

Robot ocolitor de obstacole

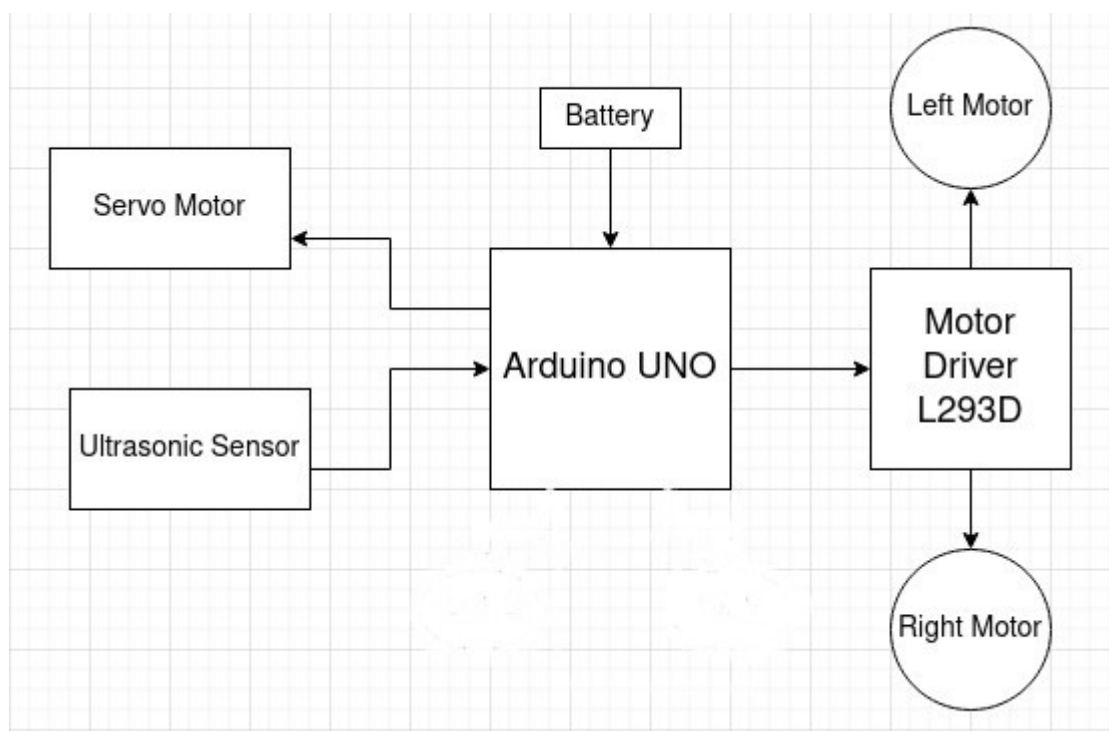
Autor: Asimionesei Daniel

Introducere

Robot complet autonom care evita obstacole bazat pe Arduino. Acesta poate evita orice obstacol cu care se confruntă atunci când se mișcă. Când robotul întâlnește un obstacol în timp ce se deplasa înaintea, face un pas înapoi și se rotește în cel mai bun mod posibil. Fie în direcția stângă, dacă există un alt obstacol în dreapta, fie în direcția dreaptă, dacă există un alt obstacol în partea stângă. Dacă în ambele direcții există obstacole va alege direcția care are distanța cea mai mare până la obstacol.

Descriere generală

Schema bloc:



Componente:

DC Motors (motor de curent continuu)

Caracteristici:

- viteza e data de tensiune;
- direcția e data de polaritate;
- rotație continuă, cât timp motorul este sub tensiune;

Tensiune de alimentare motor: 3V - 6V DC;

Cuplu: 0.8 kg * cm;

RPM: 3V:125rpm, 5V:200rpm, 6V:230rpm;

Curent: 3V:60mA, 5V:100mA, 6V:120mA;

Driver-ul de motoare L298N Dual H-Bridge

Driver-ul se conectează la platforma Arduino folosind 4 pini digitale, conectați la pinii In1, In2, In3 și In4.

