

# Maze Runner

## Autor

Cătălin STÎRCU

## Introducere

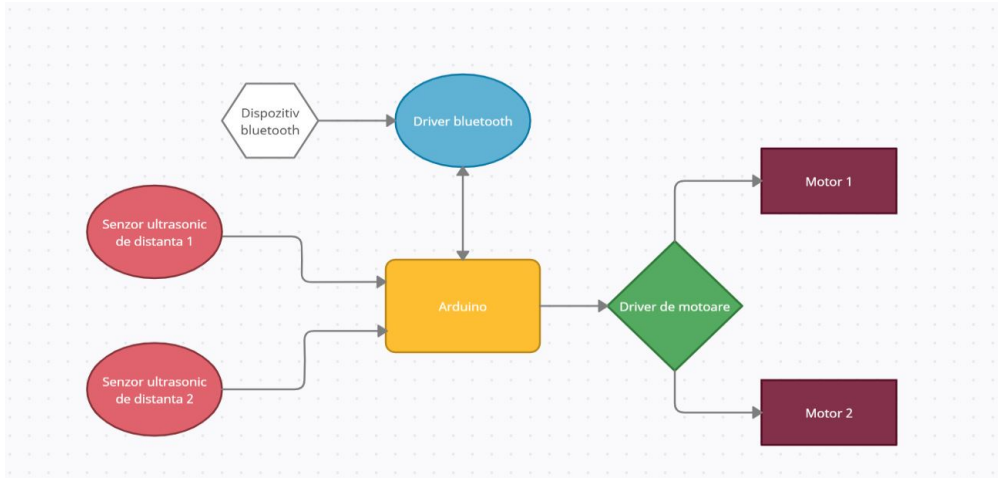
- acest proiect prezintă un robot ce caută să găsească ieșirea dintr-un labirint
- scopul acestui proiect este rezolvarea oricărui labirint folosind regula de dreapta, adică robotul urmărește peretele drept până la finalul labirintului
- mi-a venit această idee de la concursurile de roboți unde se regăsește o categorie ce are ca scop crearea unui astfel de robot și am vrut să văd dacă aș fi în stare să construiesc așa ceva
- utilitatea acestui proiect se regăsește în secreția de endorfină creată în urma construirii și perfecționării proiectului și în concurența cu alte persoane pentru a vedea al cărui robot rezolvă labirintul mai repede

## Descriere generală

### Sumar implementare

Acest robot folosește doi senzori de distanță pentru a afla dacă este o cale liberă în partea din dreapta sau este un perete în față. În primul caz virează spre dreapta, în al doilea, virează spre stânga. Direcția de mers a motoarelor ce controlează roțile este manipulată de driver-ul de motoare ce este mai departe controlat de plăcuța Arduino după codul din ea. Acest robot mai are și funcția de start și stop ce poate fi accesată cu ajutorul modulului bluetooth ce poate fi controlat cu orice alt dispozitiv bluetooth.

### Schemă bloc

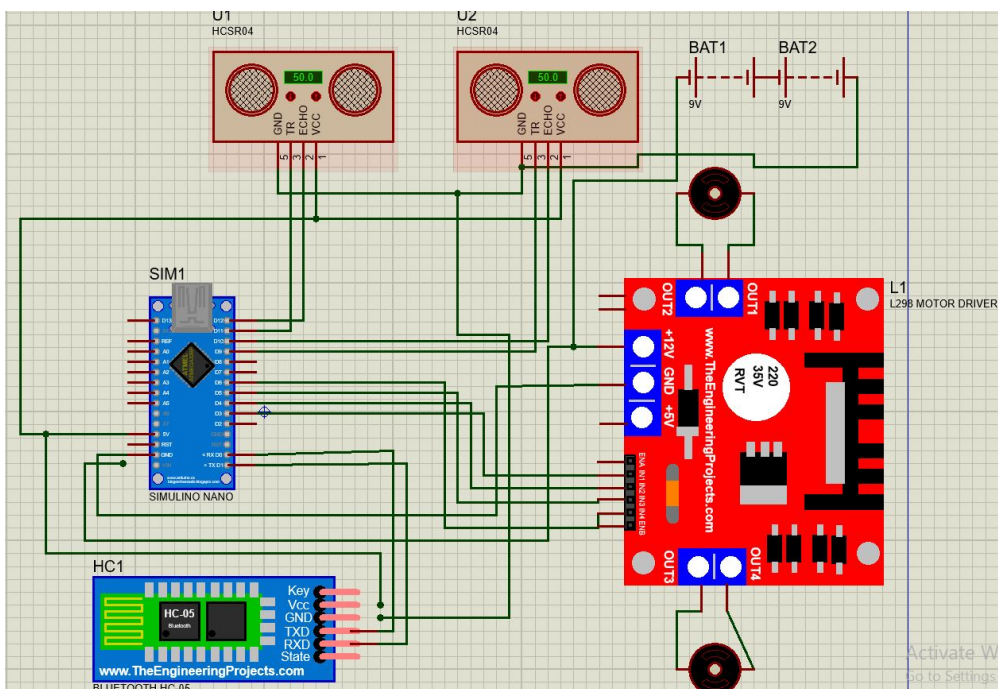


## Hardware Design

### Listă de componente

- Placă Arduino Nano
- Senzori de distanță HCSR04 x2
- Baterii 9V x2
- Driver Bluetooth HC-05
- Driver de motoare L298N
- Motoare x2

