

Racing Car

Autor

Matei Vlad Cristian

Grupa: 331CC

Introducere

Scopul proiectului este realizarea unei masinute ce poate fi controlata printr-o aplicatie mobila android. De-a lungul timpului am avut mai multe masinute, dar fiecare dintre acestea putea fi controlata printr-o telecomanda. Lipsa unei baterii sau pierderea telecomenzii a reprezentat de multe ori un motiv suficient pentru a nu mai folosi masinuta. In cadrul acestui proiect, conectarea se va realiza cu ajutorul unui modul de Bluetooth. Masinuta va dispune de leduri pentru a putea simula semnalizarea, stopurile si farurile unei masini. De asemenea, va avea lumini suplimentare ce vor putea fi aprinse pentru modul de noapte si va dispune de un mini sistem audio.

Descriere generală

Bateria de 9V va alimenta placuta Arduino. Aplicatia dezvoltata android se va conecta la masinuta, prin modulul Bluetooth HC-05 ce este legat la placuta Arduino UNO. Utilizatorul va putea trimite din aplicatie informatiile urmatoare masinutei: unghiul rotatiei volanului care va trimite datele catre driver motor, care la randul sau va porni in directia potrivita unul dintre motorasele DC, cel responsabil de stabilirea directiei, aprinderea semnalului stanga, dreapta, sau ambele (avarii) ce determina stingerea sau aprinderea de led-uri, apasarea pedalei de frana sau de acceleratie ce va determina driver-ul de motor sa porneasca motorasul responsabil de tractiune, pornirea modului de noapte si pornirea mini sistemului audio prin activarea speaker-ului.

Schema bloc



Hardware Design

Lista de componente:

- Arduino Uno
- Baterie externa 5V
- Acumulator 9V - 2 buc
- Fire de conexiune
- Rezistente 1ohm - 8 buc
- Leduri - 10 buc
- Modul Bluetooth HC-05
- Driver Motor L298N
- DC Motor small 3-6V - 2 buc
- Speaker 36mm 0.5W 0.8ohm
- Conector baterie 9V

Schema electrica



Software Design

Mediu de dezvoltare: Arduino IDE 1.8.20.

Aplicatie Android

Aplicatia mobila android a fost dezvoltata in intregime de mine. In implementarea ei am folosit o biblioteca care a oferit o interfata de conectare la comunicarea seriala prin bluetooth. Cu ajutorul aplicatiei, utilizatorul se poate conecta la orice dispozitiv bluetooth din apropiere cu care a reusit sa faca pair pana in acel moment. Daca conectarea a reusit, atunci se pot trimite informatii catre placuta arduino, ce detine modulul de bluetooth, in functie de o codificare ale carei reguli le voi explica in continuare. Pentru a putea obtine raspunsuri in timp real, a fost nevoie sa incerc sa reduc informatia trimisa astfel incat am compactat datele in char-uri pe care le voi trimite mai departe.

- Semnal stanga ON/OFF - codificare → (0/6)
- Semnal dreapta ON/OFF - codificare → (1/7)
- Semnal avarii ON/OFF - codificare → (2/8)
- Semnal muzica ON/OFF - codificare → (3/9)
- Semnal pozitii si faruri ON/OFF - codificare → (4/10)
- Semnal lumini de noapte (NEON) ON/OFF - codificare → (5/11)

- Semnal frana / mers cu spatele ON/OFF (viteza fixa) - codificare → (12/13)
- Acceleratie - dinamica/adaptabila - semnal trimis de catre pedala de acceleratie in functie de unghiul apasarii. Poate lua valori de la 0 la 10 - codificarea → (20 → 30)
- Volan - se poate roti 90 de grade la stanga, respectiv 90 de grade la dreapta. Cele 180 de grade sunt mapate in 18 unitati (10 grade = 1 unitate) care sunt codificate rezervand valorile (40 → 58)

