in viata de zi cu zi, cum ar fi : utilizarea lui ca subansamblu in aspirator-robot sau folosirea lui pentru transmitere diferitelor date preluate de senzori(ex: temperature/umiditate/prezenta umana in IOT).

Am ales IoT – Internet of things - deoarece conectarea la internet a unor ansambluri de componente fizice devine un subiect din ce in ce mai popular si dezvoltat in era tehnologiei. Tot in jurul nostru devine “inteligent”, “programat”, motiv pentru care este importanta cunoasterea conceptelor care stau la baza IoT.

O posibila definitie pentru IoT o reprezinta: “O retea de obiecte conectate la internet capabile sa colecteze si sa faca schimb de date utilizand senzori incorporati”. Astfel, de la lampi sau becuri din casa, masini de spalat, usi, toate pot fi controlate de la distanta. Acesta este IoT.

Proiectul de fata poate sta la baza unui sistem de citire a temperaturii dintr-o incapere, avand posibilitatea de a culege date legate de temperatura sau umiditate de oriunde, putand de asemenea sa se deplaseze pentru a colecta date din mai multe locatii.

## 1.2 Obiective

Proiectul a fost creat in scopul dezvoltarii si implementarii unui sistem cu ajutorul mediului de programare Arduino IDE si controlarea miscarii acestuia prin conexiune Wi-Fi cu ajutorul aplicatiei mobile Blynk, instalata pe un smartphone.

Un alt obiectiv al proiectului este afisarea temperaturii in timp real, in cadrul aplicatiei Blynk, prin intermediul unui sensor DHT11 digital de temperatura si umiditate.

## 1.3 Specificatii

Acest robot inteligent are un modul Wi-Fi conectat la reseaua locala prin intermediul caruia a fost posibila conectarea la aplicatia mobila. Rotile robotului sunt comandate prin intermediul unui modul cu dubla punte H cu care se regleaza si directia, putand executa miscari stanga, dreapta, fata, spate. Avem de asemenea si un senzor ultrasonic, positionat in partea din fata a robotului, care colecteaza date din mediul exterior. Pentru evitarea coliziunii cu obiectele din jur s-a introdus un led virtual care lumineaza in momentul in care senzorul ultrasonic detecteaza o distanta contraindicata miscarii inainte, astfel putand stii cand sa ne oprim si se va impiedica impactul.

In acelasi timp, se face o transmisie in timp real a temperaturii pe aplicatia mobila, sistemul avand un rol de monitorizare.



## 2.2 Testarea si analizarea fiecărei componente in parte

- Placa de dezvoltare MEGA 2560

Arduino Mega 2560 este o placa de dezvoltare destinata proiectelor complexe, dispunand de 54 porturi digitale I/O (dintre care 15 se pot folosi ca porturi PWM si 4 porturi seriale hardware) si 16 analogice. Are la baza un microcontroller ATmega2560 pe 8 biti, care lucreaza la o frecventa de 16 MHz si reprezinta "creierul" intregului proiect. Placa este capabila sa comunice prin protocoalele de comunicare : SPI, UART si TWI.

