

# Tesla 2

## Introducere



Proiectul reprezinta o varianta foarte simplificata, in miniatura, a unei masini, cu functionalite de evitare a accidentelor, fie prin franare, fie prin viraj. Masina este controlata prin Bluetooth, folosind o aplicatie mobila, in sa are si optiunea de "autopilot". In acest mod, masinii i se stabileste o directie/ruta de urmat, pe care va incerca apoi sa o urmeze, ocolind toate obstacolele pe care le intalneste. Masina va avea, de asemenea, un LCD si un buzzer, pentru a putea afisa informatii despre ruta si obstacole.

## Descriere generală



Masina este formata din sasiu plus 2 motoare electrice, atasate la rotile fata stanga, respectiv fata dreapta. Pe partea electronica, masina contine un Arduino UNO, cutie baterii pentru alimentare, modul Bluetooth, un senzor cu ultrasunete montat pe un servo, pentru vedere la un unghi de aproximativ 40-50 grade, LCD si buzzer si un driver dual de motoare.

Initial, Arduino face pairing-ul intre modulul Bluetooth si un telefon cu aplicatia de controlare instalata. Dupa ce conexiunea e realizata, poate incepe sa primeasca comenzi de la modul: accelerare sau franare si viraj. De asemenea, foloseste senzorul cu ultrasunete pentru a detecta obstacole. Cu cat un obstacol este mai aproape, cu atat viteza maxima permisa a masinii scade, pana cand devine 0 atunci cand obstacolul este foarte aproape.

Atunci cand se activeaza functia de autopilot, masina va incerca sa pastreze directia initiala, in sa ocolind obstacolele: acum, in loc sa franeze, va vira in directia in care nu este niciun obstacol. Folosind servo-ul pe care este montat senzorul de distanta, va "privi" in stanga si dreapta pentru a decide in ce directie nu se afla obstacole, pentru a vira.

