

Singing car

Autor: Georgescu Ioana

Introducere


Proiectul constă în realizarea unei mașinuțe comandate prin Bluetooth. Mașinuța va putea merge înainte, înapoi, în stânga și în dreapta. Va putea "cânta" prin intermediul unui buzzer și a unor fișiere în care vor fi codificate cântecele. Utilizatorul va putea controla mișcarea mașinuței și, în același timp, va putea porni / opri cântatul și schimba cântecul.

Scopul proiectului este exersarea interacțiunii cu componentele prezentate la curs și în laborator prin construirea unei aplicații distractive.

Utilitatea proiectului vine din perifericele diverse care vor fi conectate la plăcuța Arduino:

- primirea de comenzi prin Bluetooth
- citirea cântecelor din fișiere stocate pe un card Micro SD
- redarea melodiilor folosind un buzzer și funcția tone()
- controlarea nivelului de luminozitate a unui LED folosind PWM
- interacțiunea cu un driver de motoare și controlarea mai multor motoare DC
- lucru cu funcții și biblioteci Arduino

Descriere generală

Pentru interacțiunea cu mașinuța utilizatorul va folosi aplicația Dabble, special concepută pentru a  comanda Arduino prin Bluetooth. Comenzile se vor trimite din modulul Gamepad, modul Digital al aplicației:

- săgețile: controlul direcției mișcării
- SELECT: oprirea cântatului
- START: pornirea cântatului
- SQUARE: cântecul anterior
- CIRCLE: cântecul următor
- TRIANGLE: accelerează
- CROSS: încetinește

Mașinuța are 6 LED-uri: 2 galbene în față, 2 portocalii în spate, unul albastru și unul roșu pe partea de sus.

Pe parcursul setup-ului LED-urile albastru și roșu luminează cu aceeași intensitate și mașinuța nu reacționează la comenzile utilizatorului.

La finalul setup-ului, LED-ul roșu se stinge și cel albastru luminează la intensitate maximă. Acum mașinuța este gata să primească comenzi.

Mașinuța caută cântecele în fișiere cu nume "songX.txt" (X = numărul cântecului) în folderul "songs" de pe cardul SD. În timpul setup-ului, mașinuța emite o atenționare în oricare dintre următoarele situații:

- nu este introdus cardul SD
- cardul SD nu conține folderul "songs"
- folderul "songs" este gol sau conține fișiere necorespunzătoare

Atenționarea constă în stingerea LED-ului albastru, aprinderea celui roșu la intensitate maximă și emiterea unui sunet. Dacă s-a emis atenționarea, mașinuța va recepționa în continuare comenzile pentru mers și reglarea vitezei, dar le va ignora pe cele pentru cântat.

La mersul înainte sau înapoi toate LED-urile "far" (cele galbene și cele portocalii) sunt aprinse. La cotire mașinuța semnalizează clipind farurile de pe partea spre care cotește și tinându-le pe celelalte două aprinse constant. Când mașinuța stă pe loc toate farurile sunt stinse.

Deplasarea va avea loc atât timp cât utilizatorul ține apăsată săgeata corespunzătoare direcției dorite. Dacă se apasă doua sau mai multe direcții simultan, mașinuța va răspunde doar la una. Ordinea de prioritate dată direcțiilor, de la cea mai mare la cea mai mică, este: față, spate, stânga, dreapta.

La apăsarea simultană a accelerării și a frânei (TRIANGLE, respectiv CROSS) ambele comenzi sunt ignorate. Viteza poate fi reglată și cât timp mașinuța stă pe loc. În acest caz, noua viteză va fi viteza cu care va porni mașina.

La apăsarea butonului START mașinuța va începe să cânte. Apăsarea butonului START nu va avea niciun efect cât timp mașinuța cântă. Butonul SELECT urmează aceeași logică, dar pentru oprirea cântatului.

Mașinuța nu schimbă cântecul decât dacă îi cere utilizatorul prin SQUARE sau CIRCLE. Când ajunge la finalul cântecului curent începe să-l cânte din nou de la început. Mașinuța nu ține minte unde a rămas în cardul unui cântec dacă este întreruptă din cântatul lui. Dacă se apasă SELECT în timpul unui cântec și apoi se apasă START, mașinuța va lua cântecul de la început. Dacă se selectează alt cântec și apoi se revine la cântecul întrerupt, mașinuța îl va lua de la început. Dacă există un singur cântec, la apăsarea SQUARE sau CIRCLE mașinuța îl va lua de la început.