### Schemă electrică



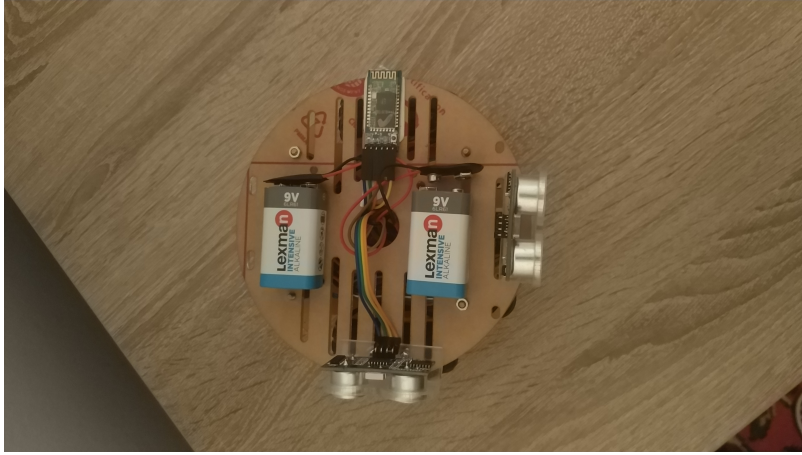
## Software Design

Mediu de dezvoltare: Arduino IDE

- Funcția `walkForward` primește ca parametru un întreg  $x$  și setează motoarele pentru a merge înainte timp de  $x$  milisecunde, după care le setează să se oprească timp de 50 de milisecunde.
- Funcția `turnRight` primește ca parametru un întreg  $x$  și apelează funcția `walkForward` pentru 150 de milisecunde la începutul și sfârșitul funcției, după care setează motoarele să întoarcă înspre dreapta.
- Funcția `turnLeft` primește ca parametru un întreg  $x$  și apelează funcția `walkForward` pentru 150 de milisecunde la începutul și sfârșitul funcției, după care setează motoarele să întoarcă înspre stânga.
- Funcția `spaceUntilFront` întoarce disanța până la peretele din față.
- Funcția `spaceUntilRight` întoarce disanța până la peretele din dreapta.
- Funcția `setup` setează starea pinilor.
- Funcția `loop` așteaptă semnalul de start din partea driver-ului de bluetooth, iar când îl primește testează să vadă dacă mai este distanță până la peretele din față, în caz contrar întoarce înspre stânga, iar dacă mai este testează dacă este un perete în dreapta, dacă este robotul o ia înainte, altfel o ia la dreapta, testând mereu dacă are loc în față și dacă trebuie să se oprească din cauza semnalului primit prin bluetooth.

## Rezultate Obținute





Link youtube <https://www.youtube.com/watch?v=GfzTKmOIQQE>

## Concluzii

Deși proiectul a ieșit destul de bine, se puteau face mai multe îmbunătățiri:

- În primul rând nu este nicio modalitate concretă de a îi spune robotului să se întoarcă 90 de grade
- În al doilea rând, cauciucurile sunt de proastă calitate și au probleme pe anumite tipuri de teren
- În al treilea rând, senzorii sunt de proastă calitate, unul dintre ei s-a defectat, noroc că aveam unul de rezervă, distanța nu este exactă, iar uneori este mult mai mare
- Nu există nicio cale de calcul a distanței parcurse, deci nu se poate face un algoritm care să repete traseul, fără a folosi senzorii pentru a doua parcurgere de labirint
- Senzorul bluetooth trebuia pus împreună cu un buton, pentru a nu fi nevoie să fie deconectat de fiecare dată când trebuie încărcat codul

## Download

[mazerunner.zip](#)

## Bibliografie/Resurse

<https://create.arduino.cc/projecthub/electropeak/getting-started-with-hc-05-bluetooth-module-arduino-e0ca81>

<https://create.arduino.cc/projecthub/ryanchan/how-to-use-the-l298n-motor-driver-b124c5>

<https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/maze\\_runner](http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/maze_runner)



Last update: **2021/06/03 08:04**