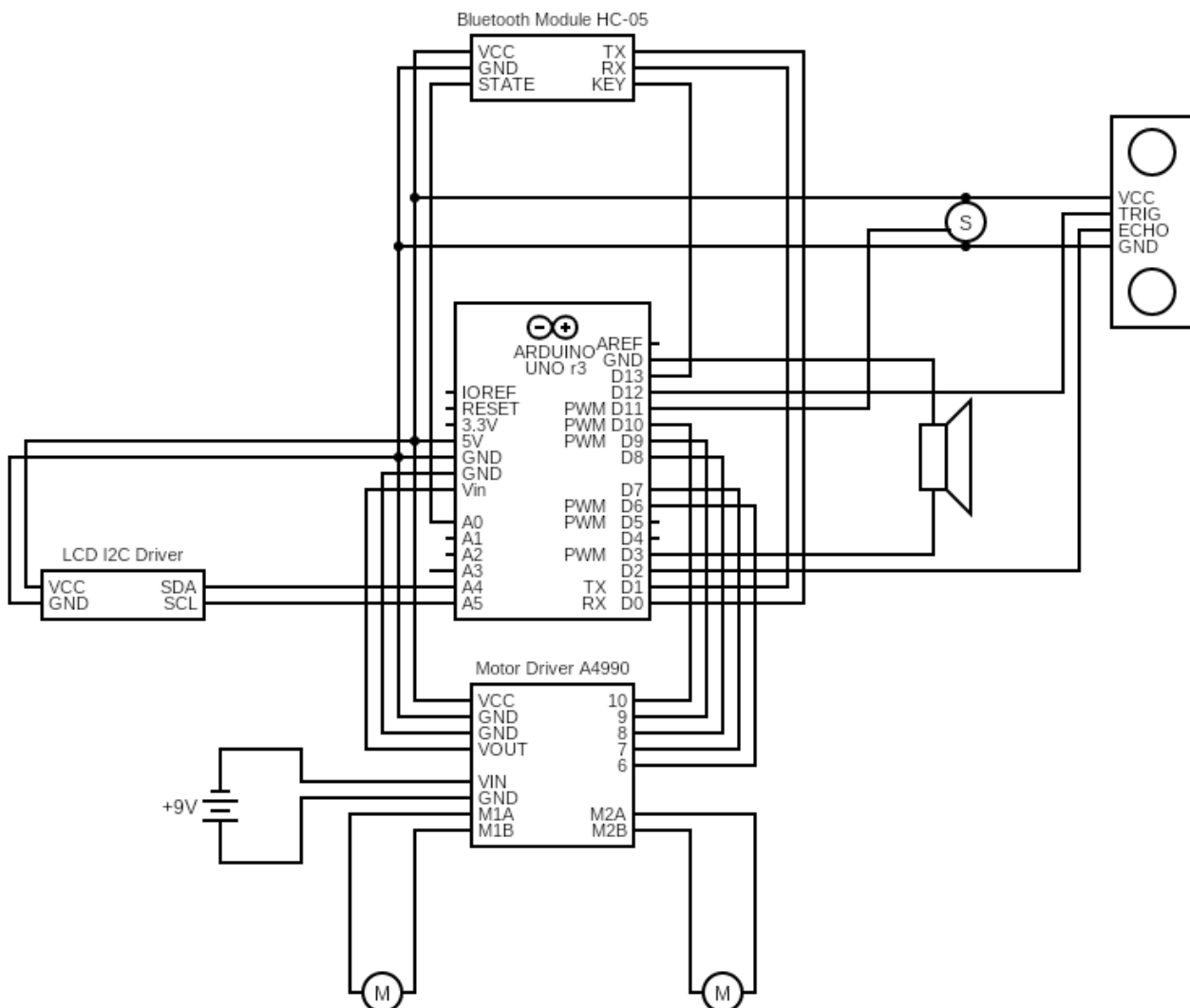
In acest timp, pe LCD se vor afisa informatii utile, cum ar fi: daca autopilotul este activat si ce obstacole sunt detectate. Buzzer-ul va fi folosit pentru a emite niste beep-uri scurte, de exemplu atunci cand pairing-ul se termina cu succes, sau cand semnalul se pierde (se taie conexiunea Bluetooth). De asemenea, pe modul autopilot, masina va semnala prin multe beep-uri atunci cand se blocheaza (de exemplu, atunci cand ajunge intr-un colt si in orice directie ar vira tot s-ar lovi de zid).