Se poate selecta alt cântec și când mașinuța nu cântă. În timpul setup-ului mașinuța numără cântecele găsite și va cicleaza prin ele la apăsarea SQUARE sau CIRCLE. Dacă este la primul cântec și se apasă SQUARE, va trece la ultimul. Dacă este la ultimul și se apasă CIRCLE, va trece la primul. Dacă se apasă SQUARE și CIRCLE simultan, ambele comenzi sunt ignorate.

În timpul cântatului, LED-urile albastru și roșu luminează în tandem cu melodia, cu intensități corelate cu notele cântate, asemănător cu luminile de poliție (când unul luminează mai intens, celălalt luminează mai slab). Când mașinuța nu cântă, doar LED-ul albastru luminează.

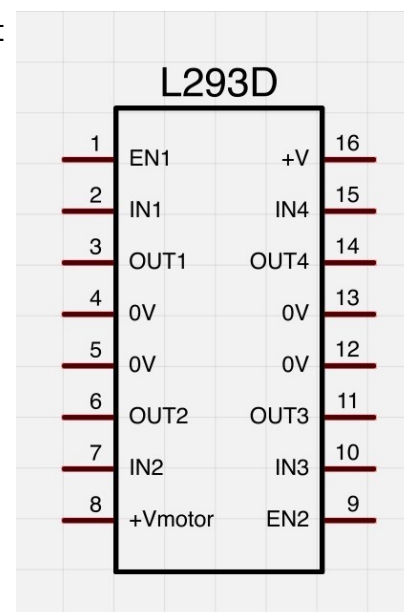


Hardware Design

Nume componentă	Cantitate	Valori	Link magazin
Placă de dezvoltare compatibilă Arduino UNO R3	1		ardushop
Modul Bluetooth HC-05	1		optimusdigital
Modul card slot compatibil cu MicroSD	1		optimusdigital
Driver motoare L293D	1		optimusdigital
Buzzer pasiv	1	5V	ardushop
LED 5mm albastru	1	2.48 ~ 3.7V	ardushop
LED 5mm roșu	1	1.63 ~ 2.03V	
LED 5mm galben	2	2.10 ~ 2.18V	
LED 5mm portocaliu	2	2.03 ~ 2.10V	
Card MicroSD SDHC	1	16GB	altex
Motor cu reductor	4	3V ~ 6V DC	optimusdigital
Roată	4		
Cadru mașinuță	1		
Baterie	2+	9V	emag
Mufă baterie	2+		ardushop
Rezistențe	1	1K	ardushop
	1	5K	
	2	200K	
Tranzistor NPN BC337	1		emag
Tranzistor PNP BC327	1		emag
Breadboard	1		ardushop
Fire mamă-tată			
Fire tată-tată			
Fire mamă-mamă			
Bandă izolatoare			



Driver-ul L293D are 16 pini și poate controla 2 motoare. Eu l-am folosit pentru a comanda 4 motoare, legându-le în paralel. Am conectat pinul 16 la Arduino și pinul 8 la 2 baterii de 9V legate în paralel. O singură baterie nu a fost destul de puternică pentru a putea mișca mașina.



Viteza este controlată prin modificarea valorilor trimise pe pinii de EN (1 și 9), conectați la pini PWM.

Direcția este controlată prin valorile transmise pinilor IN. Fiecare pereche de motoare are 2 pini de input care accepta valori HIGH sau LOW:

- pentru motorele din stânga: IN1 pe HIGH = mers înainte, IN2 pe HIGH = mers înapoi
- pentru motorele din dreapta: IN3 pe HIGH = mers înainte, IN4 pe HIGH = mers înapoi

Software Design

Biblioteci necesare

1. SPI
2. SD
3. Dabble

Proiectul a fost dezvoltat în Arduino IDE. Conține un fișier .ino și două biblioteci cu funcții ajutătoare.