Tensiune de alimentare motoare: 5...35V.

Tensiune circuit logic: 5 V (poate genera această tensiune pentru alimentare Arduino).

Poate controla maxim 2 motoare care necesita cel mult 2 Amperi (2000 mA).


Schema circuitului L298N



Fiecare motor are trei pini de control. Astfel, primul motor este controlat de pinii EnA, In1, si In2, iar motorul al doilea de pinii EnB, In3 si In4. Pinii En sunt conectati la nivelul logic 1 prin jumperi, deci prin program vom controla doar pinii In. Sunt disponibile următoarele combinatii:



Ultrasonic Sensor

Senzorul cu ultrasunete utilizează un sonar pentru a determina distanța față de un obiect. 

Masurarea distantei se bazeaza pe diferenta dintre momentul de timp la care s-a transmis pulsul sonic si momentul la care acesta se detecteaza inapoi, avand in vedere viteza sunetului si folosind formula din fizica pentru spatiu.

$$\text{Spatiu} = \text{Viteza} * \text{Timp} \quad (s = v * t)$$

Etapele procesarii:

1. Transmițătorul (trig pin) trimite un semnal: un sunet de înaltă frecvență.
2. Când semnalul găsește un obiect, acesta este reflectat.
3. Emițătorul (ecou) îl primește.



Descriere:

Tensiune: 5V

Tensiune HIGH: 5V

Tensiune LOW: 0V

Unghiul senzorului: 15 grade


Distanța detectată: 2cm - 450cm

Precizie: 0.3cm

Pini:

```
VCC: +5VDC\\
Trig : Trigger (INPUT)\\
Echo: Echo (OUTPUT)\\
GND: GND\\
```

Motor Servo

Spre deosebire de motoarele DC, care produc rotație continuă cât timp sunt conectate la o sursă de tensiune, motoarele servo sunt folosite pentru a obține rotații parțiale, stabile și controlate, pentru efectuarea unor operații cu amplitudine mică dar cu precizie ridicată: poziționare senzori. Motoarele servo au 3 fire, iar culoarea acestora variază în funcție de producător. Culoarea roșie desemnează de obicei Vcc (5V), în timp ce GND este de obicei negru sau maro. Pe lângă aceste două fire de alimentare, există un al treilea, firul de comandă, care este de obicei galben, portocaliu sau alb. Motorul servo nu va executa (de obicei!) o rotație completă, ci va devia de la poziția de echilibru cu un unghi controlat de tensiunea aplicată pinului de semnal. Folosind un semnal PWM pe acest pin, vom avea control asupra unghiului de rotație al motorului. 

Cel mai simplu mod de a controla motoarele de tip servo este prin folosirea bibliotecii Servo.

Specificatii:

Tensiune de alimentare: 4.8V;

Consum curent: 220 ± 50 mA;

Unghi rotire: 180 grade;

Consum redus de curent;

Viteza de funcționare: 0.12 s/60o @ 4.8 V;

Cuplu în blocare la 4.8V: 1.8 kgf*cm;

Frecvență PWM: 50Hz;

Temperatura de funcționare: -30° C - +60° C.

Dimensiuni: 21.5 x 11.8 x 22.7 mm

Arduino UNO

Arduino UNO este o placă de dezvoltare open-source realizată pe baza microcontrolerului ATmega328P. Acesta din urmă prezintă un procesor AVR cu o arhitectură de tip RISC (set restrâns de instrucțiuni).

Schema:



Caracteristici:



Structura plăcii:



Hardware Design

Listă de piese:

- Arduino UNO

Am ales Arduino UNO pentru ca este open-source. Pe lângă acestea include și o comunitate uriașă care se ocupă cu creația și distribuirea de proiecte care au ca scop crearea de dispozitive care pot sesiza și controla diverse activități sau procese în lumea reală.

