



Documentatie Proiect

Proiectarea cu Micro-Procesoare

Halasz Lorand Daniel
Grupa 30236

Cuprins

1. Prezentare generala	3
2. Aspecte Teoretice	5
3. Scheme de montaj si diagrame de proiectare	6
4. Mod de functionare	9
5. Codul sursa.....	10
6. Referinte	18

1. Prezentare generala

Acest proiect, realizat in mediul de dezvoltare Arduino, consta in proiectarea si implementarea unui robot experimental, care are urmatoarele functionalitati:

- Implementarea unui control precis al motoarelor pentru a asigura o deplasare in linie dreapta
- Detectia si ocolirea obstacolelor

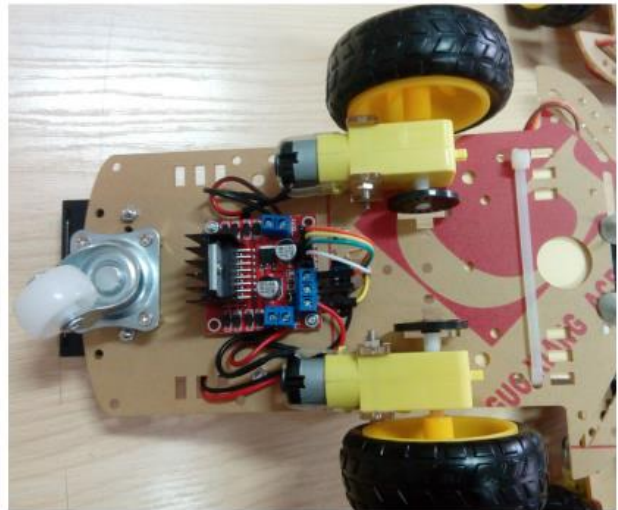
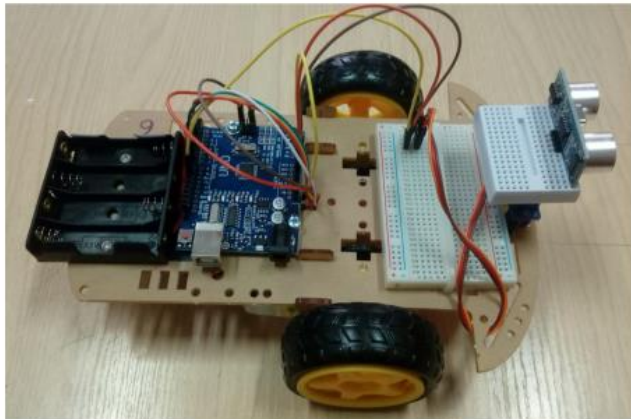
Prima functionalitatea consta in masurarea turatiei cu ajutorul a doi senzori de intrerupere infrarosu. Acesta este un senzor fotoelectric in forma de U, care are doua componente: led-ul care emite intrarosu si fototranzistorul. Senzorul este util pentru a detecta daca un obiect se afla intre cele doua brate si opreste transferul radiatiei IR, astfel ca acesta este folosit pentru a calcula un numar de rotatii pe minut la un motor, utilizand si o roata pentru encoder.

Pentru cea de-a doua functionalitate se foloseste un senzor de tip sonar pentru determinarea distantelor si un servo-motor, care are rolul de a misca modulul respectiv. Senzorul HC-SR04 emite ultrasunete la 40.000 Hz care se deplaseaza prin aer, iar daca intalneste un obiect sau un obstacol in calea sa, va reveni inapoi la modul. Avand in vedere timpul in care semnalul se intoare inapoi la senzor si viteza sunetului, cu ajutorul acestuia putem calcula distanta pana la obstacol.



