**DOCUMENTATIE ELECTRONICA APLICATA IN ROBOTICA**

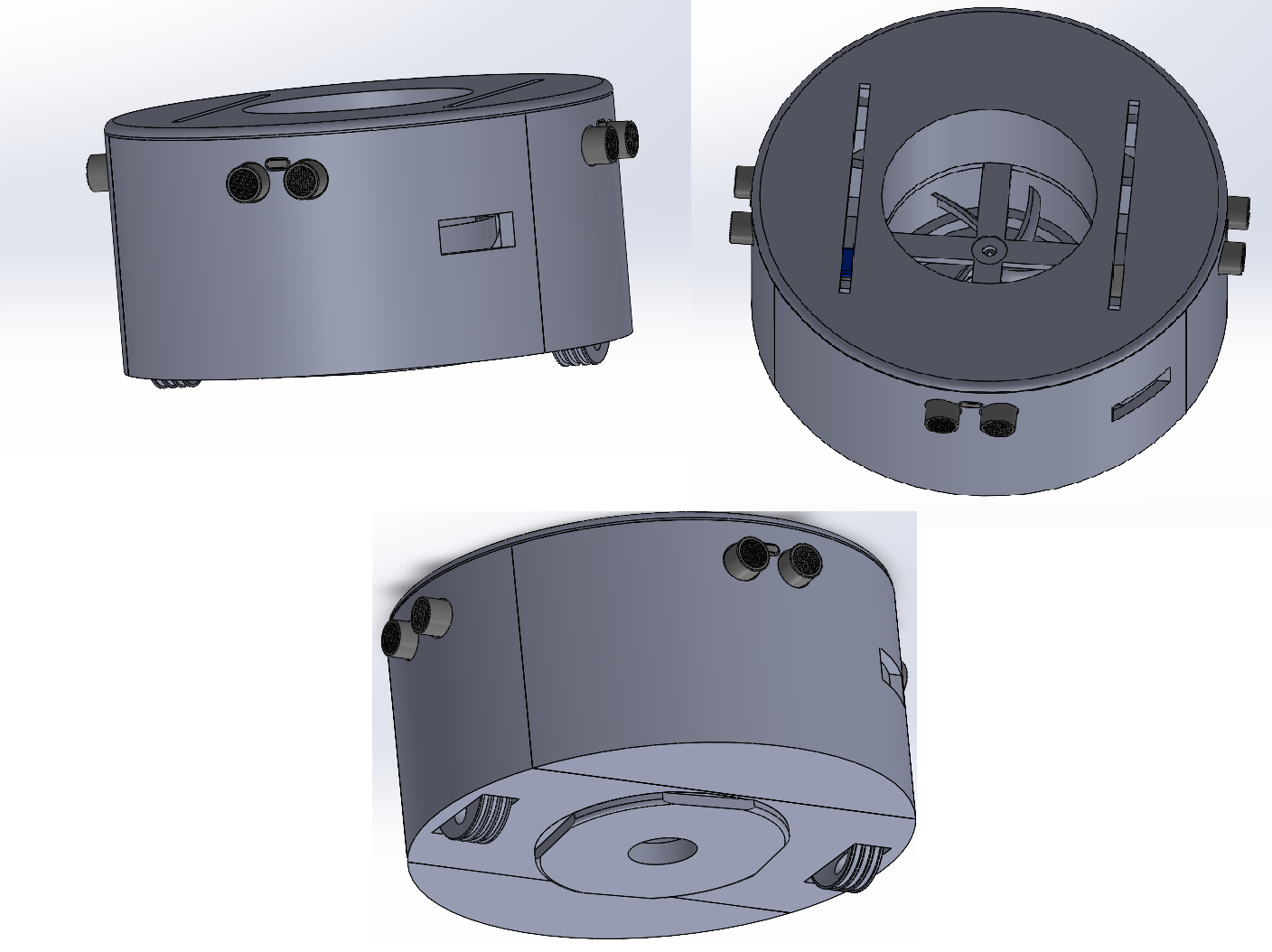
**-PROIECT-**

**-ROBOT ASPIRATOR-**

Dascal Darius

Einholcz Alexandru-Stefan

Carpinisan Andrei



**Conceptul initial**

**Lista cu componentele electronice utilizate in structura robotului:**

-Arduino UNO R3;

