

Line Follower and Obstacle Avoiding Car

Introducere

Prezentarea pe scurt a proiectului meu:

- Proiectul consta intr-un robot de tip line-follower care poate sa ocoleasca anumite obstacole aparute in calea sa. Acesta urmareste un traseu bine definit, iar in cazul in care se intalneste cu un obstacol, il va ocoli, intorcandu-se inapoi pe circuit.
- Scopul proiectului este de a fi folosit ca prototip pentru masini si roboti utili apoi in industrii precum agricultura sau constructii.
- Ideea de la care am pornit are la baza un robot de explorare. Mai in detaliu, un robot care poate urmări o linie trasată pe teren și în același timp să evite obstacolele, cum ar fi roci sau arbori. Acest tip de robot ar putea fi utilizat în cercetarea terenurilor greu accesibile sau pentru a colecta date despre mediul inconjurator

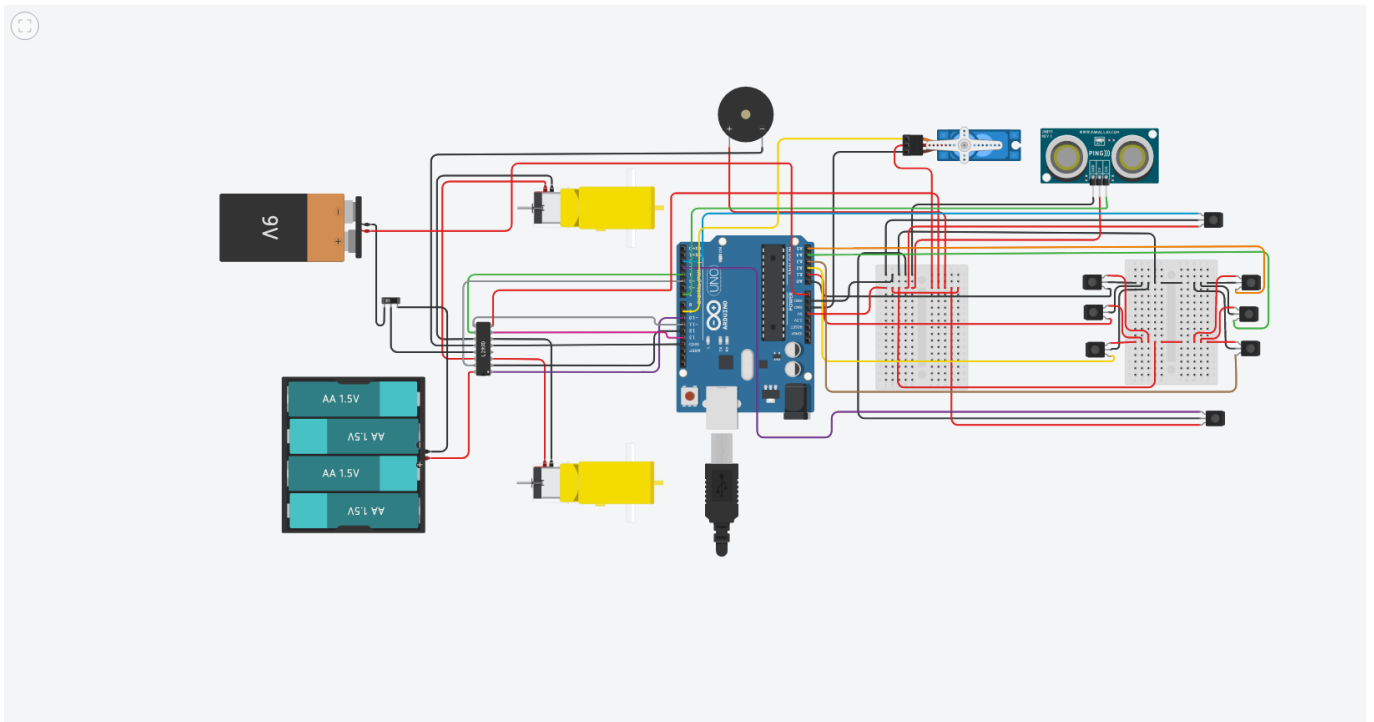
Descriere generala

Un proiect de tipul line follower and obstacle avoiding implică construirea unui robot care poate urmări o linie trasată pe o suprafață plană, cum ar fi o bandă neagră sau o linie albă pe un fundal negru, și în același timp poate evita obstacolele care apar în cale.