Cod Arduino

- Definesc notele muzicale, melodia si initializez variabile precum (tempo, divider, noteDuration, key ..).
- Definesc cod-urile corespunzatoare fiecarui semnal, precum si capetele intervalelor din codificarea Acceleratiei si a Miscarii Volanului pentru a putea interpreta informatia primita de la bluetooth.
- Definesc pinii : am folosit, pentru a ajusta viteza motoraselor, pinii 9,10 cu functia PWM; pentru comunicarea prin bluetooth am folosit chiar pinii 0-1 RX,TX. De asemenea, am configurat modulul de bluetooth sa transmita informatia printr-un baud rate = 115200.
- Functia loop() : verifica daca poate citi un char, daca un char a fost citit, atunci interpreteaza informatia. Daca s-a stabilit pornirea difuzorului, atunci se va apela functia playSong(), altfel se va executa actiunea (celelalte sunt actiuni scurte) si se va iesi din functia loop.
- Functia playSong() face ca functia loop() sa nu se mai apeleze, astfel incat, la fiecare nota cantata, voi verifica daca s-a primit ceva de la bluetooth, in caz afirmativ, interpretarea informatiei primite va avea prioritate.
- Un mic algoritm dezvoltat este cel al stabilirii tensiunii primite de motorasul responsabil de directie, prin PWM, pentru a simula cat mai bine miscarea naturala a volanului. Masinuta avea ceva mici probleme, era putin dezechilibrata in ce priveste directia, de asemenea sistemul de tip "puzzle" dispune de un arc care face ca rotile sa doreasca sa revina intotdeauna in pozitia normala. Astfel, motorasul de directie trebuie sa reziste fortei elastice a arcului. Am incercat sa ofer niste impulsuri mai puternice la inceputul schimbarii directiei si apoi sa revin la o viteza mai redusa doar pentru a-si mentine traiectoria.

Rezultate Obținute

Aplicatia Android



Masinuta



Concluzii

Desi a fost o mica provocare sa pot duce la bun sfarsit proiectul, in cele din urma s-a dovedit a fi o experienta frumoasa si utila. De-a lungul rezolvarii temei am intampinat cateva probleme: configurarea aplicatiei mobile pentru a permite conectarea la bluetooth si imposibilitatea configurarii unui modul de bluetooth care s-a dovedit a fi problematic, fiind nevoit sa il inlocuiesc cu unul care a functionat, in schimb, din prima. Partea de hardware s-a mai evidentiat prin: alegerea bateriilor si a modului de incarcare, lipirea tuturor firelor si repararea sistemului de tractiune al masinutei. In cele din urma, sunt mandru de rezultatul obtinut : o masinuta controlata cu ajutorul telefonului prin bluetooth, care dispune de o baterie externa reincarcabila, putand astfel fi folosita oricand.

Download

<https://github.com/VladuTzZ-sudo/RacingCarArduino>

<https://github.com/VladuTzZ-sudo/RacingCarAndroid>

[song.rar](#)

Jurnal

10.04.2022 - 1.05.2022

- Finalizare aplicatie Android
- Cumparare piese

25.05.2022

- Finalizare software arduino

26.05.2022

- Finalizare parte hardware

27.05.2022

- Documentatie

Bibliografie/Resurse

<https://github.com/harry1453/android-bluetooth-serial>

<https://ocw.cs.pub.ro/courses/pm/lab/lab3-2022>

<https://ocw.cs.pub.ro/courses/pm/lab/lab1-2022>

<https://ocw.cs.pub.ro/courses/pm/lab/lab2-2022>

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

<https://forum.arduino.cc/t/problem-using-l298n-board-to-control-motor-speed/545272/3>

<https://pdf1.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html>

<https://microcontrollerslab.com/l298n-dc-motor-driver-module-arduino-tutorial/>

<https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/>

<https://www.instructables.com/How-to-Configure-HC-05-Bluetooth-Module-As-Master-/>

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/racing-car>



Last update: **2022/05/27 01:28**