

Obstacle Avoiding Robot

Grupa: 335CC

Nume: Horovei Andreea-Georgiana

E-mail: andreea.horovei@stud.acs.upb.ro

Introducere

Ideea proiectului presupune construirea unui robot care se deplaseaza evitand obstacolele din jurul sau. Robotul se deplaseaza inainte pana la detectarea unui obstacol, apoi se va misca putin inapoi si va „privi” spre stanga si spre dreapta pentru a alege in continuare directia corecta de deplasare. Robotul va avea atasat un ecran LCD care va afisa informatii pe baza datelor preluate de la senzorul ultrasonic.

Scopul acestui proiect este de a pune in practica unele din notiunile prezentate in cadrul laboratoarelor de PM. Un astfel de robot, care ne aminteste de aspiratoarele automate care se gasesc in magazine, poate fi folosit, de exemplu, pentru a desena harta unei suprafete sau pentru a determina aria unei camere.

Descriere generală

Robotelul va folosi un senzor ultrasonic pentru detectarea distantelor. Senzorul va fi rotit cu ajutorul unui servomotor. La detectarea unui obstacol, un LED rosu se va aprinde iar buzzer-ul va semnala prezenta obstacolului prin 3 sunete simple. Robotul se va misca putin inapoi, in timpul mersului cu spatele fiind aprins un LED albastru (in acest timp buzzer-ul va semnala deplasarea cu spatele prin alte 3 sunete simple). Vor fi masurate distantele catre stanga si dreapta, urmand ca robotul sa aleaga directia potrivita, semnaland deplasarea intr-un anumit sens prin aprinderea unuia dintre cele 2 LED-uri galbene plasate sugestiv in partea stanga, respectiv dreapta a breadboard-ului montat pe robotel.

Ecranul LCD este folosit, de asemenea, pentru a afisa mesaje sugestive la schimbarea directiei de mers. Pe breadboard este atasat un buton care poate fi folosit pentru oprirea celor 2 motoare in timpul deplasarii robotelului.

- Schema Bloc:



Hardware Design

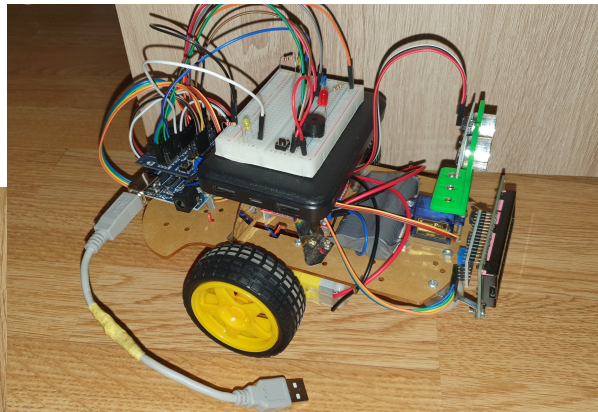
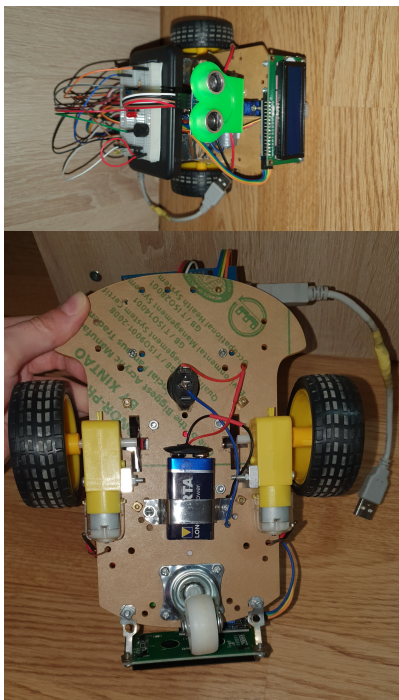
Componentele pe care le-am folosit la realizarea proiectului sunt:

- [Arduino UNO R3](#)
- [kit sasiu + 2 motoare DC](#)
- [Modul senzor ultrasonic-detector distanta HC-SR04](#)
- [Servomotor SG90](#)
- [L298N modul motor driver](#)
- [Sensor Shield Arduino Expansion Board v5.0](#)
- [LCD display 1602](#)
- [Adaptor I2C pentru LCD 1602](#)
- [Buzzer pasiv](#)
- [Breadboard 400](#)
- [Suport pentru senzor ultrasonic](#)
- [Buton on/off](#)
- [4 LED-uri + rezistente 220 ohmi, 1 Buton](#)
- [1 Baterie 9V](#)
- [Fire mama-tata, mama-mama](#)
- [Baterie externa PROMATE volTag-10 \(capacitate 10000 mAh, voltaj iesire 5 V, amperaj iesire 2 A\)](#)

Schema electrica:



Montajul realizat:



Software Design

Dezvoltarea a fost facuta in Arduino IDE.