Pentru a realiza acest proiect, voi avea nevoie de un ansamblu de piese, printre care se remarcă: un microcontroller, senzori infraroșu sau senzori de proximitate pentru a detecta linia și obstacolele, motoare pentru a propulsa robotul și un algoritm de control care să permită robotului să urmeze linia și să evite obstacolele.



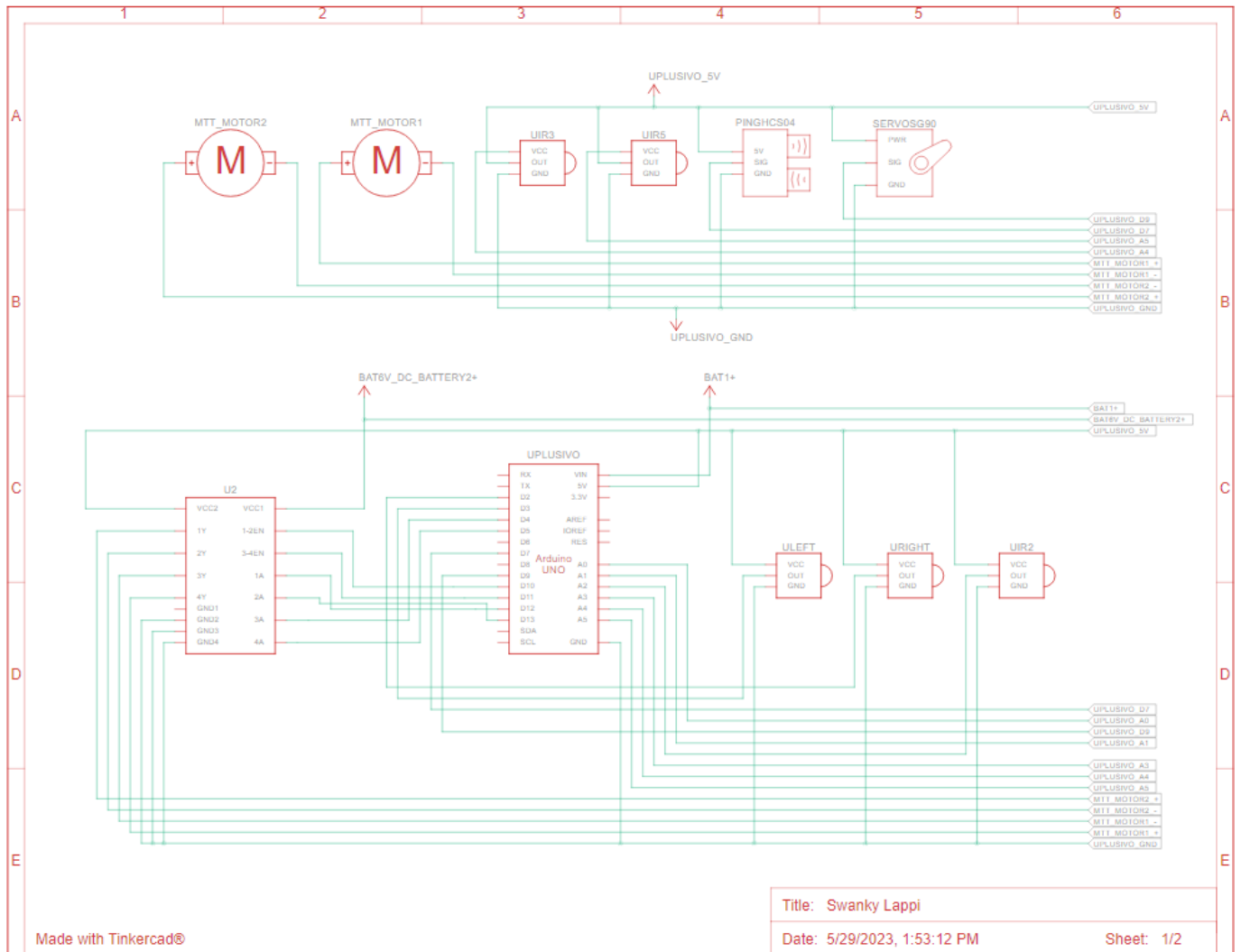
Hardware Design

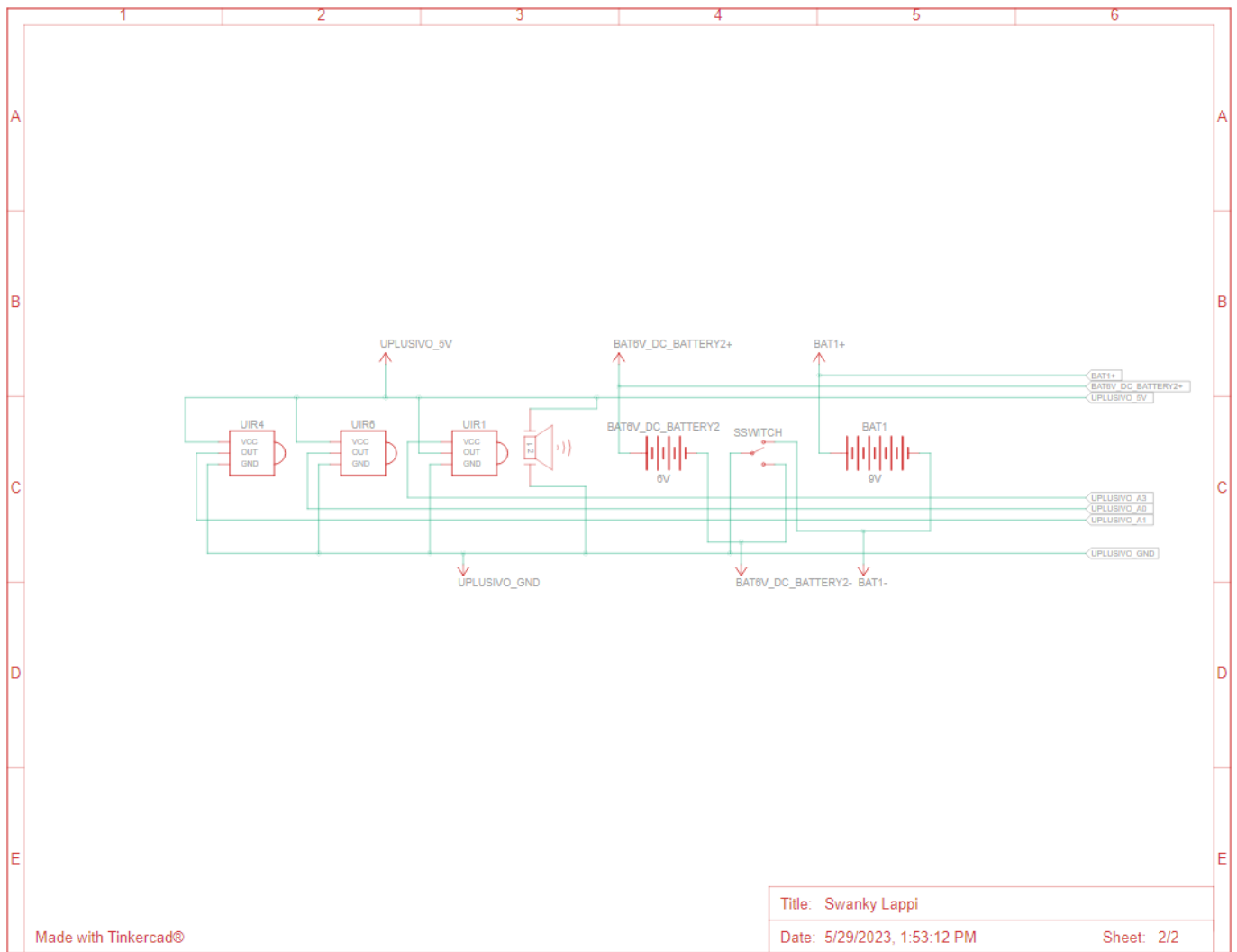


Componente Hardware utilizate:

- Placa de dezvoltare compatibila cu Arduino UNO R3
- Doua roti
- O a treia roata care nu este actionata de motor si se poate deplasa pe directia orizontala
- Doua motoare TT ce pot fi dotate cu sistem de encoder
- Patru baterii AA (1.5V) + dispozitiv de prindere
- O baterie de 9V + dispozitiv de prindere
- Senzor ultrasonic de distanta HC-SR04 + suport de prindere
- Servomotor SG90
- Doi senzori IR FC-123
- Senzor IR Reflectiv Pololu QTRX-HD-06A
- Driver Motor L298N
- Mini Breadboard
- Modul cu Buzzer activ
- Fire mama-tata, tata-tata, mama-mama
- Switch
- Banda electrica izolatoare
- Banda alba dublu adeziva
- Conectori si suruburi
- Sasiu
- Header de pini (40buc)
- Baterie externa

Schema Electrica





Conectare Senzor Ultrasonic HC-SR04

- Pinul de GND este conectat la GND-ul de pe Arduino
- Pinul de VCC este conectat la 5V de pe Arduino
- Pini de Trigger si Echo sunt conectati direct la Arduino pe pinul 6, respectiv pinul 7

Conectarea Servomotorului SG90

- Pini de GND si VCC sunt conectati la fel ca senzorul ultrasonic
- Pinul de Signal este conectat direct la Arduino pe pinul 9

Conectarea Senzorilor IR FC-123

- Pini de GND si VCC sunt conectati la fel ca mai sus
- Pini de OUT sunt conectati direct la Arduino pe pinul 2, respectiv pinul 3

Conectarea Matricii de Senzori IR QTRX-HD-06A

- Pini de GND si VCC sunt conectati la fel ca mai sus
- Cei 6 pini analogici de date sunt conectati direct la Arduino pe pini analogici [A0-A5]

Conectarea Senzorului de Buzzer

- Pinii de GND si VCC sunt conectati la fel
- Pinul de OUT se conecteaza direct la Arduino pe pinul 8

Conectarea Driver-ului Motor L298N

- Primeste 6V de la baterie pe pinul de 12+, si GND tot de la aceasta
- Primeste 5V de la Arduino pe pinul de 5+
- Pinii de Input sunt conectati direct la Arduino pe pinii: 4, 5, 12 si respectiv 13
- Pinii de Output sunt legati la cele doua motoarea TT.
- Pinii de Enable sunt conectati direct la Arduino pe pinii de PWM: 10, respectiv 11

Alimentare

- Arduino-ul impreuna cu toti ceilalti senzori sunt alimentati de o baterie externa, folosind conexiunea prin cablu USB
- Driver-ul Motor L298N este alimentat de ansamblu de baterii AA

Software Design

Pentru a realiza partea de software, am folosit mediul de dezvoltare Arduino.

Codul il pot imparti in mai multe etape dupa cum urmeaza:

Etapa de initializare

1. Este inclusa biblioteca suplimentara «servo.h».
2. Sunt definiti majoritatea pinilor folositi in conformitate cu schema hardware.
3. Sunt declarate si initializate restul de variabile globale necesare in rezolvarea probei software.

Etapa de Setup

1. Este setata starea pinilor, si anume cea de *OUTPUT*, *INPUT* sau *INPUT_PULLUP*.
2. Este initializat si configurat un pin specific pentru a controla servomotorul.
3. Este atasata o rutina de intrerupere pentru ambii senzori de **IR** de pe margini.