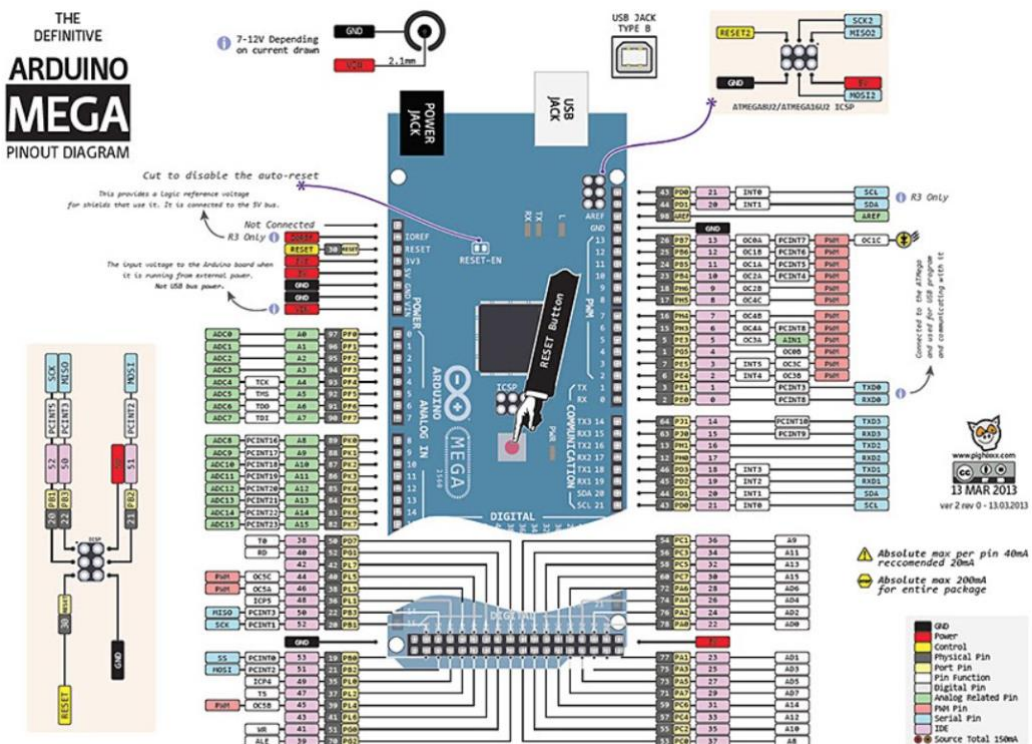


Fig 2.2 Datasheet

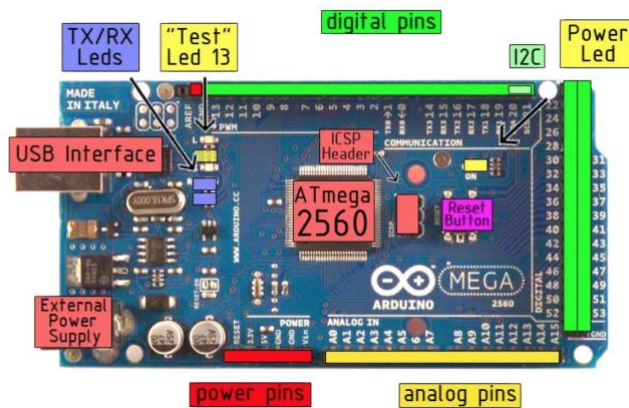


Fig 2.3 Arduino Mega2560

- **Senzor ultrasonic HC – SR04**

Detectarea obstacolelor este posibilă datorită senzorului ultrasonic. Acesta utilizează un sonar pentru a determina distanța față de un obiect, fiind capabil să detecteze obiecte de la distanța de 2cm până la 450 cm cu o precizie de 0.3 cm.

Principiul de funcționare constă în emiterea unor unde ultrasonice de frecvență înaltă la intervale de timp regulate. Dacă lovesc un obiect, acestea sunt trimise înapoi ca semnale ecou la senzor, care calculează distanța până la obiect pe baza intervalului de timp dintre emiterea semnalului și receptarea ecoului.

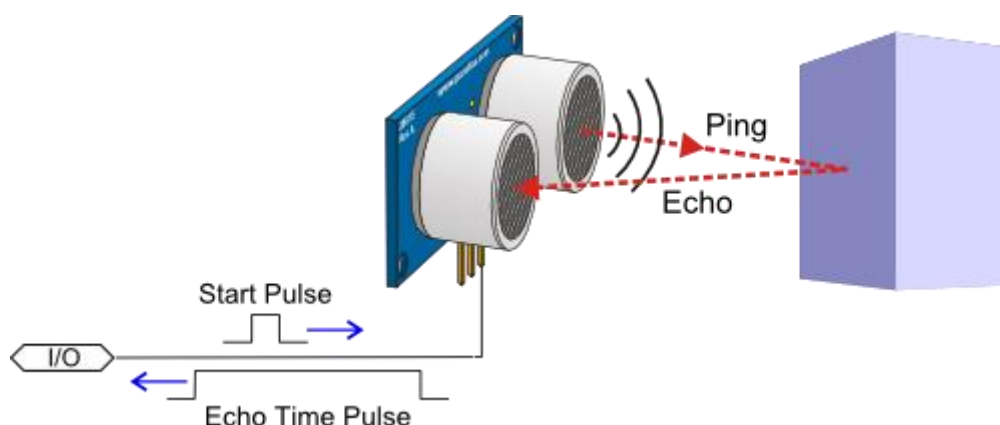


Fig 2.4 Modul de funcționare

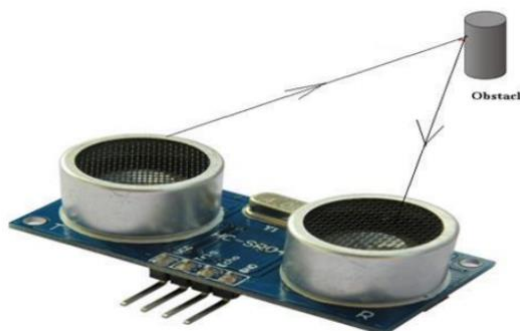


Fig 2.5 Senzor ultrasonic

Senzorul are 4 pini – VCC(5V), Trig, Echo și GND.

Pinul TRIG este ieșirea care trimite unde ultrasonice odată la 10 microsecunde și transmite datele la microcontroller, iar pinul ECHO este intrarea care primește datele din mediul exterior.

Astfel, se citește timpul de primire a ecoului în microsecunde și se calculează distanța cu formula

$Distanța = (durata / 2) / 29$  în cm.

Viteza sunetului este  $340\text{m/s} = 29\text{ cm/microsecunde}$ , iar noi împărțim la 2 deoarece unda parcurge 2 distanțe egale, spre obstacol și înapoi.

Pentru simplitatea calculelor se poate utiliza biblioteca NewPing care are metode predefinite pentru calculul distanței în inch sau cm.

Testarea senzorului am realizat-o prin diferite aplicatii precum masurarea distantei pana la un obstacol si afisarea acesteia in Serial Monitor sau aprinderea unor led-uri corespunzatoare(RGB).

In proiect senzorul se foloseste pentru detectarea obiectelor, iar la o distanta mai mica de 20cm se va aprinde un led rosu.

- **2 motoare de curent continuu 3V-6V cu reductor**

Motoarele sunt folosite pentru a roti rotile robotului. Acesta sunt de curent continuu cu reductor 1:48. Diametrul rotii pe care o pot controla este de 65mm.

Motoarele sunt controlate prin intermediul modului L298N, conectand corespunzator pinii la motor si la microcontroller.



- **L298N modul cu punte H dubla**

Acest modul este utilizat pentru a putea comanda cele 2 motoare de curent continuu care controleaza rotile robotului. L298N este un circuit integrat monolitic de voltaj si curent mare, dubla puntea H proiectat sa accepte nivele logice standard TTL pentru control.

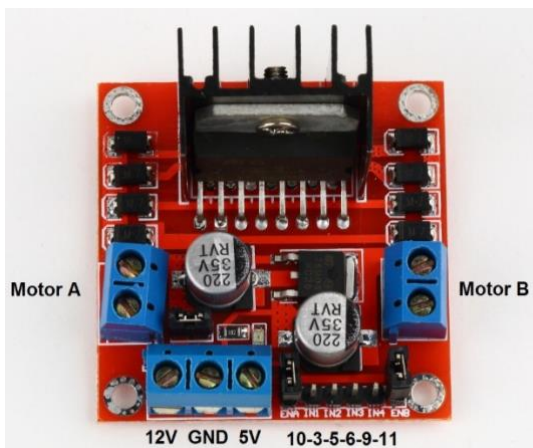
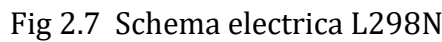


Fig 2.6 Modul L298N

Obsevam ca aceasta are 2 punti H, A si B la care am conectat cele 2 motoare, avand in vedere polaritatea acestora. Pinii Enable A si B permit activarea sau dezactivarea celor 2 punti independent.