Bibliotecile externe folosite:

- **Servo.h**: folosirea servomotorului (functii folosite: attach, write)
- **NewPing.h**: folosirea senzorului ultrasonic (functie folosita: ping_cm)
- **LiquidCrystal_I2C** - sursa: <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>: am folosit functiile de print, write, respectiv createChar.

Descrierea implementarii

Logica de implementare urmareste un algoritm simplu folosit de robot astfel incat acesta sa se deplaseze fara sa loveasca obiectele care ii stau in cale. Logica de alegere a directiei potrivite de deplasare a robotului pe care am implementat-o ar putea fi rezumata astfel:

```
if obstacle_dist < DIST_LIMIT:
    left_dist = determine_left_obs_distance()
    right_dist = determine_right_obs_distance()
    if max(left_dist, right_dist) > obstacle_dist:
        if left_dist > right_dist:
            choose_left_direction()
        else:
            choose_right_direction()
```

Astfel, robotul se deplaseaza inainte pana cand un obstacol este identificat la o distanta de cel mult DIST_LIMIT=35 cm in fata acestuia. In acest moment, trebuie sa determinam distantele la stanga, respectiv dreapta. In acest scop, dupa identificarea obstacolului, robotul de deplaseaza pe o distanta mica cu spatele, dupa care servomotorul este rotit mai intai la stanga, apoi la dreapta (de fiecare data la o diferenta de 90 grade de pozitia initiala). In acest timp, senzorul ultrasonic este folosit pentru a prelua distantele in cele doua directii.

Robotul va alege directia potrivita pe baza distantelor masurate, conform cu logica descrisa de pseudocodul de mai sus. Motoarele sunt controlate astfel incat sa se faca virajul in directia corecta. Daca niciuna dintre distantele determinate la stanga, respectiv la dreapta nu este mai mare decat distanta fata de obstacolului identificat de robot inaintea deplasarii cu spatele, atunci robotul se va deplasa iar cu spatele, urmand sa masoare din nou distantele in cele 2 directii.

Variabile importante:

Am definit macro-uri pentru pinii pe care i-am atribuit pentru controlul directiei de rotire a fiecarui motor, respectiv pentru pinii atribuiti pentru: servomotor, buzzer, cele 4 LED-uri, butonul de start/stop al motoarelor, si pentru pinii TRIG si ECHO ai senzorului ultrasonic. Am salvat ca variabile globale cele 3 distante folosite in cadrul algoritmului pentru stabilirea drumului robotului: current_dist, left_dist, right_dist.

Distanta principala, current_dist, este actualizata cu valoarea distantei catre un eventual obstacol, aceasta fiind determinata cu ajutorul functiei *ping_cm()* din biblioteca *NewPing*. Actualizarea

distantelor catre un eventual obstacol plasat la stanga sau la dreapta este realizata folosind aceeasi functie.

Funcții importante:

Toata logica algoritmului folosit de robot pentru o deplasare corecta este inclusa in functia loop().

Pentru o mai buna organizare, am definit functii simple care sa se ocupe cu setarea corespunzatoare a valorilor pentru cei patru pini care sunt asociati motoarelor, pentru fiecare tip de miscare pe care o poate face robotul:

- **move_A_B_forward():** seteaza pinii asociati celor 2 motoare astfel incat robotul se deplaseaza in fata, si actualizeaza mesajul afisat pe ecranul LCD
- **move_A_B_backward():** activeaza pinii motoarelor in mod corespunzator pentru mersul cu spatele
- **go_left_direction():** seteaza pinii motoarelor astfel incat motorul stang sa se deplaseze in fata, iar motorul drept in spate (aplica pentru aceasta un delay de 200 ms); dupa cele 200 ms se va continua mersul in fata cu ambele roti, in acest fel fiind realizat virajul la stanga.
- **go_right_direction():** similar cu functia go_left_direction, realizeaza acum virajul la dreapta.
- **stop_motors():** seteaza pe LOW toti cei patru pini asociati motoarelor.

Rezultate Obținute

Am reusit sa duc proiectul la final si sa obtin un robotel functional, care sa respecte cerintele pe care mi le-am propus la alegerea temei.

