

Mihai POGONARU (78288) - Obstacle Avoiding Robot

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul consta in realizarea unui robot care sa se poata deplasa singur evitand obstacolele intalnite pe parcurs prin schimbarea directiei.

Descriere generală

❌ La pornire robotul va incepe sa mearga in linie dreapta (daca nu este niciun obstacol in fata sa). La detectarea unui obstacol se va schimba directia de deplasare prin viraje. Viteza robotului si intensitatea virajului vor depinde de distanta pana la primul obstacol.

Hardware Design

Lista de piese:

- Kit sasiu robot 2WD
- Driver de motoare L298n
- Senzor ultrasonic distanta HC-SR04

Schema electrica: ❌

Software Design

Ca IDE am folosit Visual Studio Code, librarii sau surse 3rd-party. Am folosit api-ul de LCD din laborator pentru debug.

Am folosit timer-ul 1 pentru senzor pentru ca trebuiau numarate uS-le in care pin-ul ECHO este pozitiv, iar senzoul masoara pana la 4m → 12000 uS. Am folosit timer-ul 2 pentru PWM-ul motoarelor. Viteza ambelor motoare este calculata in intreruperea pin-ului de ECHO.

Detalii implementare:

In functia main se initializeaza LCD-ul pentru debug, portul motorului, variabilele globale folosite, timerul 2 care va fi folosit pentru PWM-ul motoarelor, pinii de echo si trigger si intreruperea pe pinul de echo.

In bucla principala a programului se da un impuls scurt de 10uS pe pinul trigger pentru a porni o citire, dupa care se asteapta 20mS pentru echo.

In intreruperea pinului echo se verifica starea acestuia: cand este high se porneste timerul 1 care va numara $2 * uS$ -le pe care pulsul ultrasonic le va petrece in aer.

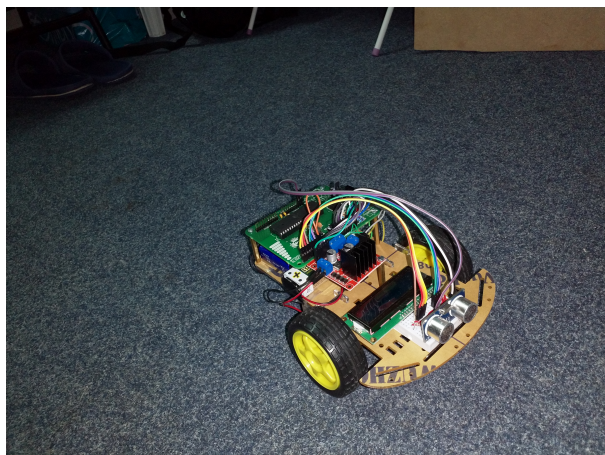
Cand este low se opreste timerul, se citeste valoarea registrului contor si se calculeaza distanta pana la cel mai apropiat obstacol. Apoi se apeleaza decide care va cataloga distanta si va trece robotul in starea corespunzatoare de miscare. Robotul are 5 stari de miscare: viraj rapid, viraj normal, viraj incet, mers normal si mers rapid. In functie de aceasta stare se va calcula viteza fiecarui motor in % din viteza maxima. Viteza efectiva este data de `adjust_pwm` care, in functie de procentajul de viteza al fiecarui motor, seteaza PWM-ul pentru motoare (acesta va avea o valoare intre `MIN_PWM` si `MAX_PWM`) si schimba directia de deplasare daca procentajul este negativ.

Exista si un caz exceptional in care distanta este ≤ 6 cm, caz in care, dupa schimbarea starii, se va apela functia `go_back()` care seteaza ambele motoare la -100% si asteapta 400ms pentru a redresa robotul dintr-un posibil impact.

Viteza de deplasare variaza cu distanta in cazul starilor de mers si este constanta pe motorul principal (cel care 'merge inainte') in cazul celor de viraj. La starile de viraj, viteza motorului secundar variaza in functie de distanta: cu cat distanta este mai mica, cu atat viteza acestuia va fi mai departata de cea a motorului principal $\rightarrow$ virajul va fi mai rapid.

Rezultate Obținute

Robotul se poate deplasa evitand obstacolele, in majoritatea cazurilor (sunt unele unghiuri pe care senzorul nu le percepe). De asemenea, functioneaza in trepte de viteza inclusiv la curbe si se poate redresa in cazul unui impact



Video: [YouTube](#)

Concluzii

A fost un proiect interesant, nu am reusit sa-l fac exact cum imi propusesem (as fi avut nevoie, probabil, de trei senzori pentru a-l face sa evite mai bine obstacolele), dar am invatat cum sa folosesc un senzor de distanta, un driver de motoare, faptul ca motoarele consuma mult si au nevoie de bateria proprie (probabil ar trebui doua baterii de 9v in serie pentru ca driverul sa stabilizeze consumul) si cum sa conectez un LCD 16x2. Si cum sa lipesc

Download

Cod: [Cod ZIP](#)

Schema electrica: [Schema EAGLE](#)

Bibliografie/Resurse

- DataSheet HC-SR04: [PDF](#)
- Tutorial driver L298N: [Youtube](#)
- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/cpetrisor/pogo>

Last update: **2021/04/14 15:07**