ENA	IN1	IN2	Starea motorului A (in curent continuu)
0	X	X	Stop
1	0	0	Frana
1	0	1	Rotatie in sensul acelor de ceasornic
1	1	0	Rotatie in sensul invers acelor de ceasornic
1	1	1	Frana

Se conecteaza corespunzator pinii IN1, IN2, IN3, IN4 la pinii PWM ai placii de dezvoltare si se verifica functionarea corespunzatoare amotoarelor – rotire in ambele directii si franare.

Breadboard-ul este o placuta care ne permite inserarea componentelor electronice si crearea unui "prototip" dintr-un circuit electric. Principiul pe care se bazeaza este : vertical pe cele 2 linii laterale unde se realizeaza alimentarea VCC(+) si GND(-) si pe orizontala pentru conectarea componentelor.





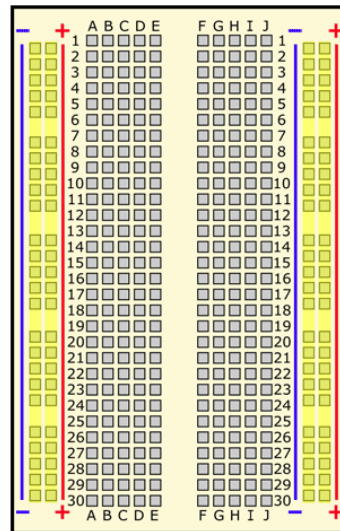


Fig 2.10 Breadboard principiu de functionare

- **Modul Wi-Fi ESP8266**

Modulul Wifi este elementul cheie pentru realizarea partii de IoT. Cu ajutorul acestuia se poate face conexiunea cu aplicatia Blynk si, mai apoi, controlarea robotului cu widget-urile puse la dispozitie de blynk.

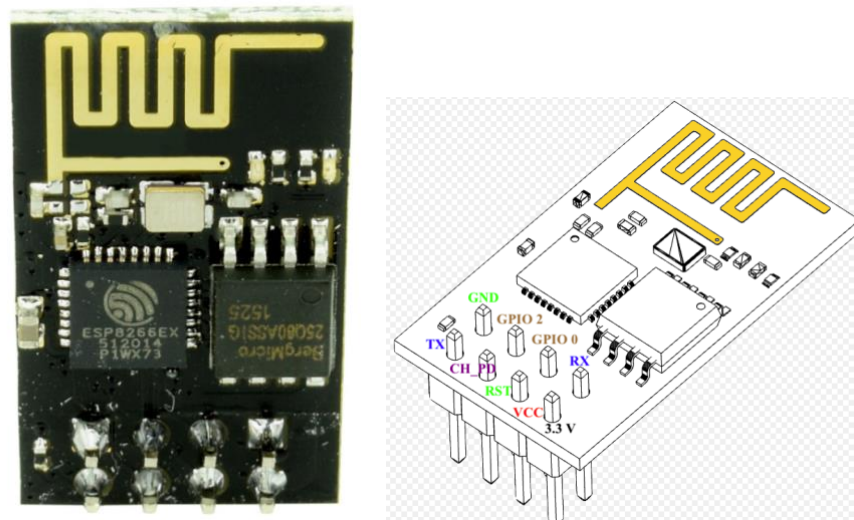


Fig 2.11 Modul Wifi

De remarcat este faptul ca trebuie facuta conexiunea corespunzatoare a pinilor, in special TX si RX – pinii de comunicare (prin TX se transmit octeti, prin RX se citesc octeti), care, avand in vedere utilizarea placii MEGA2560, vor fi conectati la pinii Serial1 – 18 si 19. Se alege baud\_rate 115200 deoarece este o versiune mai veche a modulului prezent in proiect.

BOARD	USB CDC NAME	SERIAL PINS	SERIAL1 PINS	SERIAL2 PINS	SERIAL3 PINS
Uno, Nano, Mini		0(RX), 1(TX)			
Mega		0(RX), 1(TX)	19(RX), 18(TX)	17(RX), 16(TX)	15(RX), 14(TX)

In continuare s-au realizat cateva testari de transmitere a temperaturii in Serial.Monitor careu au avut success.

Urmatorul pas a constat in conectarea prin intermediul acestui Wi-Fi la aplicatia mobila Blynk. Aplicatia mobile consta in realizarea unui proiect pentru care se selecteaza placa de dezvoltare si

modul de comunicare, după care transmite pe mail un Token de autentificare cu care se face conexiunea în Arduino IDE.

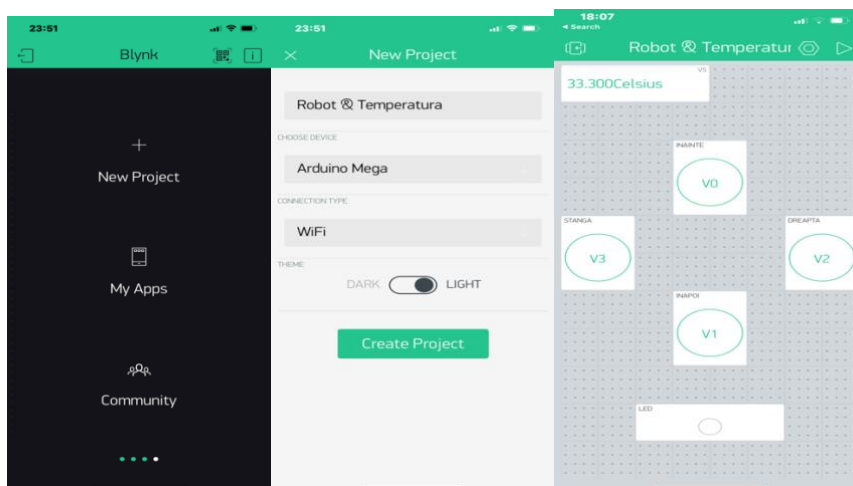


Fig 2.12 Interfața aplicației Blynk

Blynk are o interfață grafică simplă și ușor de folosit, astfel încât se pot adăuga Widget-uri prin “drag&drop”. Pentru fiecare widget adăugat trebuie specificat pin-ul prin care se va face referire la el mai târziu, existând posibilitatea de a alege pin analogic/digital/virtual. Spre deosebire de pinii analogi și digitali, pinii virtuali se referă la pinii fizici de pe microcontroller.