Robotul experimental contine urmatoarele componente:

- Placa microcontroller compatibila Arduino Uno
- Driver motoare L298N Dual H-Bridge
- 2 Motoare DC
- Motor Servo
- Carcasa baterii 4xAA (R6)
- 2 roti conectate la motoare
- Roata suport
- Suport plexiglas
- Placa de prototipizare (Breadboard)
- Senzor sonar
- 2 senzori de intrerupere infrarosu



2. Aspecte Teoretice

Mai departe este prezentata modalitatea de determinare a distantei pana la obstacolul intalnit, care se calculeaza astfel:

Se foloseste formula: $v = \frac{d}{t}$, unde v = viteza sunetului

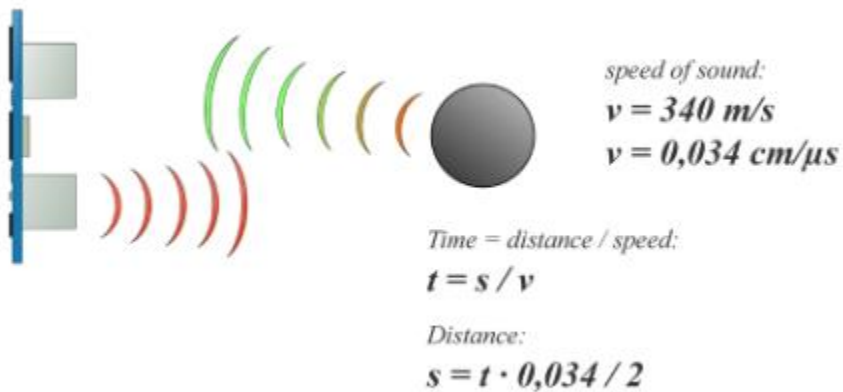
d = distanta pana la obstacol, care trebuie determinata

t = timpul de la emiterea ultrasunetelor pana in
momentul receptarii acestora

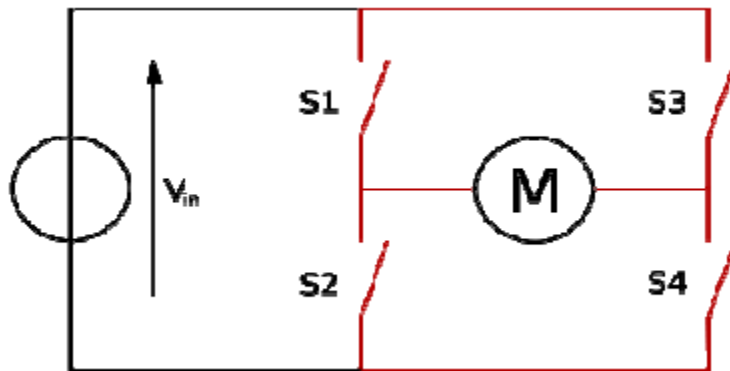
Viteza sunetului este de 340 m/s sau 0,034 cm/ μ s

In cazul de fata mai este nevoie sa impartim cu formula cu 2, deoarece sunetul parcurge aceasta distanta de doua ori, cand este emis si in momentul in care acesta se intoarce. Astfel, formula pentru determinarea distantei pana la obstacolul intalnit este:

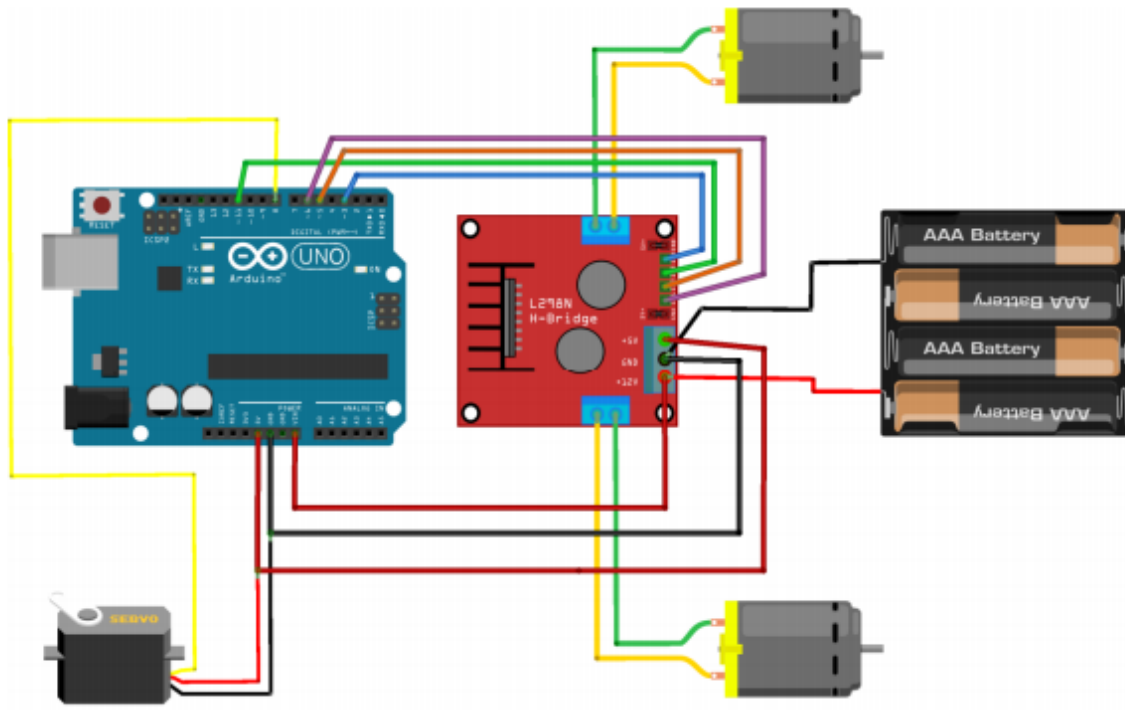
$$d = \frac{0.034 \cdot t}{2}$$



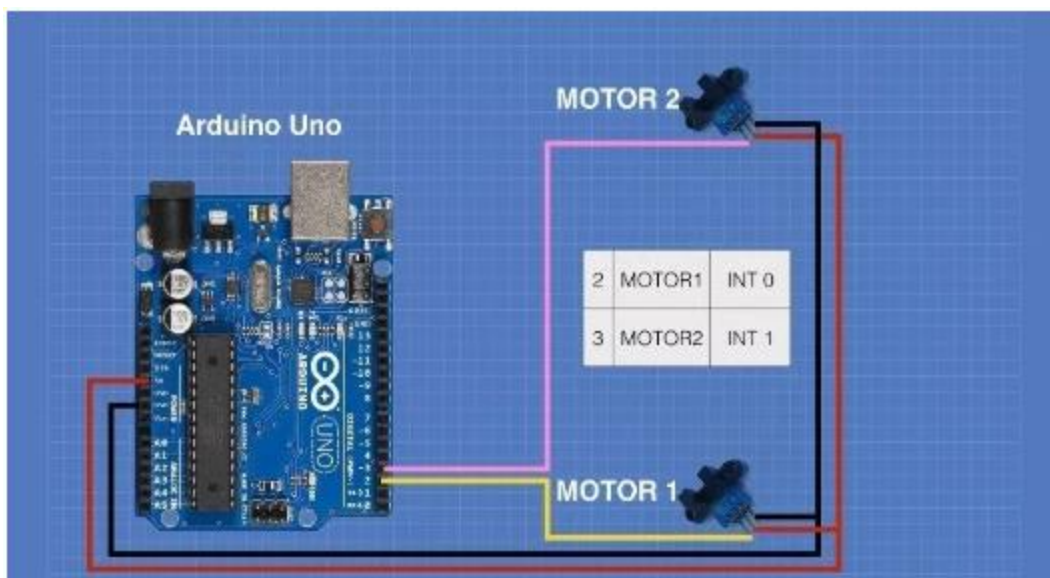
3. Scheme de montaj si diagrame de proiectare