Demo proiect:



Concluzii

Fiind primul proiect hardware pe care l-am realizat, pot spune ca a fost interesant sa lucrez la el. Desi pe parcurs am intampinat cateva probleme (in principal pe partea hardware si anume in legatura cu alimentarea si functionalitatea celor doua motoare DC), am reusit sa duc proiectul la bun sfarsit. Contrar asteptarilor, am descoperit ca exista foarte multe resurse utile pentru a lucra cu Arduino. Consider ca a fost un proiect util deoarece m-a ajutat sa inteleg mai bine anumite notiuni prezentate de-a lungul semestrului, si, in plus, rezultatul obtinut este unul satisfacator ☑.

Download

Cod sursa:

[obstacleavoidingrobot_horovei_andreea_335cc.zip](#)

Schema bloc + schema electrica:

[obstacle_avoiding_robot_scheme.zip](#)

Readme:

[readme_obstacle_avoiding_robot.docx](#)

Jurnal

21.04.2021 - Obținerea acordului asistentului în privința temei alese.

23.04.2021 - Au sosit o parte din piesele de care am nevoie. Am realizat montarea șasiului (suportul împreună cu cele două motoare cu roți, respectiv roata mobilă poziționată în partea din față a robotelului).

25.04.2021 - Realizarea schemei bloc și crearea paginii de wiki.

29.04.2021 - Am realizat primii pași pentru începerea montării hardware-ului: am identificat legăturile dintre componente și cum urmează să le folosesc pe fiecare. Am testat funcționalitatea individuală a LCD-ului, a senzorului ultrasonic și a servo-motor-ului.

01.05.2021 - Am început efectiv montajul hardware. Am lipit firele de alimentare ale motoarelor, respectiv pinii adaptor-ului I2C la ecranul LCD.

02.05.2021 - Am montat pe șasiu suportul cu cele 4 baterii de 1.5V, respectiv servo-motor-ul.

03.05.2021 - Am continuat cu montajul hardware, am adăugat pe șasiu: placuta de Arduino și Sensor shield-ul atasat acesteia, puntea H dublă folosită pentru cele 2 motoare DC, respectiv am realizat un suport pentru bateria de 9V și altul pentru breadboard. Am montat pe servo-motor suportul pentru senzorul ultrasonic.

07.05.2021 - Am adăugat un întrerupător pentru cele 4 baterii de 1.5V. Am finalizat conexiunile dintre componente. Am început implementarea.

14.05.2021 - Am finalizat implementarea. Robotelul este funcțional. Am testat și identificat faptul că cele 4 baterii care alimentează motoarele nu sunt cea mai bună soluție (după ceva timp robotelul nu mai are aceleași "performanțe" de viteză).

15.05.2021 - Am adăugat pe breadboard buzzer-ul și am implementat partea de generare a sunetelor la detectarea de obstacole, respectiv la mersul cu spatele.

16.05.2021 - Am adăugat pe breadboard LED-urile și am adăugat implementarea pentru aprinderea acestora conform cu detaliile din descriere.

21.05.2021 - Am înlocuit bateria de 9V cu o baterie externă cu un voltaj de 5V, respectiv cele 4 baterii de 1.5V cu o baterie de 9V. Am testat din nou funcționalitatea robotelului.

22.05.2021 - Am realizat schema hardware a proiectului.

23.05.2021 - Am actualizat lista de componente pe pagina de wiki conform cu modificarile realizate.

Bibliografie

- <https://www.youtube.com/watch?v=IsLsZTinUIY> - folosire Arduino Sensor Shield v5.0
- <https://www.learnrobotics.org/blog/arduino-lcd-using-sensor-shield-v5/> - conectarea unui modul LCD la placa de expansiunea sensor shield pentru Arduino
- <http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf> - conectarea a doua motoare DC la L298N dual H-bridge motor driver
- <https://www.youtube.com/watch?v=dyZolgNOomk> - tutorial pentru folosirea modului L289N
- <https://forum.fritzing.org/> - aici am gasit modulele componentelor folosite pentru a le putea importa in Fritzing, la realizarea schemei hardware a proiectului.
- <https://lastminuteengineers.com/arduino-1602-character-lcd-tutorial/> - afisarea unor caractere speciale pe ecranul LCD

Document

Documentul în format PDF: [Obstacle Avoiding Robot](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/obstacle_avoiding_robot



Last update: **2021/05/30 22:22**