Pentru controlul robotului s-au ales 4 butoane cu 4 funcționalități diferite- deplasare în față, în spate, în dreapta, în stanga. De asemenea s-a ales un widget pentru afișarea temperaturii și un led care afișează roșu în cazul în care robotul se află aproape de un obstacol. (<20cm)

După ce conexiunile au fost făcute fizic și implementarea în interiorul codului din Arduino IDE, unde trebuie specificat Token-ul de autentificare, wifi-ul la care va fi conectat dispozitivul și parola acestuia, se așteaptă realizarea conexiunii, după care robotul răspunde la comenzile date din aplicația Blynk, transmitând constant schimbările de temperatură din încăpere.

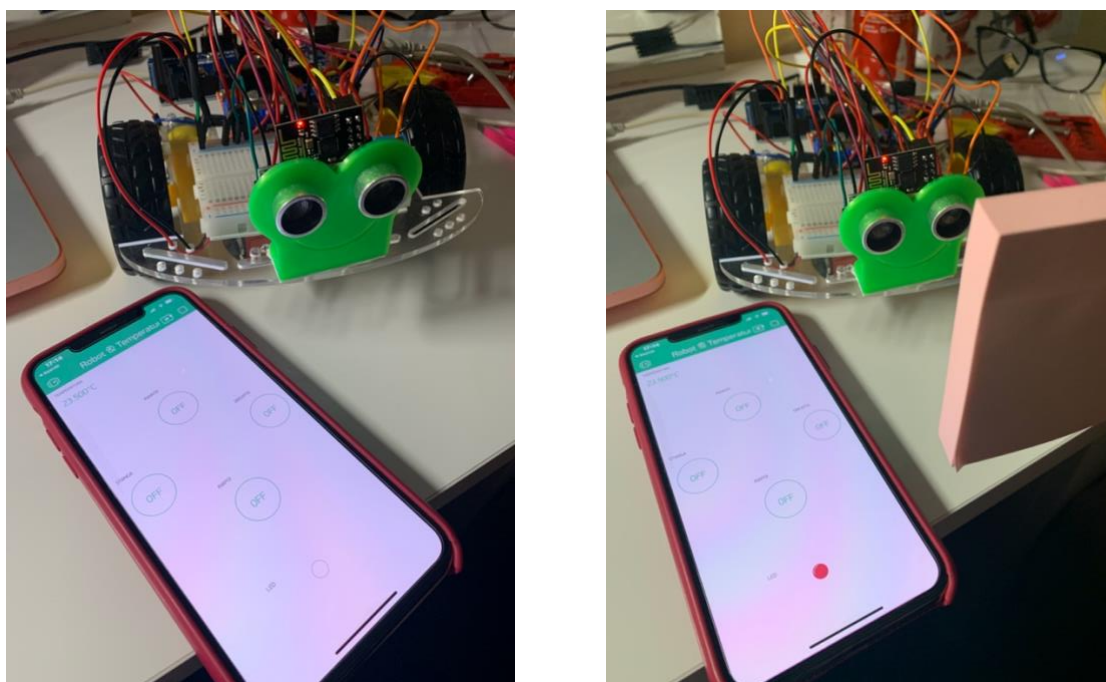


Fig 2.13 Testarea aplicației Blynk

- **Senzor de temperatura si umiditate digital DHT11**

DIAGRAMA PINILOR:

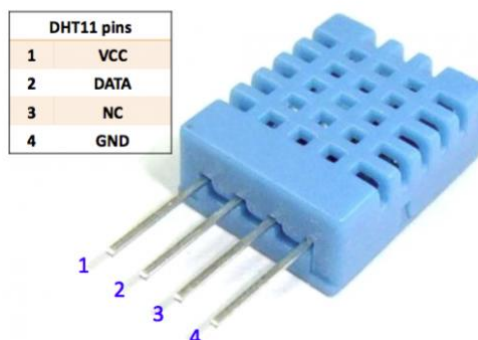


Fig 2.14 Senzor DHT11

Senzorul de temperatura si umiditate DHT11 incorporeaza un sensor de umiditate capacitive si un termistor, transmitand astfel temperature din jur printr-un semnal digital pe pinul de date. Acesta transmite data noi la fiecare 2 secunde.

Acesta are un rol insemnat in proiect, desi s-a implementat doar un sistem de monitorizare, acesta se poate extinde la un sistem mai avansat de control care poate fi conectat la centrala din casa si controlarea temperaturii astfel. De asemenea, poate fi folosit ca statie meteo in cazul in care se citesc temperaturi din exterior sau monitorizarea umiditatii din gradina.

Senzorul DHT11 are o acuratete de +/-5% in masurarea umiditatii si +/-2% in masurarea temperaturii.

Principiul de functionare consta in detectarea vaporilor de apa si masurarea rezistentei electrice dintre electrozi. Senzorul care simte umiditatea este un substrat de retinere a umezelii cu electrozi aplicati pe suprafata. Cand vaporii de apa sunt absorbiti de substrat, sunt eliberati ioni de substrat care cresc conductivitatea intre electrozi. In ceea ce priveste temperature, calculul se face cu un termistor(senzor de temperature NTC).