- Ultrasonic Sensor

L-am ales pentru ca este compatibil cu Arduino și are unele avantaje față de senzorii analogici de distanță: are nevoie doar de pini I / O digitali, are o imunitate mai mare la zgomot și o precizie mai bună. Senzorul emite ultrasunete la o frecvență de 40000Hz care circulă prin aer, iar dacă întâlnește un obstacol, acesta se va întoarce înapoi spre modul, astfel, luând în considerare viteza sunetului se poate calcula distanța până la obiect.

- 2X DC Motors with wheels

Motoarele de curent continuu (DC motors) clasice convertesc energia electrică în lucru mecanic. Viteza de rotație a unui motor este proporțională cu tensiunea de alimentare de la bornele acestuia, iar direcția de rotație depinde de polaritate (conectarea celor 2 fire de alimentare ale motorului la +Vcc și Gnd, sau vice-versa). Motoarele au cutie de viteze (reductor de rotație) cu raport de 1:48, ceea ce înseamnă că pentru o rotație completă a axului extern se efectuează de fapt 48 de rotații ale motorului electric. Folosirea unui reductor are avantajul că mărește forța de acționare, cu costul vitezei. Datorită faptului că motoarele necesită o intensitate a curentului semnificativă pentru a produce mișcare, ele nu pot fi conectate direct la ieșirile (pinii) unui microcontroller. Se impune

separarea semnalelor de comandă de circuitul de putere, și acest lucru se realizează prin folosirea punților H ("H bridges").

L-am ales pentru ca:

- 1) Orientarea câmpului are loc prin intermediul comutatorului mecanic
- 2) Variabilele de control sunt Curentul de curent și Curentul de câmp, măsurate DIRECT de la motor
- 3) Controlul cuplului este direct
- 4) Control precis și rapid al cuplului
- 5) Răspuns înalt de viteză dinamică
- 6) Simplu de controlat
- 7) Tensiunea de lucru a motorului: 3-6V

- Jumper wires
- Motor Driver L298

Punțile H sunt circuite care conțin 4 comutatoare (de obicei tranzistori), numerotate S1, S2, S3 și S4.

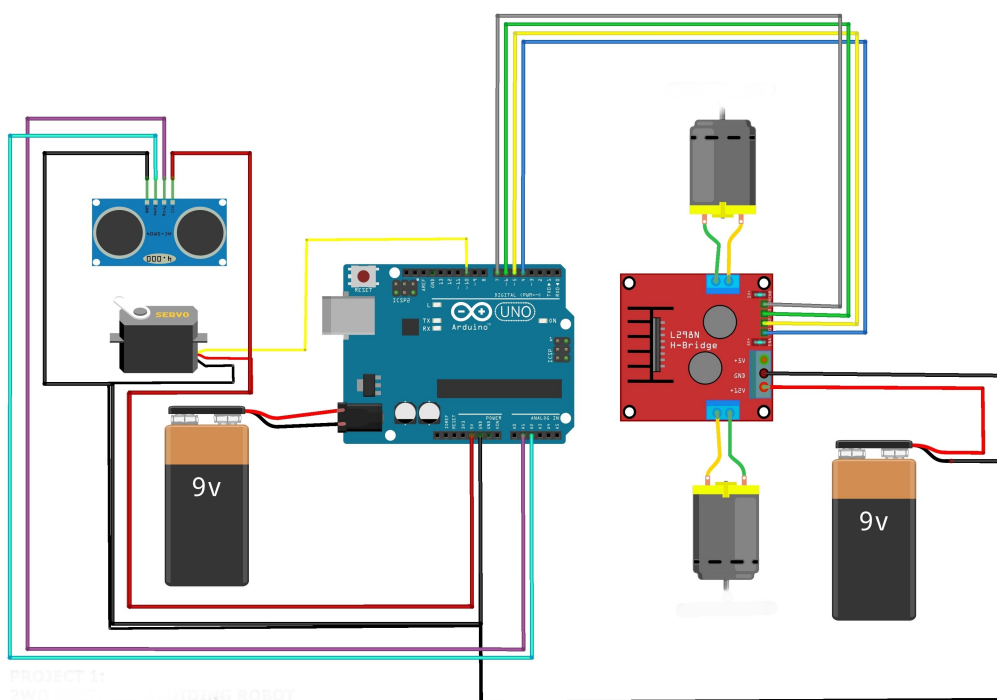


Am folosi puntea duală L298N Dual H-Bridge, pentru ca este capabilă să actioneze două motoare DC în același timp.