Împărțirea funcționalităților în fișiere:

- **SingingCar.ino**: fluxul principal al programului și comunicarea cu cardul SD
- **Biblioteca DRIVING**: fișierele Driving.h și Driving.cpp; funcțiile, variabilele și constantele necesare pentru deplasarea mașinii și reglarea vitezei
- **Biblioteca SINGING**: fișierele Singing.h și Singing.cpp; funcțiile, variabilele și constantele necesare pentru redarea cântecelor

Pentru lucrul cu cardul SD am folosit biblioteca SD. Aceasta a impus limitarea că fișierele trebuie să aibă nume de maxim 8 caractere (excluzând extensia) și extensii de 3 caractere, altfel nu vor fi văzute. M-am folosit de asta pentru a reține doar numărul cântecului în loc de întregul nume. Compun numele în cod prin concatenare de String-uri.

Formatul unui fișier de cântec este următorul:

```
<tempo> <frecvența maximă>
<frecvența notei 1> <durata notei 1>
<frecvența notei 2> <durata notei 2>
...
-1
```

Frecvența maximă este folosită pentru a calcula valoarea ce trebuie trimisă LED-urilor roșu și albastru. Un simplu apel al funcției map() pe frecvența notei curente nu era destul de satisfăcător deoarece unele cântece au frecvențe cu valori apropiate și nu se observau destul de bine fluctuațiile LED-urilor. Funcția folosită a fost $(\text{frecvențaCurentă}^4) / (\text{frecvențaMaximă}^4) * 255$.

La fiecare iterație a loop() se verifică dacă s-a primit o comandă de la Dabble, se evaluează viteza, se verifică selectarea unui alt cântec, se citește din fișier și se cântă nota curentă, dacă e cazul, și se evaluează direcția de mers.

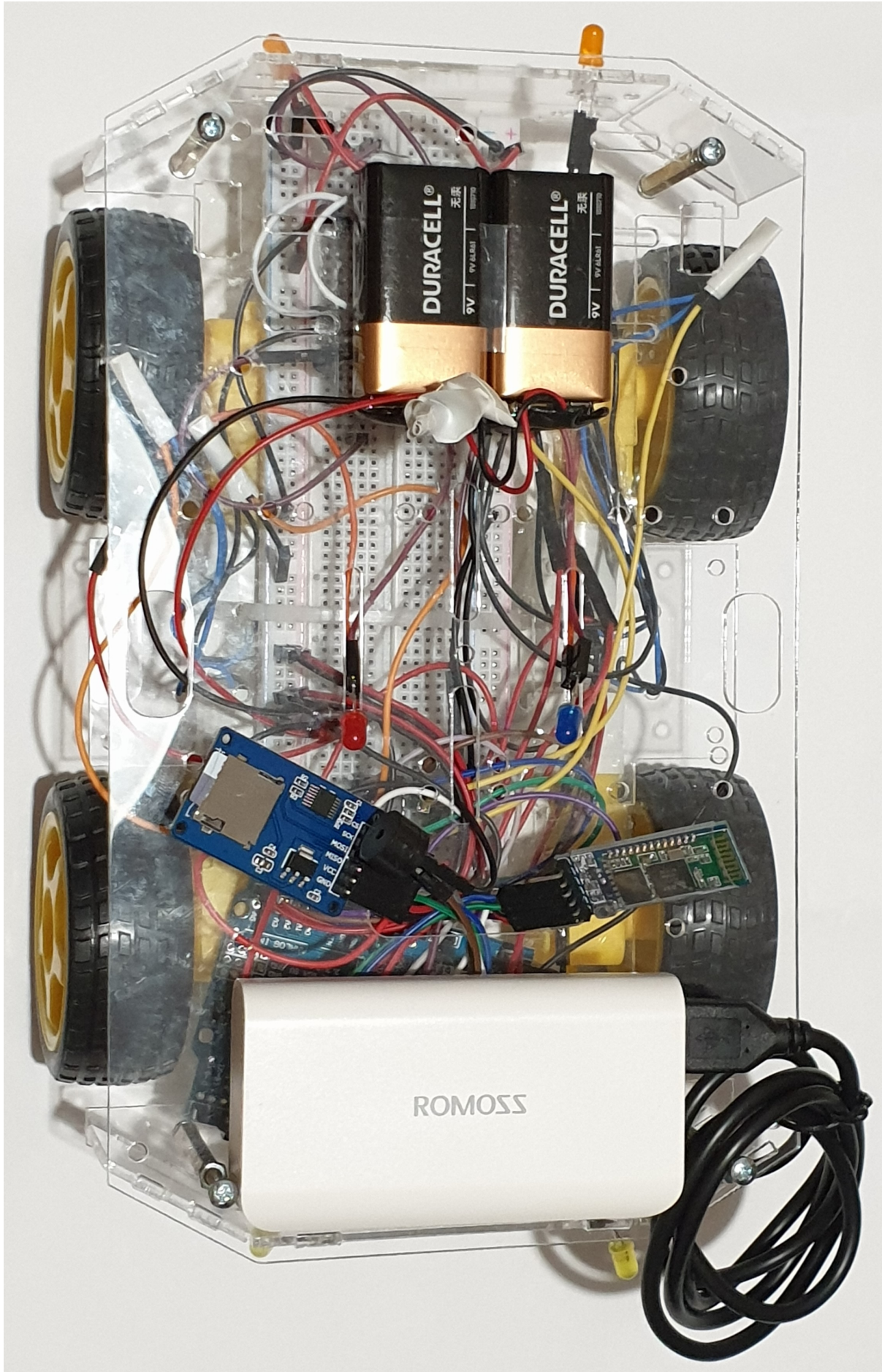
De aceea a fost nevoie de un caracter special pentru a marca sfârșitul unui cântec. Am ales -1 deoarece este o valoare imposibilă pentru frecvență, dar poate fi citită ca un întreg, la fel ca celelalte valori din fișier.