# Hardware Design

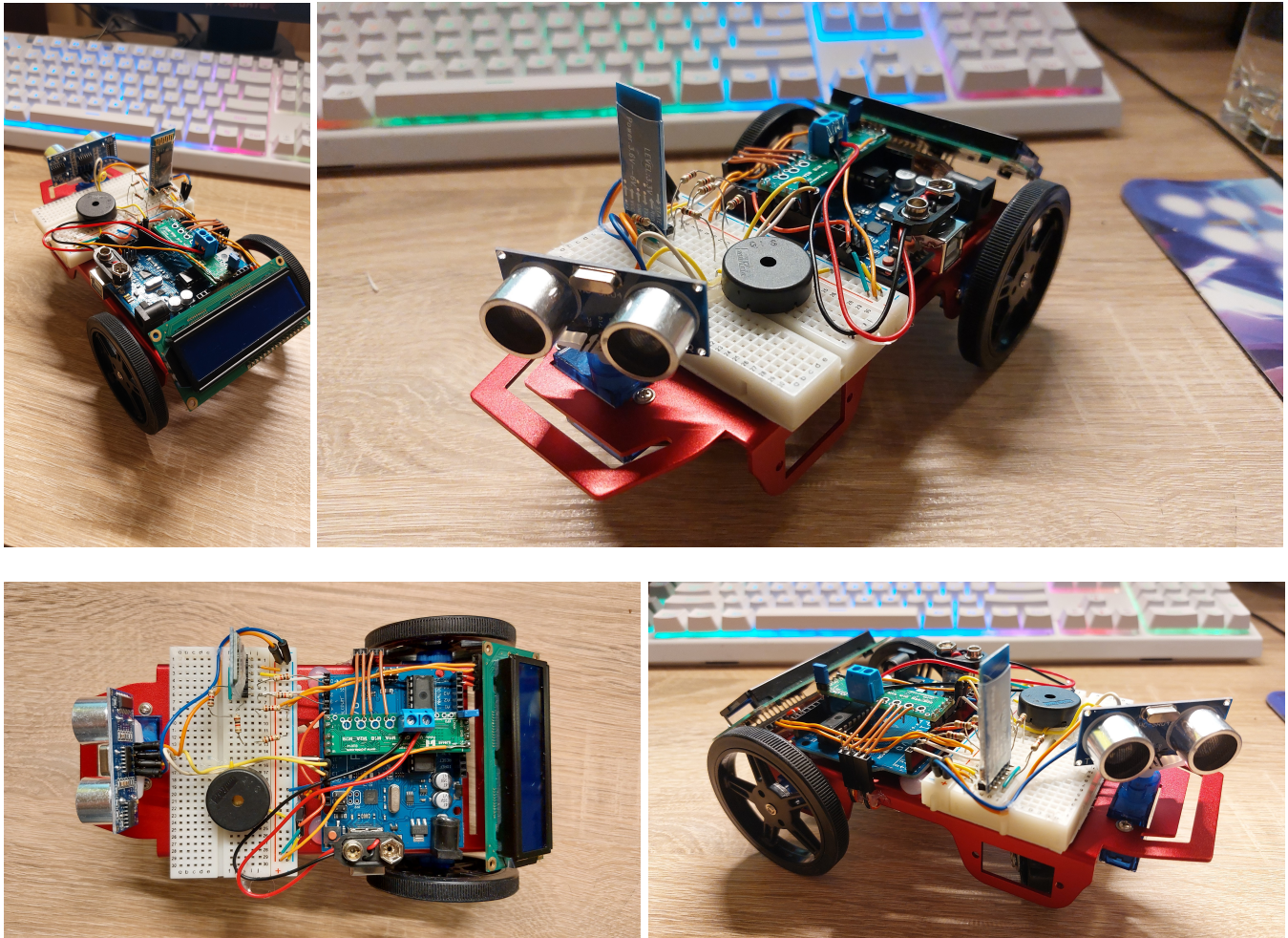
Lista de piese:

- [Kit masina 2WD](#) - contine sasiu, 2 roti spate cu motoare, 1 roata fata simpla, de tip carucior de cumparaturi
- [Placa compatibila Arduino UNO](#)
- [Senzor de distanta cu ultrasunete HC-SR04](#)
- [LCD cu interfata I2C](#)
- [Shield cu Driver de Motor Dual A4990](#)
- [Modul bluetooth HC-05](#)
- [Passive Buzzer](#)

Schema de conectare este urmatoarea:



Proiectul asamblat arata asa:



Din motive de spatiu si pozitionare a pieselor, pinii folositi nu sunt aceeasi cu cei din schema de mai sus. De asemenea, pinii HC-05 functioneaza la 3.3V, nu 5V, asa ca legatura HC-05 ↔ Arduino este facuta cu un divizor de tensiune (un pin HC-05 este conectat la Arduino prin rezistor 1K si la GND prin rezistor 2K). Acest lucru nu este necesar la pinii de iesire ai modulului, deoarece Arduino inregistreaza 3.3V pe un pin digital ca 1 logic fara probleme.

Diferentele dintre schema de mai sus si hardware sunt urmatoarele:

- KEY al HC-05 → 4
- TRIG, ECHO ale HC-SR04 → 12 respectiv 13
- Buzzer → ~11
- Servo → ~5

Pentru a extrage VCC (din moment ce scutul cu motor driver acopera acest pin), am lipit un fir de VCC-ul scutului, care apoi alimenteaza celelalte componente. Arduino este alimentat direct de la scut, de la bateria de 9V, deoarece scutul are optiunea de a scurta intre VIN (input baterie) si VOUT (output catre Arduino VIN). Acest scurt este activat printr-un jumper, care trebuie scos atunci cand Arduino este conectat la USB (pentru a nu fi dublu alimentat).