- breadboard
- Motor Servo

Am ales Servo motor-ul pentru ca are rotatii parțiale, stabile și controlate, pentru efectuarea unor operații cu amplitudine mică dar cu precizie ridicată. Folosesc acest motor pentru poziționarea senzorului ultrasonic.

- Battery



Schema electrica:**Diagrame de semnal:****Ultrasonic Sensor**

Senzorul cu ultrasunete emite unde ultrasonice la o frecvență de 40 000 Hz și se deplasează prin aer. Dacă există obiecte sau obstacole în cale, acesta va fi reflectat înapoi la modul. Având în vedere timpul de călătorie și viteza sunetului, putem calcula distanța.

Pentru a genera ultrasunete, setez Trig high pentru 10μs. Aceasta va trimite un impuls sonic de 8 cicluri care se va propaga la viteza v și va fi primit în pinul Echo. Pinul Echo va emite timpul în care unda sonoră se deplasează în microsecunde.

**DC Motors**

Într-un circuit analogic, viteza motorului este controlată de nivelul tensiunii. Într-un circuit digital, avem doar două soluții:

-Folosirea unui circuit de rezistență variabilă pentru a controla tensiunea aplicată motorului (soluție complicată, care irosește energie sub formă de căldură)

-Aplicarea intermitentă a tensiunii sub formă PWM.

Când tensiunea este aplicată, motorul este acționat de forța electromagnetică.

Când tensiunea e oprită, inerția cauzează motorul să continue rotație pentru scurt timp.

Dacă frecvența pulsurilor este suficient de mare, acest proces de pornire + mers din inerție permite motorului o rotație uniformă, controlabilă prin logica digitală.

**Motor Servo**

-Lățimea pulsului controlează amplitudinea rotației

-1.5ms-poziția neutră

-1ms-poziție maxim stânga(dreapta)

-2ms-poziție maxim dreapta(stânga)