## 2.3 Asamblarea tuturor componentelor si testarea

Dupa asamblarea tuturor componentelor si realizarea conexiunilor corespunzatoare, am testat robotul si a fost functional.

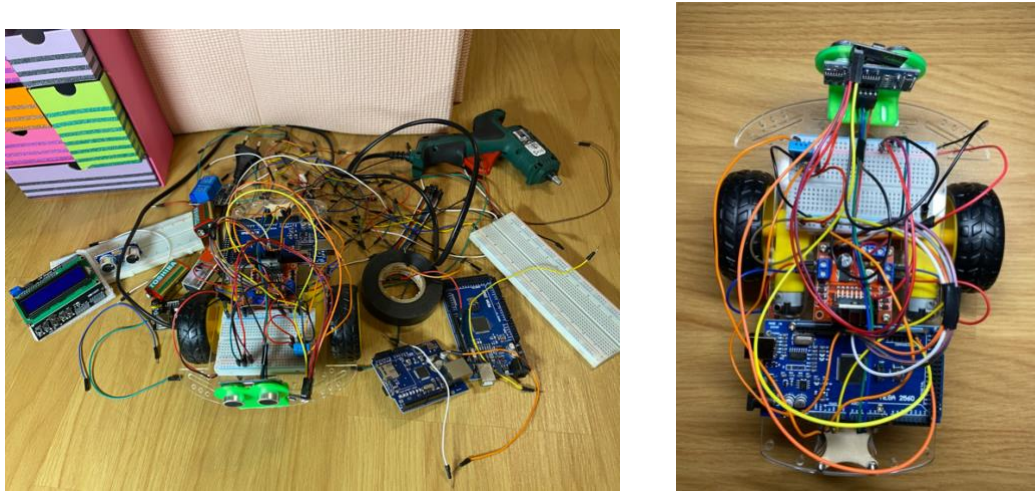


Fig 2.15 Imagini Robot asamblat

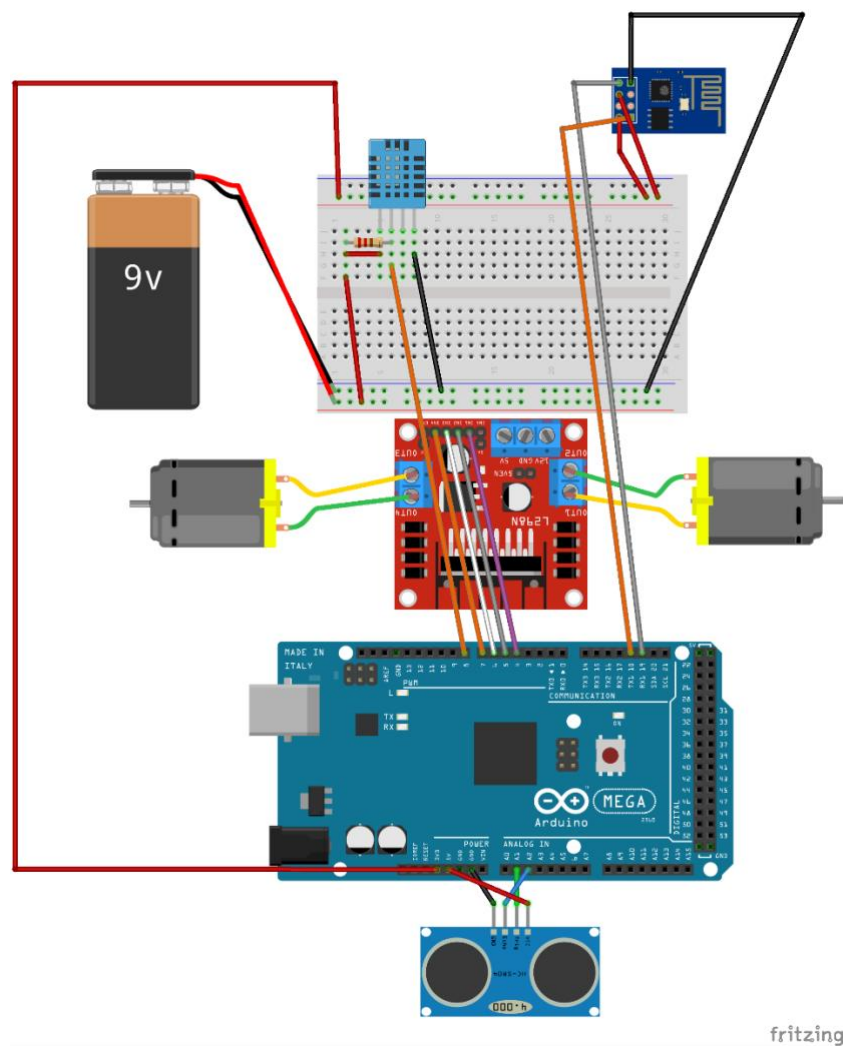


Fig 2.16 Schema de montaj

Pentru realizarea masinii de stare cu functionalitatea sistemului am ales sa folosim Stateflow, din cadrul Matlab. Astfel am hotarat ca sistemul are doua mari stari: starea OFF, cand nu e alimentat de la baterie sau alimentarea e mai mica de 9V si starea ON, cand robotul e functional, adica alimentat la 9V sau mai mult.

In starea OFF, ambele motoare sunt pe pozitia LOW, la fel si led-ul din aplicatie.