## Software Design

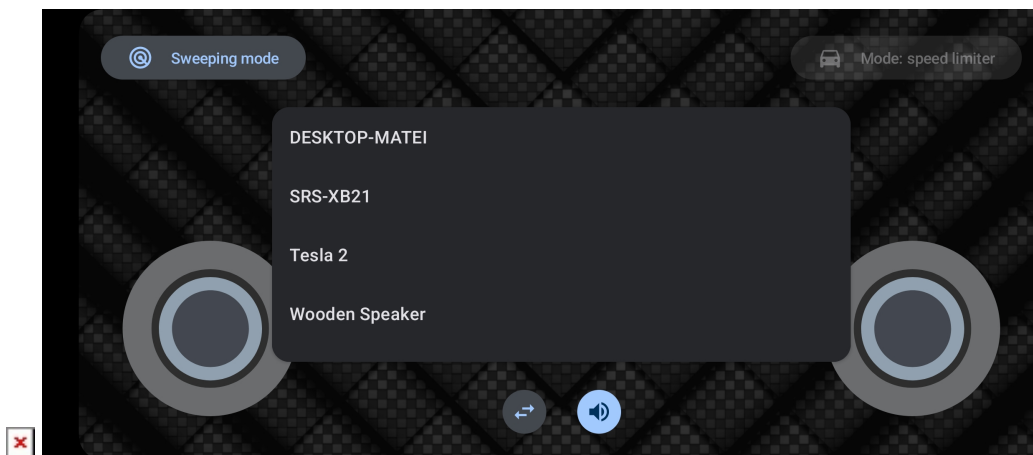
Software-ul se imparte in 2 categorii:

1. Aplicatia mobila - dezvoltat in Android Studio
2. Codul pentru microcontroller-ul Arduino - dezvoltat in Arduino IDE

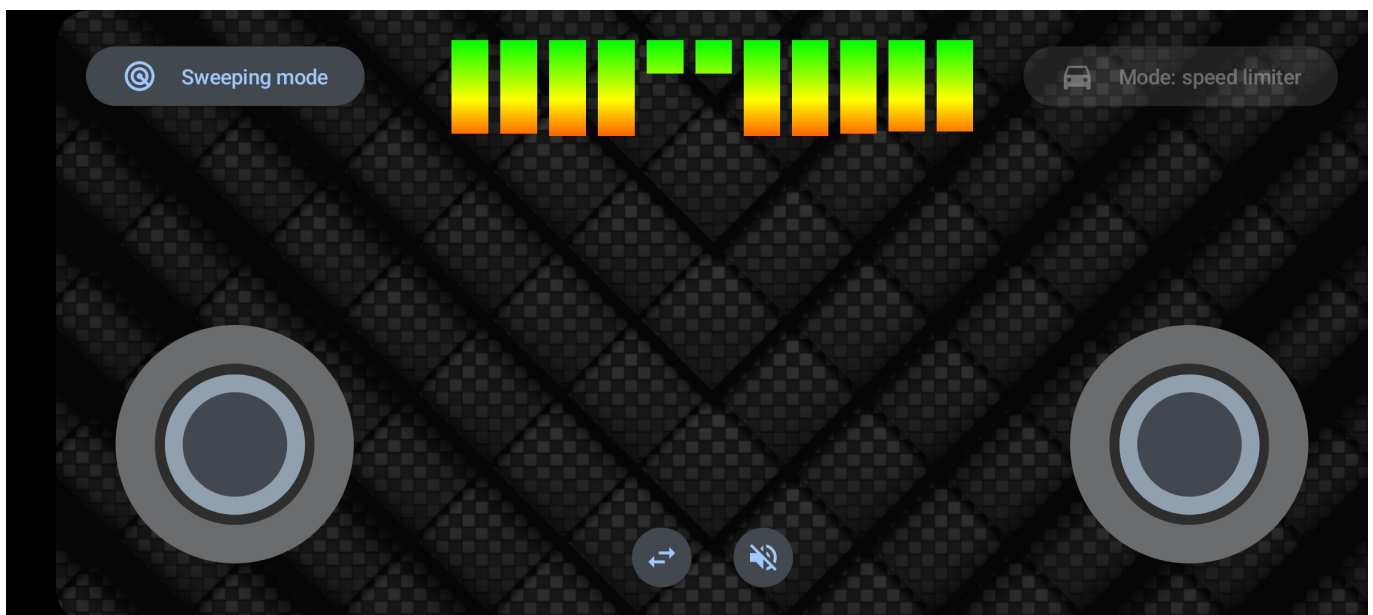
## Rezultate Obținute

In repository, puteti gasi [cateva videoclipuri](#) care demonstreaza functionalitatea proiectului. Functionalitatile, in detaliu, sunt:

- Masina controlabila de pe telefon (Bluetooth) - se face pairing din setarile telefonului, apoi se deschide aplicatia si se alege masina din lista



- Din UI, se poate controla masina cu cele 2 Joystick-uri, iar in partea de sus se observa distanta pana la obstacole. Se mai pot controla: sunetul (On/Off), Sweeping Mode (rotire senzor distanta constanta stanga-dreapta), toggle intre Auto-breaking si Obstacle avoidance (care este disponibil doar daca este activat si Sweeping mode, pentru a putea decide in ce directie sa vireze).



- Atunci cand masina se apropie de un obstacol, isi reduce automat viteza, proportional cu distanta pana la obstacol, pentru a evita lovirea. Simultan, masina emite beep-uri, pentru a semnala distanta pana la obstacol, similar unor senzori de parcare. Distanța pana la obstacole se poate vedea si pe aplicatia mobila.



- Daca "obstacle avoidance" este activat, masina, pe langa cele de mai sus, va incerca sa ocoleasca obstacolele, prin viraj, inainte de a frana (daca nu este posibil, tot se va opri). Virajul se face pe acolo unde observa ca nu sunt obstacole.
- Autopilot - prin mentinerea Joystick-ului de viteza in pozitia inainte, masina va merge inainte, ocolind obstacolele, oprindu-se doar daca se blocheaza (e.g. colt), caz in care trebuie interventie manuala, deoarece nu va da automat cu spatele

## Concluzii

In acest proiect, am invatat cum functioneaza comunicarea Bluetooth seriala, cum se pot comunica astfel date intre un microcontroller si un telefon, dar si multe altele despre motoare, timere, PWM si lipirea firelor/pieselor.

Unui din lucrurile pe care le-am invatat din acest proiect este cat de multe probleme neasteptate pot aparea in cadrul unor task-uri, chiar si simple:

- daca primele 3 librarii de LCD nu au mers, nu este LCD-ul stricat, ci a 4a va merge
- aparent, un fir de tip breadboard se poate rupe *in interiorul izolatiei*, invizibil din exterior, si sa faca apoi contact intermitent, ceea ce duce la foarte multe ore de depanare
- un senzorul de distanta stricat face un feedback-loop cu el insusi, indicand permanent 3-4cm, mai putin daca este pusa o bucata de scotch intre tranzmitator/receptor
- 2 motoare identice, la aceeasi tensiune, se vor invarti la viteze diferite
- atunci cand modulul Bluetooth comunica cu Arduino prin UART, Arduino IDE se va bloca cand incearca sa faca upload, deoarece foloseste tot UART
- orice librarie foloseste timer 1, ceea ce duce la interferenta intre librarii; pentru orice librarie exista o varianta care foloseste timer 2 sau niciunul

## Download

Intreg proiectul se gaseste pe [github](#). De asemenea, aici sunt si arhive ale codului:

Codul sursa Arduino: [tesla2-arduino.zip](#)

Codul sursa Android: [tesla2-android.zip](#)

## Bibliografie/Resurse

Librarii folosite pentru Arduino:

- pentru LCD I2C - [HD44780](#)
- pentru controlare servo folosind timer2 - [ServoTimer2](#)
- pentru driver-ul de motor A4990 - [A4990 Motor Shield](#)
- pentru senzorul de distanta HC-SR04 = [NewPing](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/iotelea/tesla-2>

Last update: **2023/05/30 01:05**