-Breadboard;

-3 senzori ultrasonici HG-SR04;

-2 motoare Stepper DC 28BYJ-48 + drivere ULN2003

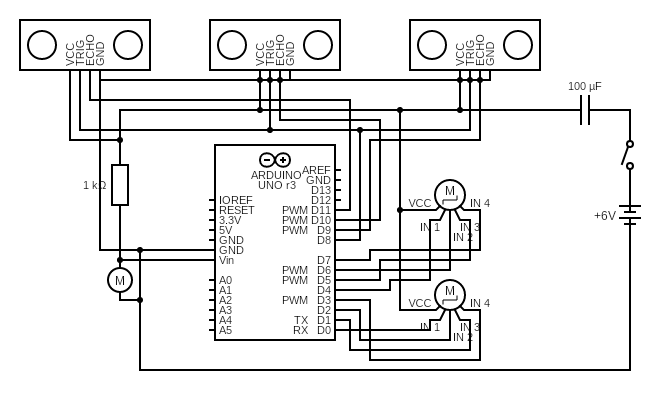
-1 motor DC 5v;

-rezistenta 1kΩ

-condensator 100μF

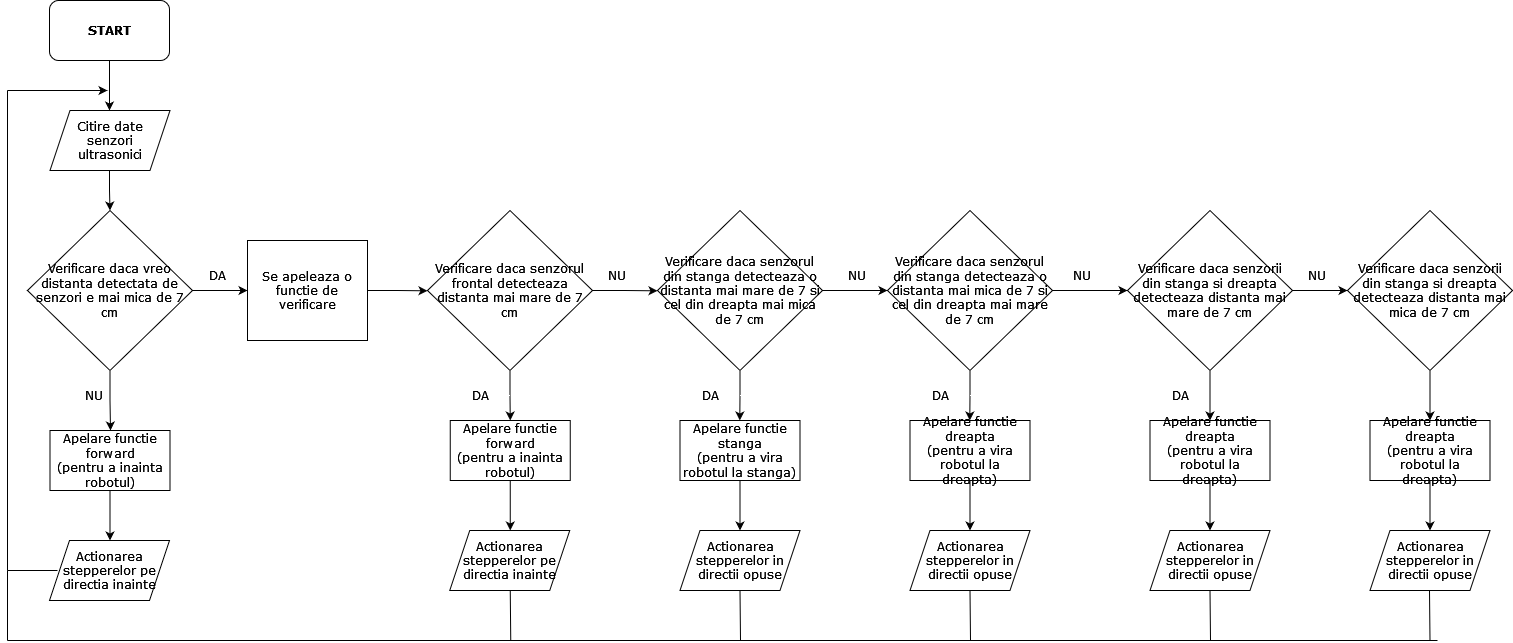
-1 switch pentru comutarea ON/OFF

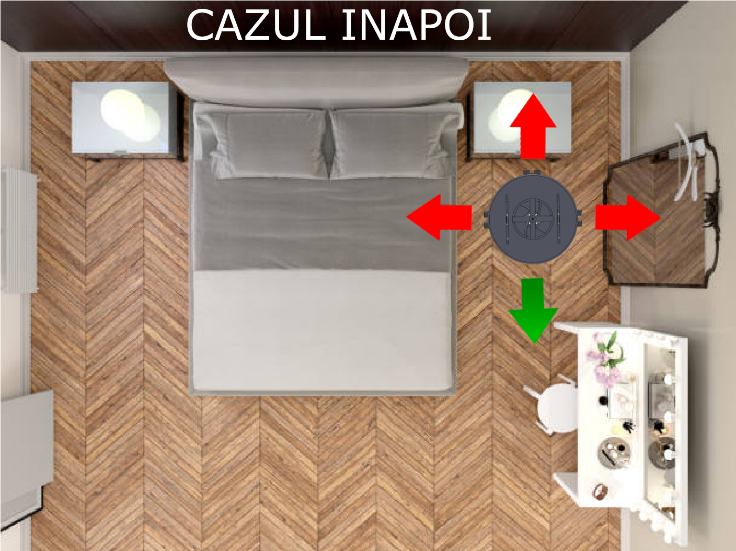
-battery pack

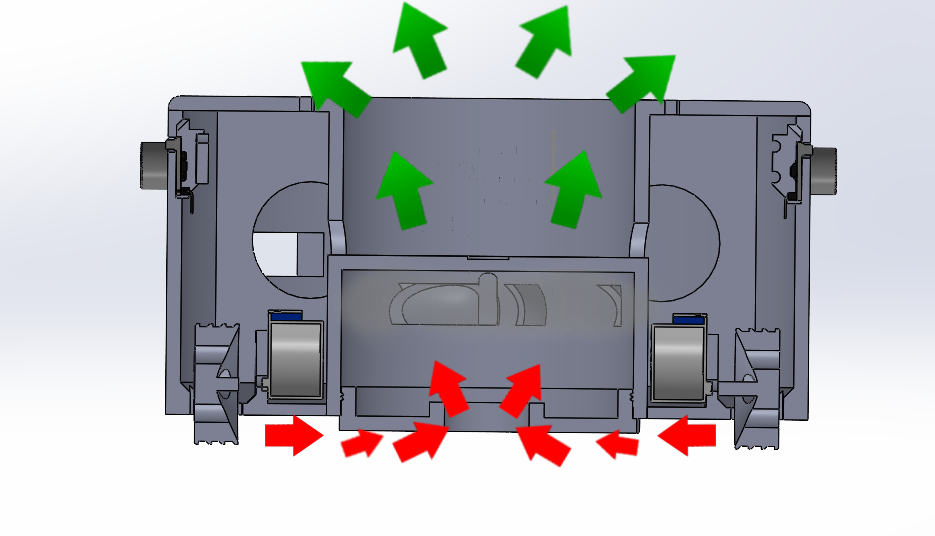
 **Schema electrica a circuitului:**

**Modul de functionare:**

Algoritmul pe care se bazeaza robotul poate fi ilustrat in urmatorul flow-chart si este explicat in ilustratiile de mai jos:







**Date importante despre senzori si steppere:**

Senzor ultrasonic HC-SR04

* Alimentare:+5V DC
* Curent in sarcina: 15mA
* Effectual Angle: <15°
* Distante detectabile de senzor: 2cm – 400 cm
* Unghi de functionare: 30 degree
* Dimensiuni: 45mm x 20mm x 15mm