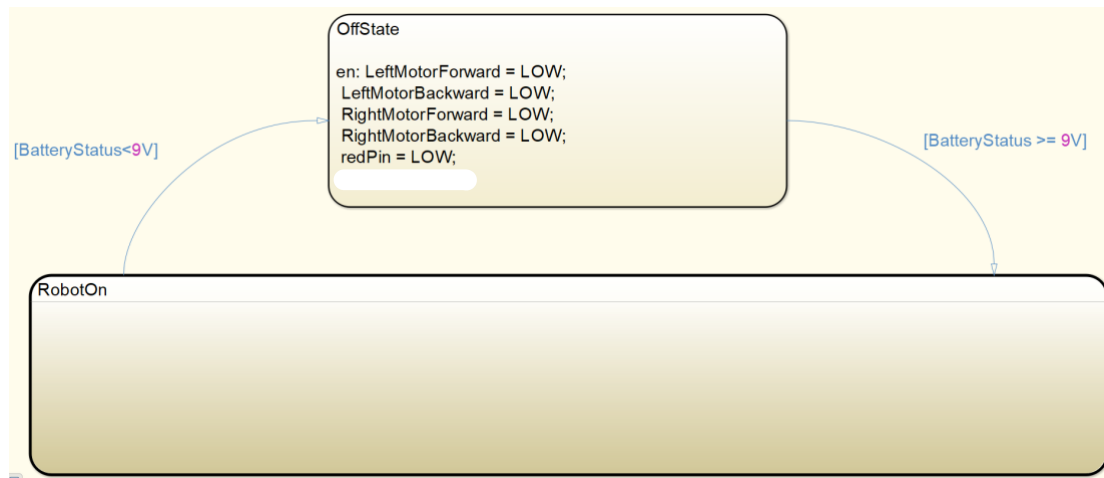


Fig 2.17 Starile principale ale robotului

In starea ON, se vor executa toate functionalitatile robotului.

In momentul in care sistemul intra in starea ON, inseamna ca avem, de asemenea, conexiunile la internet si la aplicatie deja stabilite.

Pe tot parcursul functionarii, senzorul ultrasonic va trimite unde si va calcula distante, cu ajutorul carora putem stabili daca robotul se afla intr-o situatie contraindicata deplasarii inainte. Daca distanta pana la obstacol e mai mica de 20 cm, atunci se va aprinde un led rosu in aplicatia mobile.

Avand un robot controlat printr-o aplicatie mobila, am implementat o functie if-else care verifica daca un buton este apasat si actioneaza ca atare: de exemplu daca e apasat butonul V0, care e corespunzator deplasarii inainte, atunci sistemul va merge inainte, iar daca niciunul dintre aceste butoane nu va fi apasat, robotul e oprit.

Am ales ca starile MoveStop, MoveForward, MoveBackward etc. sa aiba o decompozitie paralela(AND) deoarece ele descriu moduri operationale concurente; avem nevoie de toate starile sa fie active in acelasi timp, pentru a le putea folosi unde avem nevoie.