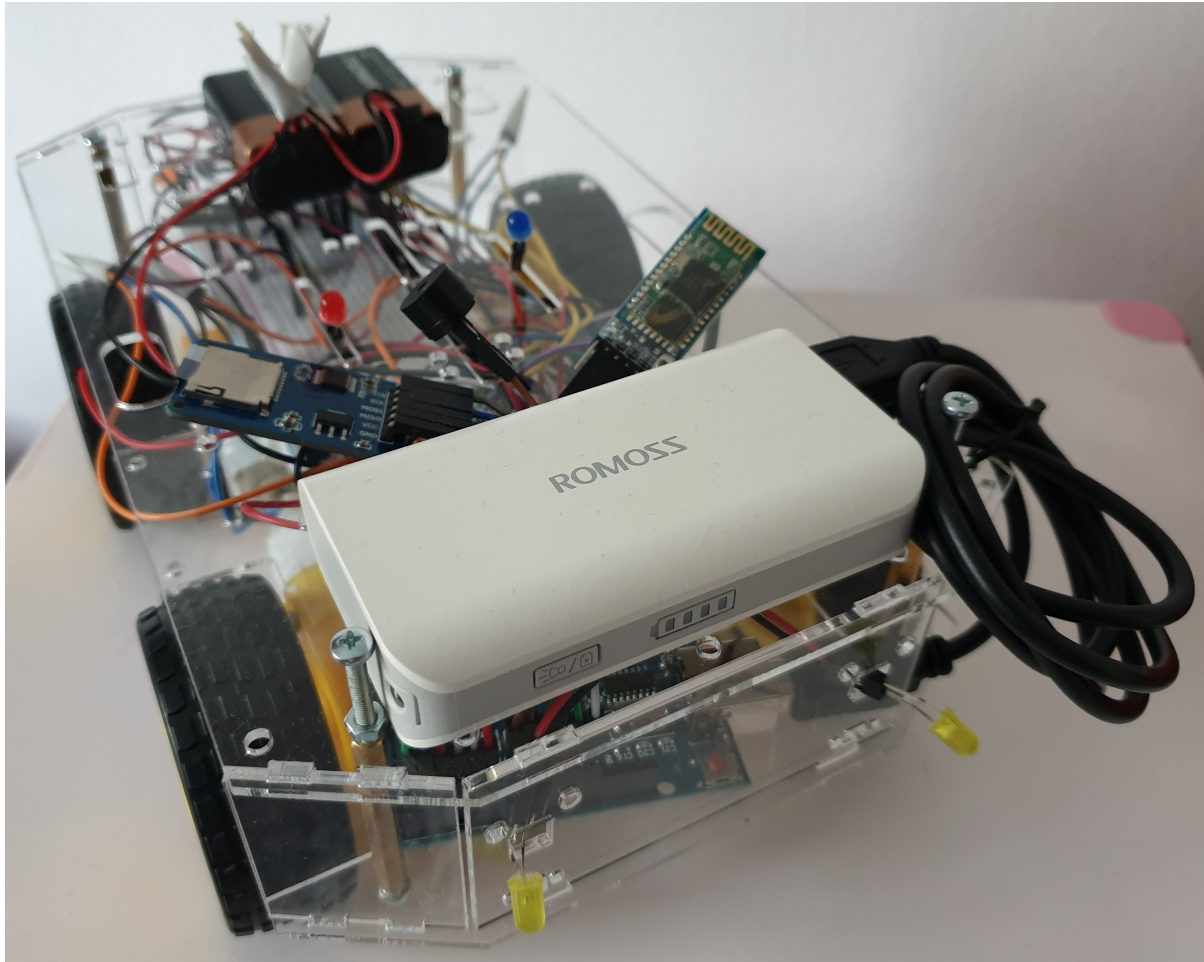
Melodiile, codificarea notelor și a duratelor și calculul duratei unei note întregi au fost preluate din proiectul marcat Cântece Arduino din secțiunea Bibliografie/Resurse.