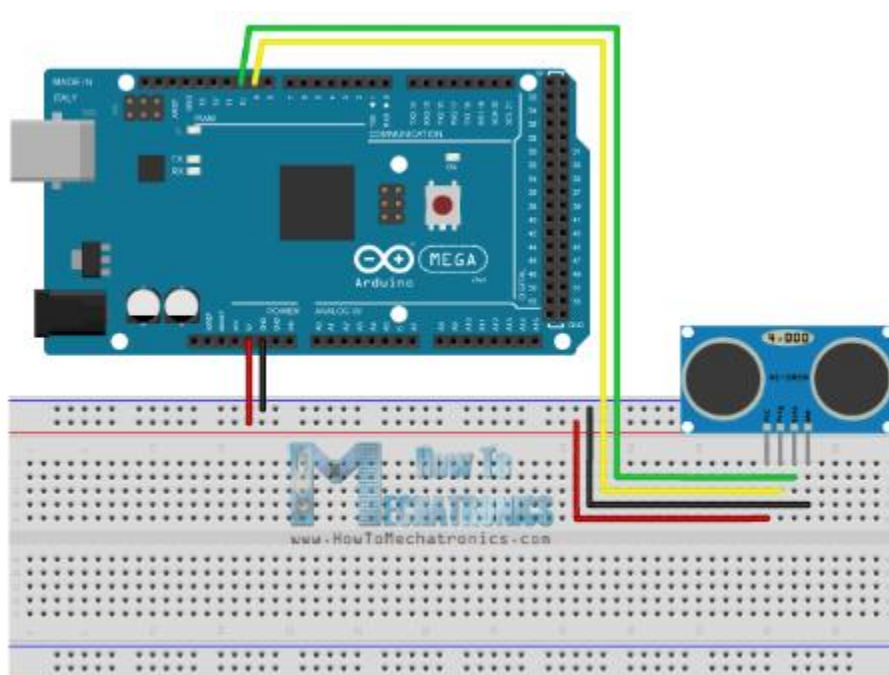
Punte H: S1-S4 sunt comutatoarele, iar M reprezinta motorul