Etapa de implementare a miscarii

1. Functiile: *go_stop*, *go_forward*, *go_left*, *go_right*, sunt foarte asemanatoare si indeplinesc acelasi scop: de a seta diferite viteze pe motoare si valori pe pinii corespunzatori.
2. Functia *followLine* se ocupa cu calcularea unor sume partiale pentru a decide directia in care sa se indrepte robotelul.
3. Sumele provin din valorile analogice ale pinilor de **IR** aflati pe matrice QTRX-HD-06A.
4. Functiile: *turnObstacleRight* si *turnObstacleLeft* se ocupa cu partea de ocolire a unui obstacol, folosindu-se la baza de functiile definite mai sus.

Etapa de identificare a unui obstacol

1. Sunt definite variabile locale care sa calculeze distanta si durata.
2. Urmeaza o secventa de cod care e menita sa genereze semnalul de trigger catre senzorul de ultrasunete.

3. Apoi este folosita functia de *pulseIn* pentru a extrage durata de timp necesara primirii unui puls sau impuls.
4. Intr-un final se poate aplica o formula matematica pentru a extrage distanta.(*avand in vedere, o mica eroare*)

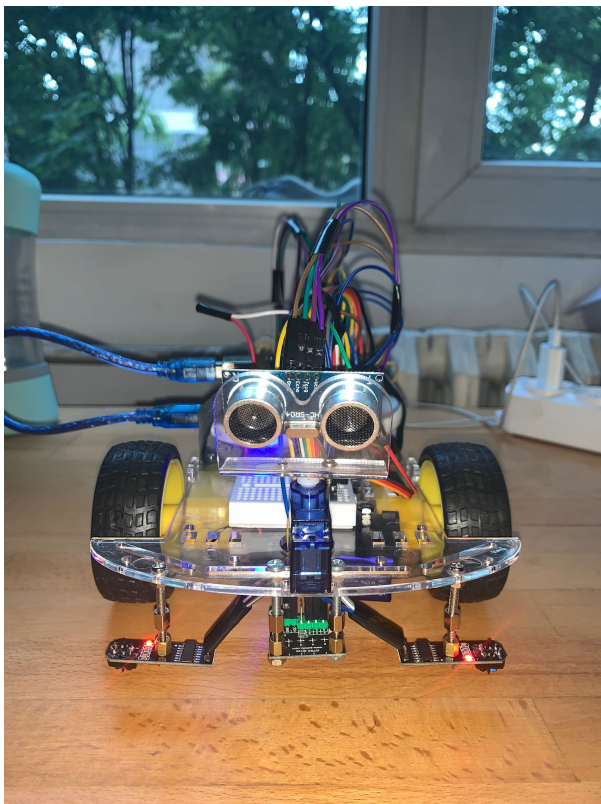
Etapa de evitare a unui obstacol

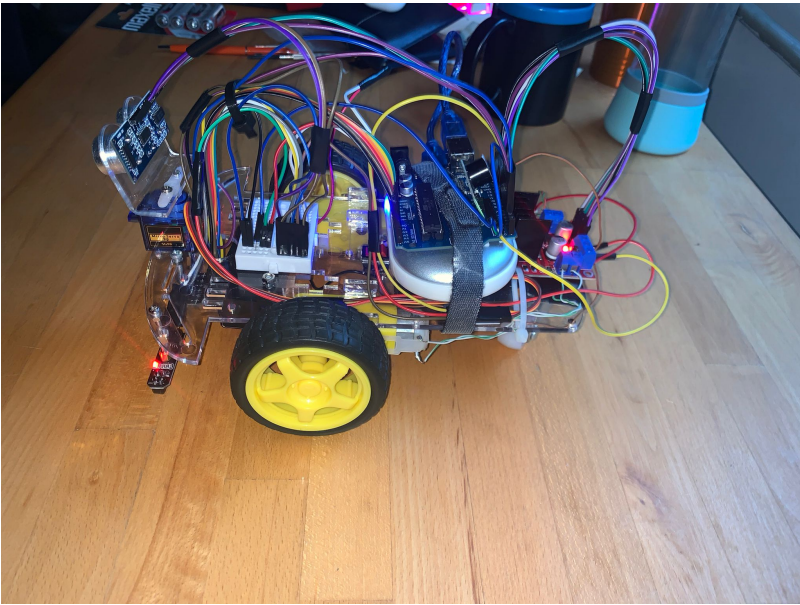
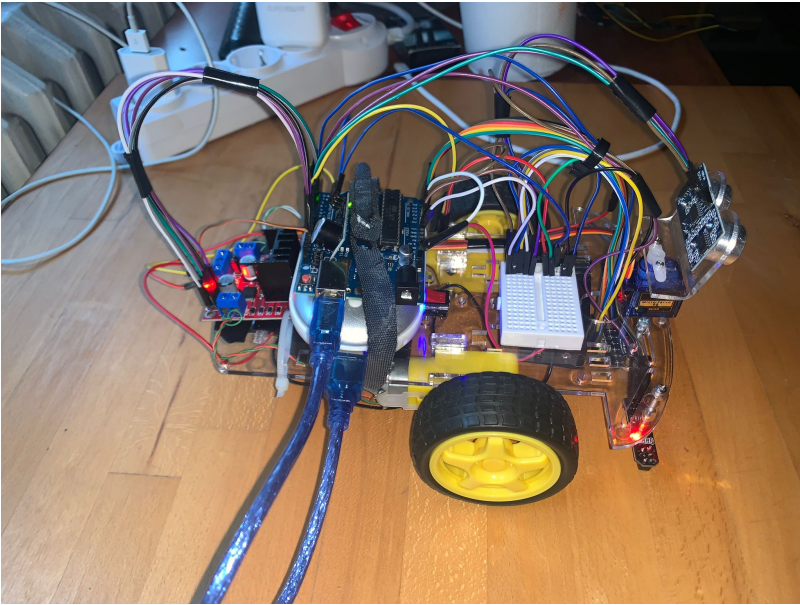
1. Este actionat servomotorul sub trei unghiuri diferite menite sa acopere directiile: "stanga", "dreapta", si "fata".
2. Datorita cablului USB care poate sa agate, am decis in teste sa realizem ocolirea numai pe partea dreapta.

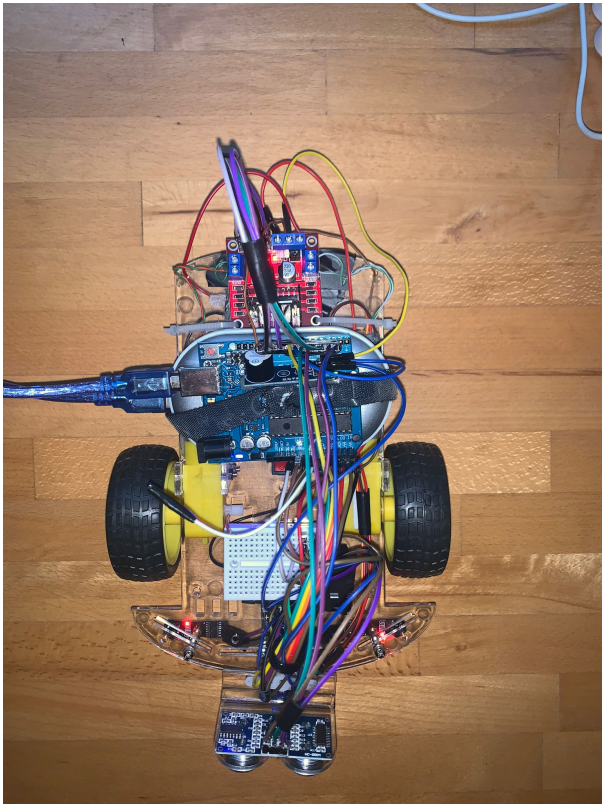
Etapa de loop

1. Calculeaza distanta pana la obiect folosindu-se de functiile descrise mai sus.
2. Daca distanta obtinuta este mai mica decat o valoare de **THRESHOLD**, atunci avem un obstacol in fata si va trebui sa il evitam.
3. In caz contrar, robotul trebuie doar sa urmareasca linia.

Fotografii







Youtube