-Codificare PWM, frecvența purtătoare între 30Hz și 60Hz

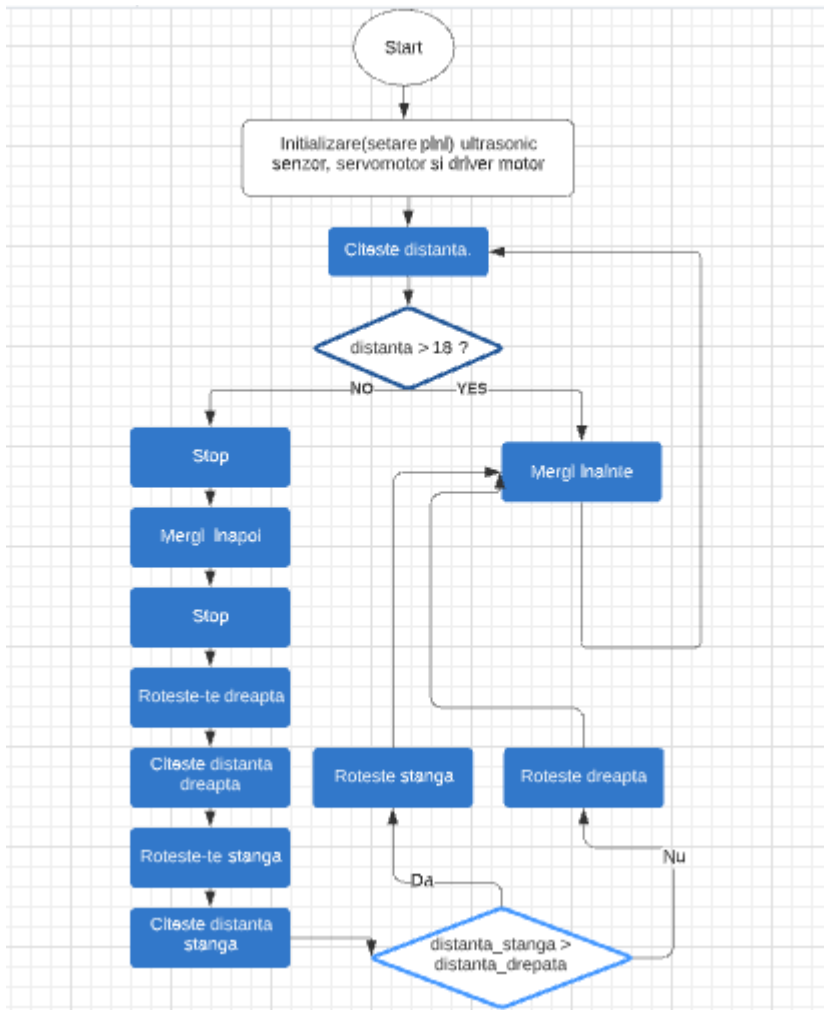


Software Design

Mediu de dezvoltare:

ArduinoIDE

**Diagrama de activitati(UML):**



Biblioteci folosite:

Servo.h

Această bibliotecă permite unei plăci Arduino să controleze servo-motoarele RC.

Folosește cu atenție timers: bibliotecă poate controla 12 servo-uri folosind doar 1 timer.

Servo-urile au roți dințate integrate și un arbore care poate fi controlat cu precizie.

Servo-urile standard permit poziționarea arborelui în diferite unghiuri, de obicei între 0 și 180 de grade.

Servomotoarele de rotație continuă permit reglarea rotației arborelui la diferite viteze.

La Arduino Uno, această bibliotecă va dezactiva PWM pe pinii 9 și 10, indiferent de câte motoare se folosesc.

Functii folosite din bibliotecă:

1) servo.attach(pin) - atașează obiectul Servo la pini

→ servo: un obiect instanță a clasei Servo

→ pin: numărul pinului digital unde va fi atașat semnalul pentru motorul Servo

2) servo.write (angle) - scrie o valoare (0 .. 180) către servo, controlând mișcarea:

Pentru Servo standard setează unghiul axului (grade) cauzând motorul să se orienteze în direcția specificată.

Functiile principale ale codului sursa:

void setup() → setez valorile pinilor pentru senzorul ultrasonic, servomotor si driver motor, masor distanta initiala si setez pozitia servo motoru-lui la 125 grade;