Pentru a delimita apăsările butoanelor de selectare a unui cântec am considerat ca nu vor exista apăsări distincte la mai puțin de 500 de milisecunde.

Rezultate Obținute

Mașinuța acceptă cântece din folderul songs de pe cardul SD care sunt numite "songX.txt", unde X este numărul cântecului. Poate funcționa și cu card și fără și poate cânta și dacă bateriile pentru motoare sunt deconectate.





[Conectarea buzzerului și a LED-urilor roșu și albastru](#)

[Conectarea modului Bluetooth și a driverului L293D](#)

[Conectarea motoarelor și a roților la driver](#)

[Montajul electric](#)

[Teste filmate](#)

Videoclipul de prezentare

[Prezentare PM Fair \(pptx\)](#)

Concluzii

Un proiect foarte interesant, la care am avut de căutat și de adaptat diverse soluții pentru probleme clasice de Arduino. M-am distrat făcându-l și obținerea rezultatului final mi-a oferit o satisfacție foarte mare.

Dificultăți la care nu m-am așteptat:

- Kit-ul pentru mașinuță nu conținea toate piesele necesare atașării motoarelor și a trebuit eu sa

improvizez un sistem de prindere, tăind bucăți dintr-o șipcă de plastic în unghi de 90° și înșurubându-le de motoare și șasiu

- Nu merge să dai găuri în plexiglass pentru că se sparge. Găurile cu care a venit sunt singurele pe care le poate avea
- Poate fi destul de greu să pui toate firele și să te asiguri că nu se fac scurtcircuite și că fac toate contact unde trebuie pe breadboard. Sau dacă s-a smuls sau a ieșit un fir să-ti dai seama de unde a ieșit. Mai ales dacă ai înșurubat deja șasiul și ai presat firele ca să încapă în șasiu și trebuie să rearanjezi firele cu penseta printre plăcile de plastic. Am ars un LED
- Circuitul final consumă mult mai mult decât ma așteptam. A fost nevoie de 2 baterii pentru motoare și de o sursă separată pentru Arduino. Pentru Arduino am folosit o baterie externă mai veche și cablul USB prin care se conecta plăcuța la calculator
- Memoria este limitată. Trebuie găsit un echilibru între variabilele globale și cele locale. Dacă sistemul rămâne fără memorie în timpul rulării își pierde din funcționalități și începe să se comporte bizar. În prezent, mașinuța merge bine cu 62% din memoria dinamică ocupată cu variabile globale și 771 de bytes liberi pentru cele locale
- Linoleumul de pe hol este mai denivelat și mai alunecos decât credeam și mașinuța se împotmolește pe el din când în când

Download

- Pagina în format PDF: [singing_car.pdf](#)
- Codul și cântecele: [singing_car.zip](#)

Formatul arhivei singing_car.zip:

- SingingCar: codul sursă
- songs: fișierele cu cântece și notes_and_songs.txt, un fișier cu corespondența dintre frecvențe și note și cu numele cântecelor din fiecare fișier

Jurnal

19-20.04.2020

- Căutare tutoriale
 - Căutare piese (modul Bluetooth, modul card SD, scut driver motoare L293D)
-