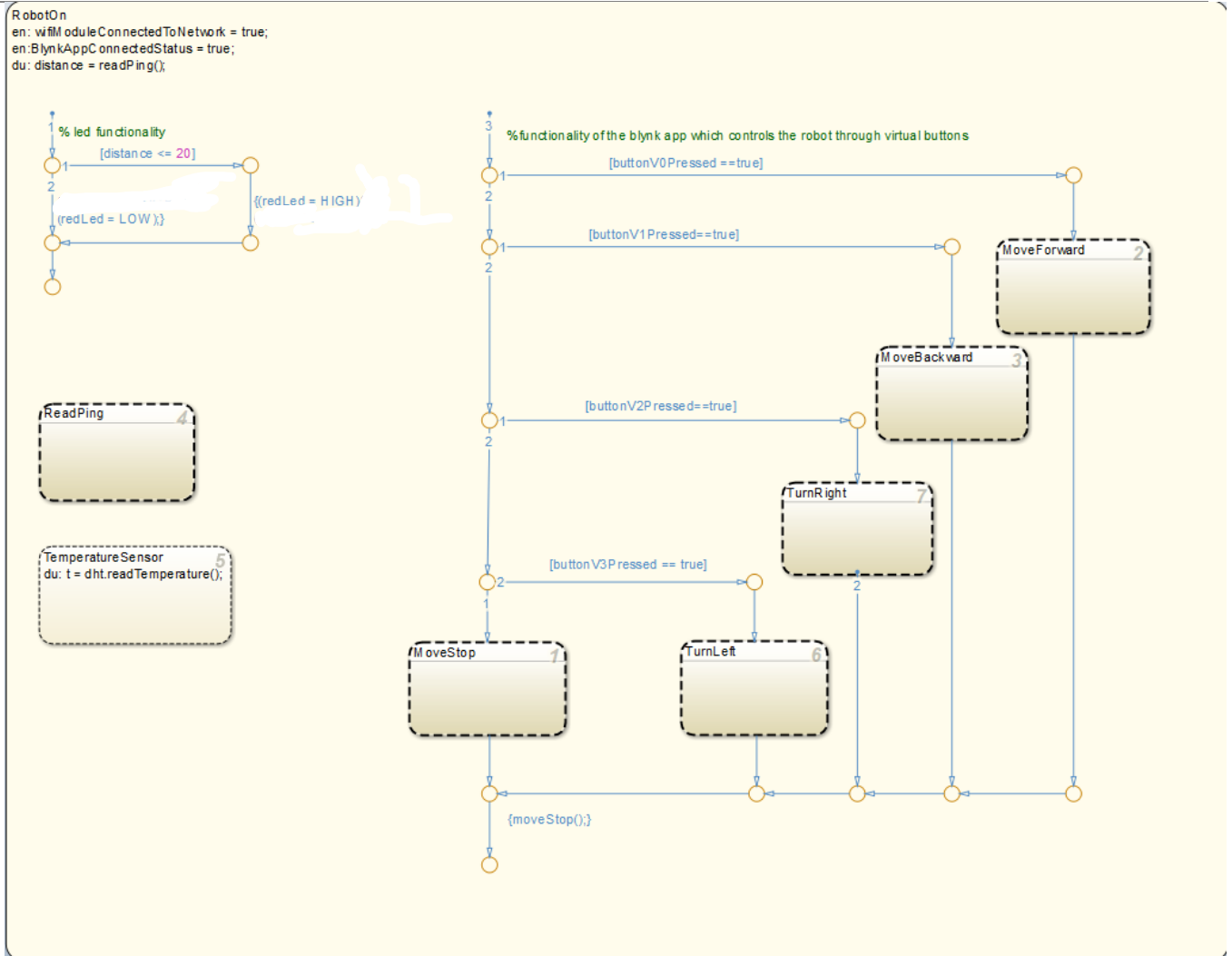


Fig 2.18 Modul de functionare al robotului

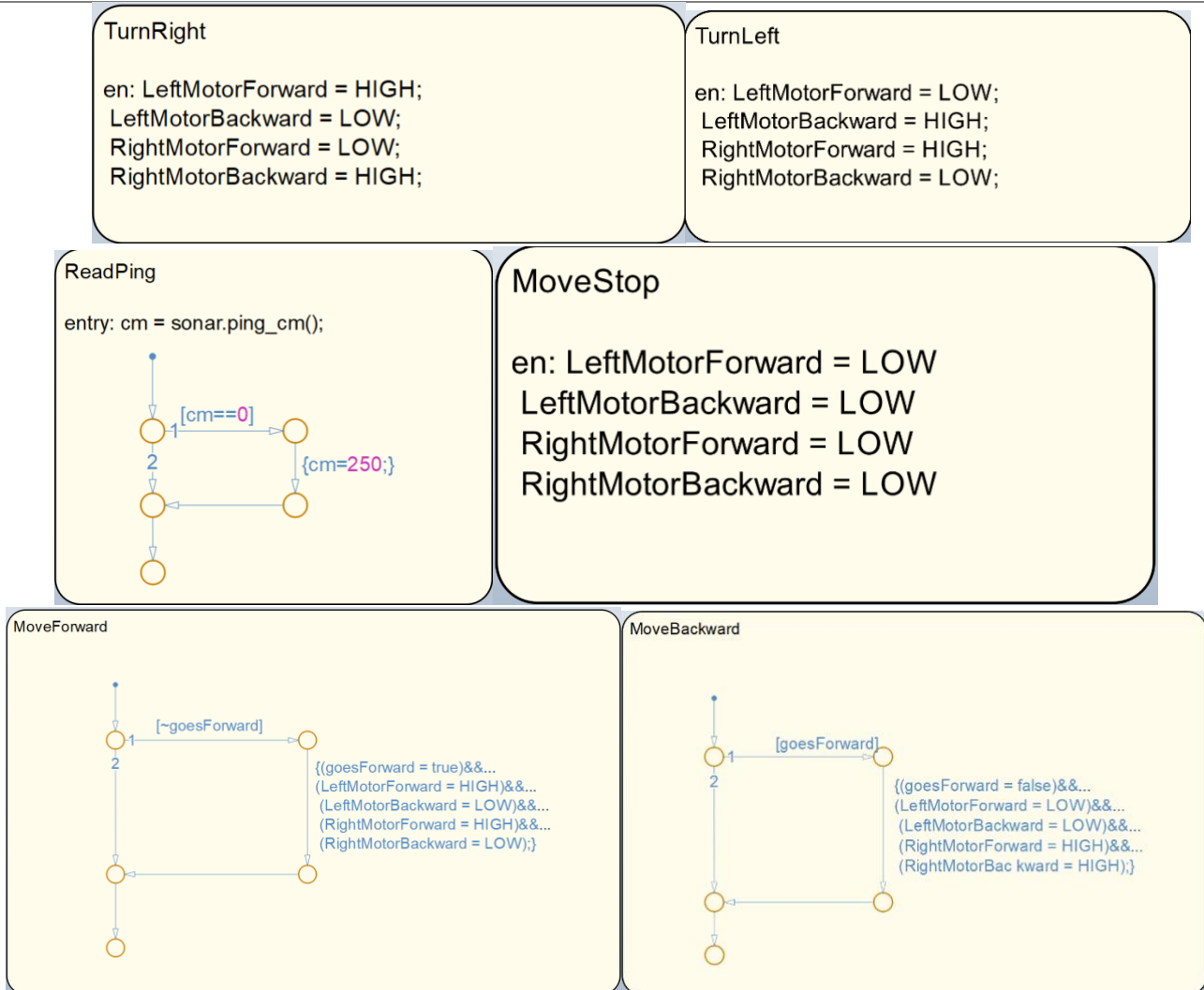


Fig 2.19 Starile active ale robotului

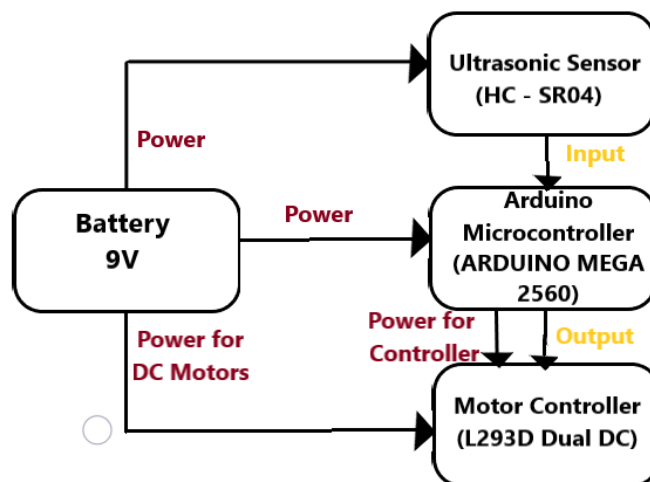


Fig 2.20 Diagrama de alimentare a componentelor



## 3 Concluzii

### 3.1 Rezultate obținute

Rezultatele obținute sunt cele dorite având în vedere costurile reduse ale componentelor și simplitatea programării având la dispoziție diferite librării și documentație vastă pentru mediul de programare Arduino și aplicația mobilă Blynk. Sistemul are funcționalitatea pe care am dorit-o: este controlat virtual prin intermediul conexiunii Wi-Fi cu ajutorul butoanelor de direcție din aplicația Blynk și, de asemenea, afișează în timp real temperatura de la locația în care se afla.