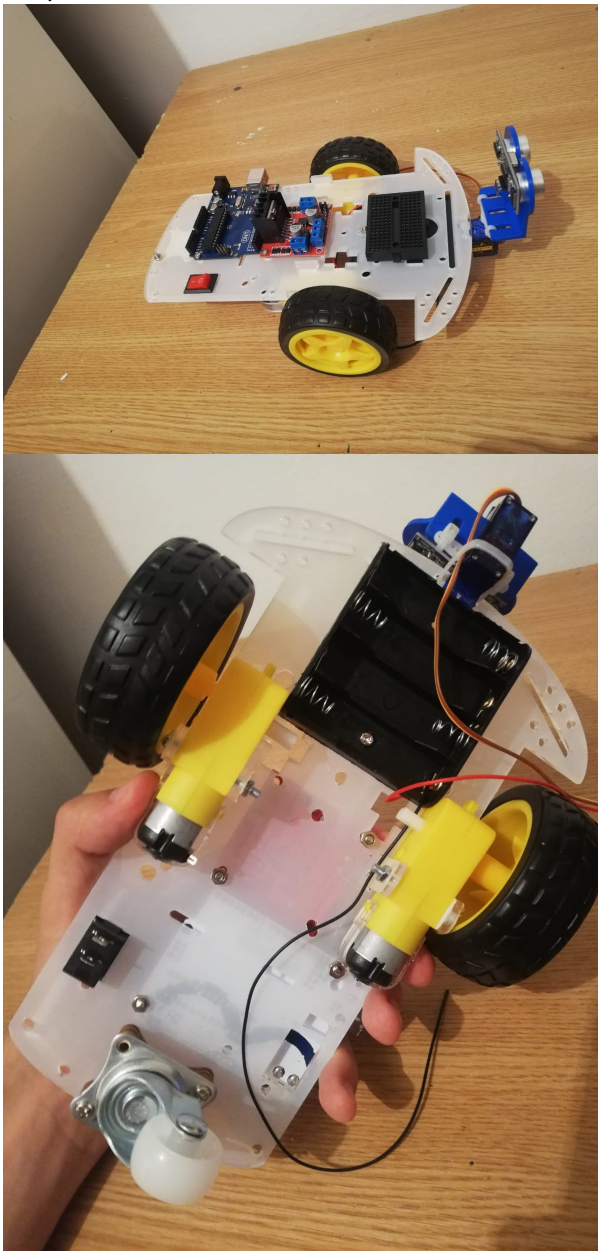
void loop() → logica programului ilustrata in diagrama de activitati;

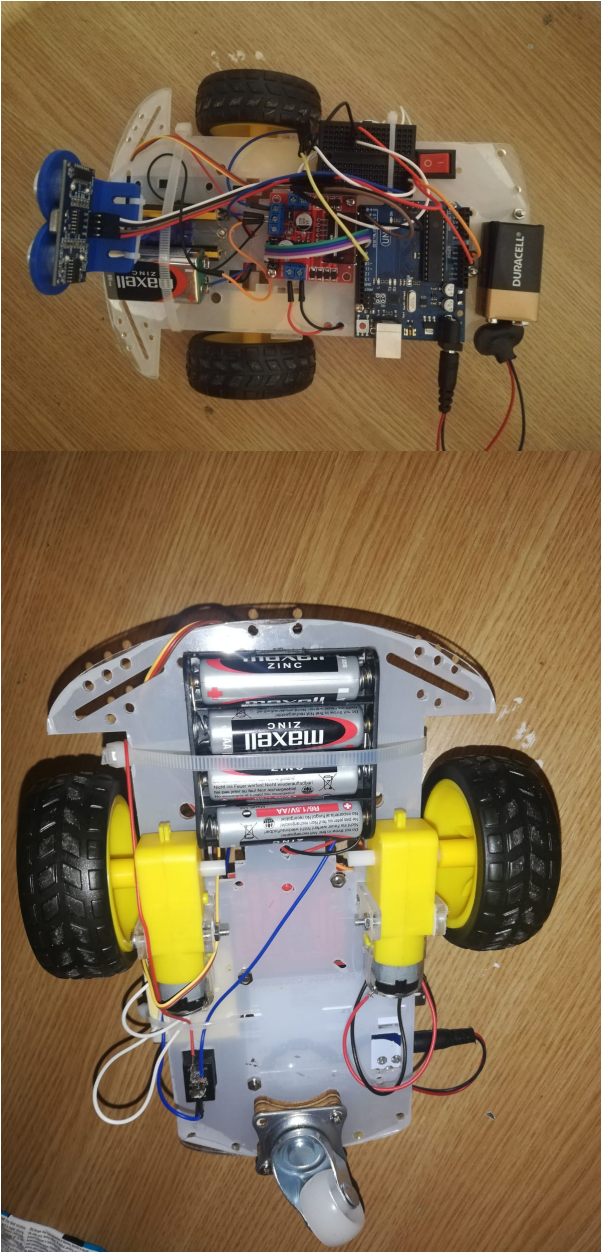
void leftRotate() → rotesc roata din dreapta (pun pinii corespunzatori rotii din dreapta pe HIGH si pinii rotii din stanga pe LOW);