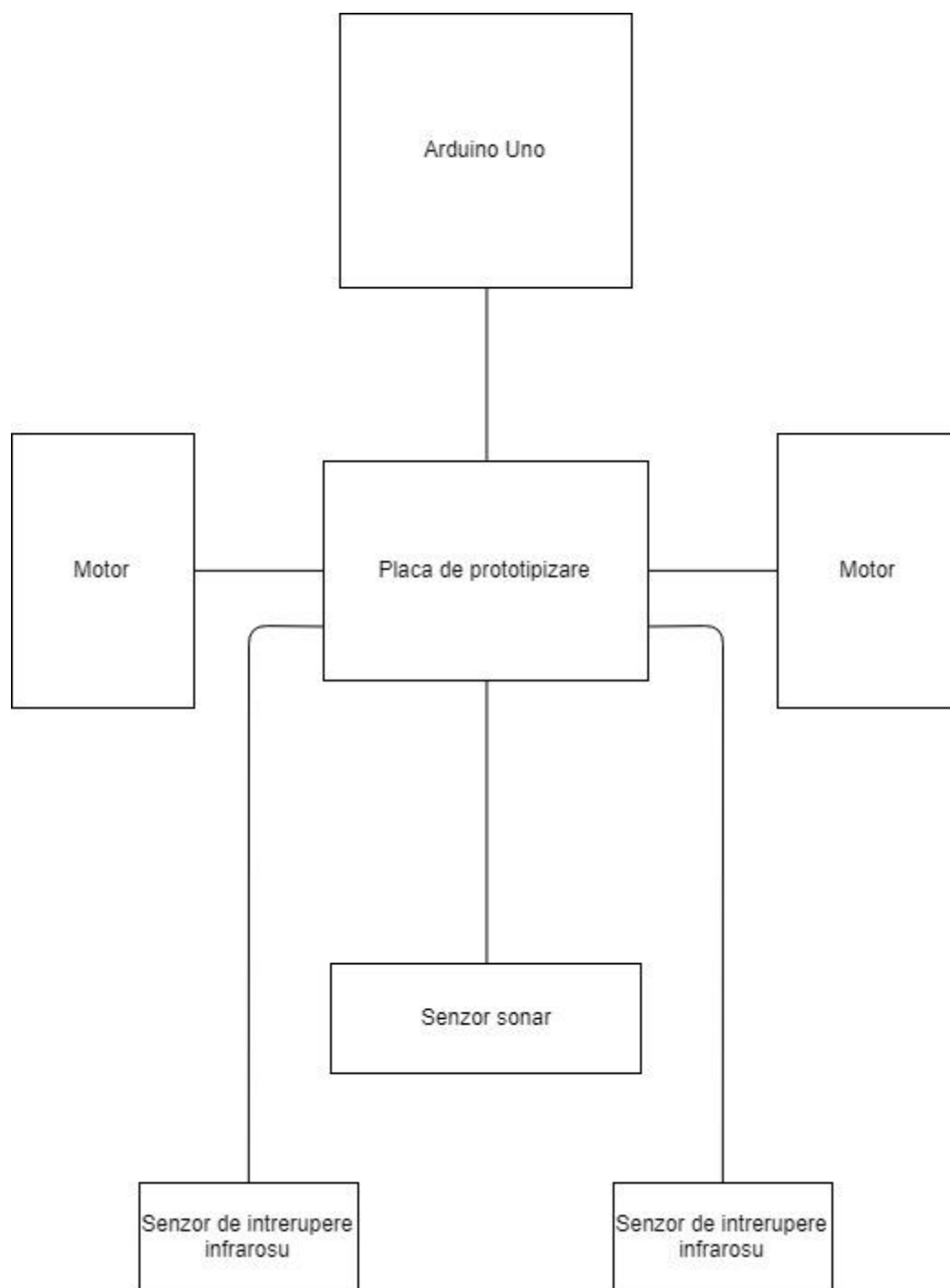
Schema electrica a robotului



Schema pentru conectarea celor doi senzori



Schema pentru conectarea sonarului



Schema bloc a proiectului

4. Mod de functionare

Avand in vedere faptul ca robotul are 2 moduri de functionare, acestea se vor realiza astfel:

In prima faza, dupa incarcarea codului sursa pe placuta si alimentarea robotului, va intra in functiune primul mod de functionare, si anume cel pentru implementarea unui control precis al motoarelor pentru a asigura o deplasare în linie dreaptă. Acesta se rezolva cu ajutorul functiei `attachInterrupt`, care are rolul de a atasa o functie ISR la o intrerupere externa, folosind orice functie precedenta care a fost atasata anterior, si de a activa intreruperea. Aceasta are 3 parametri: `interrupt`, `ISR` si `mode`. Primul parametru al functiei, reprezinta numarul intreruperii. Parametrul `interrupt` nu corespunde cu numarul intreruperii externe, astfel ca este necesara utilizarea functiei `digitalPinToInterrupt`, care va returna numarul intreruperii atasata pinului digital. Dupa calcularea celor 2 turatii, acestea sunt comparate, iar daca nu sunt egale, se creste viteza motorului cu turatia mai mica, pana in momentul in care va ajunge la aceeasi turatie cu a celuilalt motor.

Dupa reglarea celor doua turatii, va intra in functiune cel de-al doilea mod, de detectie si ocolire a obstacolelor. Acesta este realizat cu ajutorul servomotorului, care este folosit pentru a obtine rotatii partiale, stabile si controlate pentru efectuarea de operatii cu amplitudine mica, dar cu precizie ridicata, si a senzorului de tip sonar, care, initial verifica daca se gaseste un obstacol in fata la o distanta mai mica decat o distanta data, iar in cazul unui raspuns afirmativ, face acelasi lucru pentru partea stanga, respectiv pentru partea dreapta, dupa care isi continua deplasarea spre partea cu distanta maxima pana la un alt obstacol.

5. Codul sursa

Codul Arduino necesar implementarii proiectului:

```
#include <Servo.h>

// Pinii motor 1
#define mpin00 6
#define mpin01 5

// Pinii motor 2
#define mpin10 11
#define mpin11 10

// Pinii pentru senzor ultrasonic
#define trigPin 9
#define echoPin 12

Servo srv;

float value=0;
volatile int rev1=0, rev2=0;
float rpm1 = 0, rpm2 = 0;
int oldtime=0;
int times;
int speed1 = 70, speed2 = 70;
int ok = 0;

void isr1() //interrupt service routine
{
```

```

    rev1++;
}

void isr2() //interrupt service routine
{
    rev2++;
}

void setup()
{
    digitalWrite(mpin00, 0);
    digitalWrite(mpin01, 0);
    digitalWrite(mpin10, 0);
    digitalWrite(mpin11, 0);

    pinMode(2 ,INPUT);
    pinMode(3, INPUT);

    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);

    pinMode (mpin00, OUTPUT);
    pinMode (mpin01, OUTPUT);
    pinMode (mpin10, OUTPUT);
    pinMode (mpin11, OUTPUT);

    Serial.begin(9600);

    attachInterrupt(digitalPinToInterrupt(2), isr1, RISING); //attaching the interrupt