### 3.2 Direcții de dezvoltare

După cum am precizat anterior, conceptul IOT este din ce în ce mai popular și mai utilizat în viața de zi cu zi, de aceea un proiect ca acesta poate fi un punct de pornire pentru un sistem mult mai complex.

Pe lângă faptul că, a controla un sistem de la distanță este practic și eficient, transmiterea temperaturii și umidității în timp real poate servi ca bază pentru proiecte precum monitorizarea umidității într-o sere sau într-o grădina unde se cresc plante care necesită supraveghere în acest sens.

O altă aplicație ar putea consta în trimiterea robotului în diferite părți ale casei pentru a monitoriza temperatura, iar mai apoi setarea temperaturii în încăperi după preferințe prin conectare la centrala termică.

De asemenea, există posibilitatea de a trimite robotul în zone exterioare și de a servi ca stație meteo în anumite contexte.

Cert este faptul că IOT este în continuă dezvoltare și merita cunoașterea principiilor care stau la baza acestuia, având în vedere că suntem în continuă dezvoltare și constant înconjurați de o asemenea tehnologie.

## 4 Cod sursa

```
#include <NewPing.h>           //librarie pentru senzor
ultrasonic                    //librarie pentru
#define BLYNK_PRINT Serial    //librarie pentru
aplicatia mobila
#include <ESP8266_Lib.h>       //librarie pentru modul
wifi
#include <BlynkSimpleShieldEsp8266.h>
#include <DHT.h>               //librarie pentru senzor
de temperatura
#define EspSerial Serial1     //port serial1 pentru a
putea conecta TX,RX la 18 si 19
#define ESP8266_BAUD 115200    //setare baudrate specific
modulului nostru wifi
ESP8266 wifi(&EspSerial);     //conectarea placii la modul
#define DHTPIN 8
#define DHTTYPE DHT11         // DHT 11
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

WidgetLED led4(V4);           // pin
virtual Led rosu din aplicatie

char auth[] = "Uz1OqYxjzRxXjbwM-AMZ_7ilbVJakY8c";
//token de autorizare
char ssid[] = "Ispas";
//denumire wifi
char pass[] = "08728833";
//parola wifi

const int LeftMotorForward = 6; //pin miscare robot
stanga-inainte
const int LeftMotorBackward = 7; //pin miscare robot
stanga-inaoi
const int RightMotorForward = 5; //pin miscare robot
dreapta-inainte
const int RightMotorBackward = 4; //pin miscare robot
dreapta-inaoi

//conectarea senzorului ultrasonic la pinii analogici
#define trig_pin A1 //pin trig
#define echo_pin A2 //pin echo

#define maximum_distance 200 //initializare distanta
maxima la care poate detecta senzorul obiecte
boolean goesForward = false;
int distance = 100;           //initializare distanta
NewPing sonar(trig_pin, echo_pin, maximum_distance);
//functionarea senzorului folosind sonar

void sendSensor()             //trimiterea temperaturii pe tot
parcursul functionarii
{
    float t = dht.readTemperature(); // temp in grade
    celsius
    Serial.println("Temp: ");
    Serial.println(t);
    if (isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    Blynk.virtualWrite(V5, t); //pe pin virtual v5 trimitem
    temperatura
}

int readPing(){ //citire distanta in cm
    delay(70);
    int cm = sonar.ping_cm();
    if (cm==0){
        cm=250;
    }

    return cm;
}

void sendLed() {              //semnalizare led rosu pentru
distanza <=20
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    if(distance<=20){
        led4.on();
    }
    else {
        led4.off();
    }
}

void setup()
{
    Serial.begin(9600);
    EspSerial.begin(ESP8266_BAUD);
    delay(10);

    Blynk.begin(auth, wifi, ssid, pass);
    dht.begin();
    timer.setInterval(1000L, sendSensor);

    pinMode(RightMotorForward, OUTPUT);
    pinMode(LeftMotorForward, OUTPUT);
    pinMode(LeftMotorBackward, OUTPUT);
    pinMode(RightMotorBackward, OUTPUT);
}

void loop()
{
    Blynk.run(); //pornire aplicare
    timer.run();
}

void moveStop() { //oprire robot

    digitalWrite(RightMotorForward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorBackward, LOW);
}

void moveForward() { //miscare inainte

    if (!goesForward) {

        sendLed();
        sendSensor();

        goesForward = true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}

void moveBackward() { //miscare inapoi

    goesForward = false;

    digitalWrite(LeftMotorBackward, HIGH);
```

```
digitalWrite(RightMotorBackward, HIGH);
}

digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorForward, LOW);

void turnRight() {          //intoarcere la dreapta

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(500);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft() {          //intoarcere la stanga

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(500);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
}

//atribuirea fiecarui buton din aplicatie la miscarea
corespunzatoare
BLYNK_WRITE(V0)
{
    if (param[0]){
        moveForward();
    }
    else{
        moveStop();
    }
}

BLYNK_WRITE(V1)
{
    if (param[0]){
        moveBackward();
    }
    else{
        moveStop();
    }
}

BLYNK_WRITE(V2)
{
    if (param[0]){
        turnRight();
    }
    else{
        moveStop();
    }
}

BLYNK_WRITE(V3)
{
    if (param[0]){
        turnLeft();
    }
    else{
        moveStop();
    }
}
```

## 5 Bibliografie

- [1] <https://www.electronicshub.org/arduino-project-ideas/>
- [2] <https://www.arduino.cc/reference/en/language/functions/communication/serial/>
- [3] <https://blynk.io/en/getting-started/>
- [4] <https://howtomechatronics.com/>
- [5] <https://www.tutorialspoint.com/index.htm>  
<https://www.tutorialspoint.com/index.htm>