28BYJ-48 – 5V Stepper

* Frecventa: 100Hz
* Rezistenta: 50Ω±7%(25°C)
* Alimentare: +5V DC
* Stepper unipolar
* Se foloseste in tandem cu driverul ULN2003
* Gear ratio: 64:1
* Noise <35dB(120Hz,No load,10cm)

**Codul:**

**#include <NewPing.h>**

**#include <AccelStepper.h>**

**#define SONAR\_NUM 3**

**#define MAX\_DISTANCE 200**

**NewPing sonar[SONAR\_NUM] = {**

**NewPing(8, 9, MAX\_DISTANCE),**

**NewPing(8, 10, MAX\_DISTANCE),**

**NewPing(8, 11, MAX\_DISTANCE)**

**};**

**AccelStepper myStepper(8, 0, 1, 2, 3);**

**AccelStepper myStepper2(8, 4, 5, 6, 7);**

**void setup() {**

**myStepper.setMaxSpeed(1000);**

**myStepper2.setMaxSpeed(1000);**

**}**

**void forward()**

**{**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=-1000 && myStepper2.currentPosition()!=-1000)**

**{myStepper.setSpeed(-500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(-500);**

**myStepper2.runSpeed();**

**}**

**}**

**void backward()**

**{**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=2000 && myStepper2.currentPosition()!=2000)**

**{myStepper.setSpeed(500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(500);**

**myStepper2.runSpeed();**

**right();**

**}**

**}**

**void left()**

**{**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=-4500 && myStepper2.currentPosition()!=4500)**

**{myStepper.setSpeed(-500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(500);**

**myStepper2.runSpeed();**

**}**

**forward();**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=-4500 && myStepper2.currentPosition()!=4500)**

**{myStepper.setSpeed(-500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(500);**

**myStepper2.runSpeed();**

**}**

**}**

**void right()**

**{**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=4500 && myStepper2.currentPosition()!=-4500)**

**{myStepper.setSpeed(500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(-500);**

**myStepper2.runSpeed();**

**}**

**forward();**

**myStepper.setCurrentPosition(0);**

**myStepper2.setCurrentPosition(0);**

**while(myStepper.currentPosition()!=4500 && myStepper2.currentPosition()!=-4500)**

**{myStepper.setSpeed(500);**

**myStepper.runSpeed();**

**myStepper2.setSpeed(-500);**

**myStepper2.runSpeed();**

**}**

**}**

**/\***

**sonar[0]=dreapta**

**sonar[1]=fata**

**sonar[2]=stanga**

**\*/**

**void verificare()**

**{**

**if(sonar[0].convert\_cm(sonar[0].ping\_median()) && sonar[1].convert\_cm(sonar[1].ping\_median()) && sonar[2].convert\_cm(sonar[2].ping\_median()))**

**if(sonar[1].convert\_cm(sonar[1].ping\_median())>7.)**

**forward();**

**else**

**{**

**if(sonar[0].convert\_cm(sonar[0].ping\_median())<7. && sonar[2].convert\_cm(sonar[2].ping\_median())<7.)**

**backward();**

**else**

**{if(sonar[0].convert\_cm(sonar[0].ping\_median())<7. && sonar[2].convert\_cm(sonar[2].ping\_median())>7.)**

**left();**

**else{**

**if(sonar[0].convert\_cm(sonar[0].ping\_median())>7. && sonar[2].convert\_cm(sonar[2].ping\_median())<7.)**

**right();**

**else**

**{if(sonar[0].convert\_cm(sonar[0].ping\_median())>7. && sonar[2].convert\_cm(sonar[2].ping\_median())>7.)**

**right();**

**}**

**}**

**}**

**}**

**}**

**/\***

**verificare:**

**0 - forward**

**1 - backward**

**2 - right**

**3 - left**

**4 - com**

**\*/**

**void loop() {**

**delay(100);**

**bool ok=true;**

**for(uint8\_t i=0;i<SONAR\_NUM && ok;i++)**

**{**

**if(sonar[i].convert\_cm(sonar[i].ping\_median())!=0.)**

**if(sonar[i].convert\_cm(sonar[i].ping\_median())<7.)**

**ok=false;**

**}**

**if(ok)**

**{**

**forward();**

**}**

**else**

**{**

**verificare();**

**}**

**}**