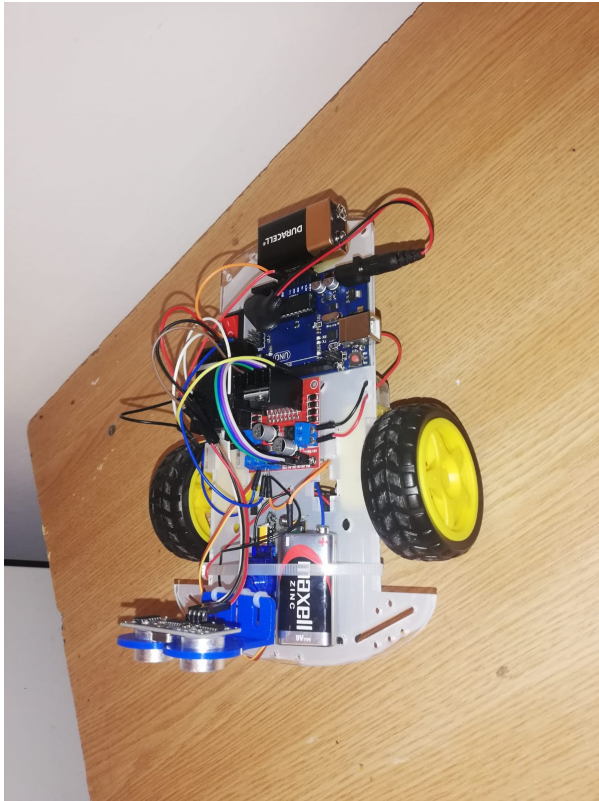
`void rightRotate()` → rotesc roata din stanga (pun pinii corespunzatori rotii din stanga pe HIGH si pinii rotii din dreapta pe LOW);
`void start_back_robot()` → rotesc rotile in sens invers acului de ceasornic;
`void start_front_robot()` → rotesc rotile in sensul acului de ceasornic;
`void stop_robot()` → pun toti pinii asociati rotilor pe LOW;
`int leftDistance()` → rotesc servo motorul la 180 grade si calculez distanta pana la cel mai apropiat obiect din stanga pe baza functiei `readDistance()`;
`int rightDistance()` → rotesc servo motorul la 55 grade si calculez distanta pana la cel mai apropiat obiect din dreapta pe baza functiei `readDistance()`;
`int readDistance()` → calculez, folosind functia `pulseIn()`, lungimea impulsului în microsecunde si folosind formula $distance = (duration/2) / 29.1$, calculez distanta parcursa de unda in cm.

Rezultate Obținute

Etape asamblare robot:







Concluzii

Proiectul ales mi s-a parut foarte interesant deoarece am invatat lucruri noi atat pe partea de software (lucru cu biblioteci noi) cat si pe partea de hardware (sa lipesc fire, sa asamblez un circuit de la 0 si sa fac debugg pe fiecare componenta in parte). Pe parcursul proiectului am avut nenumarate probleme, de la scurtcircuit pana la probleme cu alimentarea sau lipituri gresite. Un exemplu de problema intalnita : nu conectam gnd-urile de la driver L298N , servomotor si senzor ultrasunet in acela loc si nu stiam de ce nu merge robotul.

Bibliografie/Resurse

[robot_ocolitor_de_obstacole.zip](#)
[robot_ocolitor_de_obstacole.pdf](#)

Link catre pagina: https://ocw.cs.pub.ro/courses/pm/prj2021/abirlica/robot_ocolitor_de_obstacole

<https://programmersought.com/article/5307380139/>

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

<https://create.arduino.cc/projecthub/ambhatt/dc-motor-controlling-library-a74319>

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/abirlica/robot_ocolitor_de_obstacole

Last update: **2021/05/31 01:09**