21.04.2020

- Comunicare temă
 - Comandare piese
-

23.04.2020

- Sosire piese
 - Scoatere L239D de pe scut
 - Conectare orientativă cu Arduino și breadboard ⇒ loc numai pentru 3 pin-uri de LED-uri ⇒ grupare LED-uri
-

25.04.2020

- Achiziționare, formatare și testare card MicroSD
 - Testare buzzer și LED-uri pentru melodie; conectare 2 LED-uri la 1 singur pin ⇒ intensitate prea scăzută
 - Testare alimentare LED-uri de la baterie de 9V ⇒ intensitate satisfăcătoare
 - Comandare tranzistori + mai multe fire
-

16.05.2020

- Testare buzzer și LED-uri pentru melodie cu tranzistoare ⇒ rezultate mulțumitoare
 - Testare conectare la Dabble, primire comenzi și comandare motoare
-

20.05.2020

- Realizarea tuturor conexiunilor pe breadboard
 - Realizarea unui schelet de cod și testarea lui cu placa Arduino alimentată de la baterie
 - Realizarea schemei electrice în EAGLE
-

22.05.2020 - 24.05.2020

- Scrierea și testarea codului pentru cântat
 - Scrierea fișierelor de cântece
-

25.05.2020 - 26.05.2020

- Scrierea și testarea codului pentru condus
-

27.05.2020

- Testare funcționalități cântat și condus simultan
- Alimentare Arduino și motoare de la aceeași baterie de 9V
- Sistemul se comportă bizar. Se aude un țiuit continuu și toate farurile stau aprinse. Roțile nu încep să se învârtă decât dacă le învârt eu cu mâna puțin

28.05.2020 - 29.05.2020

- Testare funcționalități cântat și condus simultan
 - Alimentare Arduino de la calculator sau din priză și motoarele de la o baterie de 9V nou-nouță
 - Sistemul se comportă mai bine. Încă are momente în care nu merge bine. Dacă adaug prea multe cântece nu mai merge. Posibil să fie de la memorie
-

01.06.2020 - 02.06.2020

- Testare funcționalități cântat și condus simultan
 - Optimizare cod: constante de tipul cel mai mic posibil (int care nu depășesc 255 ⇒ unsigned char), în loc de a reține numele fișierelor într-un vector de String am reținut doar numărul și am compus numele prin concatenarea cu String-uri constante
 - Trecerea de la folosirea map() pentru intensitatea LED-urilor roșu și albastru la funcția numită în secțiunea Software Design
 - Alimentare Arduino de la acumulator și motoarele de la 2 baterii de 9V noi
 - Sistemul se comportă corespunzător. Consumul de memorie a scăzut destul de mult ca sistemul să fie stabil
-

03.06.2020

- Modularizarea codului și adăugarea de comentarii
- Filmarea videoclipului
- Finalizarea paginii proiectului

Bibliografie/Resurse

- Datasheet L203D: <https://www.ti.com/lit/ds/symlink/l293.pdf>
- Specificații HC-05: <https://components101.com/wireless/hc-05-bluetooth-module>
- Specificații modul card SD: https://curtocircuito.com.br/datasheet/modulo/cartao_micro_SD.pdf
- Dabble Gamepad: <https://thetempedia.com/docs/dabble/game-pad-module/>
- Cântece Arduino:
<https://create.arduino.cc/projecthub/adithyalokesh17/let-s-make-arduino-sing-with-a-buzzer-ee2e34>
- Conectare card SD: <https://www.instructables.com/Micro-SD-Card-Tutorial/>
- Conectare și control motoare: <https://lastminuteengineers.com/l293d-dc-motor-arduino-tutorial/>
- Comandare motoare cu Dabble:
<https://create.arduino.cc/projecthub/theSTEMpedia/4-wheel-robot-made-with-arduino-controlled-using-dabble-4650f7>
- L293D pe scurt: <https://learn.adafruit.com/adafruit-arduino-lesson-15-dc-motor-reversing/lm293d>
- Citire întregi de pe cardul SD: <https://forum.arduino.cc/t/read-int-values-from-the-sd-card/144766>
- Clipire periodică a LED-urilor fără delay:
<https://www.arduino.cc/en/Tutorial/BuiltInExamples/BlinkWithoutDelay>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/singingcar>



Last update: **2021/06/19 08:26**