```

```
attachInterrupt(digitalPinToInterrupt(3), isr2, RISING); //attaching the interrupt
```

```
while((ok == 0 || ((int)(rpm1 * 100) != (int)(rpm2 * 100)))){
```

```
    ok = 1;
```

```
    StartMotor(mpin00, mpin01, 1, speed1);
```

```
    StartMotor(mpin10, mpin11, 1, speed2);
```

```
    calcRpm();
```

```
    if(rpm1 > rpm2)
```

```
        speed2 += 10;
```

```
    else
```

```
        speed1 += 10;
```

```
    Serial.print("Speed1 ");
```

```
    Serial.println(speed1);
```

```
    Serial.print("Speed2 ");
```

```
    Serial.println(speed2);
```

```
    Serial.println();
```

```
}
```

```
// sensor ultrasonic
```

```
pinMode(trigPin, OUTPUT);
```

```
pinMode(echoPin, INPUT);
```

```
}
```

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

```
{
```

```

if (speed==0)
{
    digitalWrite(m1, 0);
    digitalWrite(m2, 0);
}
else
{
    if (forward)
    {
        digitalWrite(m2, 0);
        analogWrite (m1, speed);
    }
    else
    {
        digitalWrite(m1, 0);
        analogWrite(m2, speed);
    }
}

}

void calcRpm(){
    delay(1000);

    detachInterrupt(digitalPinToInterrupt(2));    //detaches the interrupt
    detachInterrupt(digitalPinToInterrupt(3));    //detaches the interrupt

    times = millis() - oldtime;    //finds the time
    rpm1 = (float)rev1 / times;    //calculates rpm

```

```
rpm2 = (float)rev2 / times;    //calculates rpm  
oldtime = millis();           //saves the current time
```

```
Serial.print("rpm1 ");  
Serial.println(rpm1);  
Serial.print("rpm2 ");  
Serial.println(rpm2);  
Serial.println();
```

```
rev1 = 0;  
rev2 = 0;
```

```
attachInterrupt(digitalPinToInterrupt(2), isr1, RISING);  
attachInterrupt(digitalPinToInterrupt(3), isr2, RISING);  
}
```

```
void loop()  
{  
  StartMotor(mpin00, mpin01, 1, speed1);  
  StartMotor(mpin10, mpin11, 1, speed2);  
  
  delay(1000);  
  
  playWithServo(13);  
  
  StartMotor(mpin00, mpin01, 1, 0);  
  StartMotor(mpin10, mpin11, 1, 0);
```

```

    delay(500);
}

void playWithServo(int pin)
{
    double distFront, distLeft, distRight;
    srv.attach(pin);
    srv.write(90);
    distFront = calcDist();

    Serial.println();
    Serial.println(distFront);
    Serial.println();
    if(distFront < 30)
    {
        StartMotor(mpin00, mpin01, 1, 0);
        StartMotor(mpin10, mpin11, 1, 0);
        srv.write(0);
        delay(1000);
        distRight = calcDist();
        srv.write(180);
        delay(1000);

        distLeft = calcDist();
        srv.write(90);
        delay(1000);
    }
}

```

```

Serial.print("distLeft ");
Serial.println(distLeft);
Serial.print("distRight ");
Serial.println(distRight);
if(distLeft > distRight )
{
    StartMotor(mpin00, mpin01, 0, 0);
    StartMotor(mpin10, mpin11, 1, speed2);

    delay(1000);
    delayStopped(200);
}
else
{

StartMotor(mpin00, mpin01, 1, speed1);
    StartMotor(mpin10, mpin11, 0, 0);
    delay(1000);
    delayStopped(200);
}

}
srv.detach();
}

double calcDist() {

```



```

// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
double duration = pulseIn(echoPin, HIGH);

// Calculating the distance
double distance = duration * 0.034 / 2;

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
return distance;
}

void delayStopped(int ms)
{
  StartMotor (mpin00, mpin01, 0, 0);
  StartMotor (mpin10, mpin11, 0, 0);
  delay(ms);
}

```

6. Referinte

<http://users.utcluj.ro/~negrum/src/html/dmp.html>

<https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

<https://dronebotworkshop.com/robot-car-with-speed-sensors/>