

Propeller Car

Autor

Nume: [Alexandra-Ioana Cune](#)

Grupă: 336CC

Introducere

Propeller car este o masinuta controlata cu ajutorul unor elice, prin intermediul unei telecomenzi.

Proiectul este alcatuit din doua parti principale: masinuta propriu-zisa legata la un arduino ce va comunica cu un dispozitiv separat, telecomanda, prin intermediul unui modul radio. Prin telecomanda se vor putea seta directia de mers a masinii cu un joystick si viteza de deplasare (viteza de rotatie a elicelor) cu un potentiometru.

Utilitatea acestui proiect rezida preponderent in faptul ca acesta poate scala. Intregul circuit amplasat pe masina poate fi utilizat si pe alte mijloace de transport, de exemplu o ambarcatiune.

Descriere generală

Functionalitatea de baza:

Masinuta si telecomanda vor fi conectate prin intermediul modului radio. Cele doua componente vor avea cate o sursa de alimentare si cate un Arduino UNO incorporat. Joystick-ul amplasat pe telecomanda va avea efect asupra directiei masinii, el controland servomotorul care va fi legat de roata din fata care va da sensul de mers. Viteza este controlata prin intermediul potentiometeului. Rotind de acesta se va modifica viteza de rotatie a motoarelor cu elice atasate la spatele masinii, ele fiind controlate simultan printr-un Motor Driver L293D.



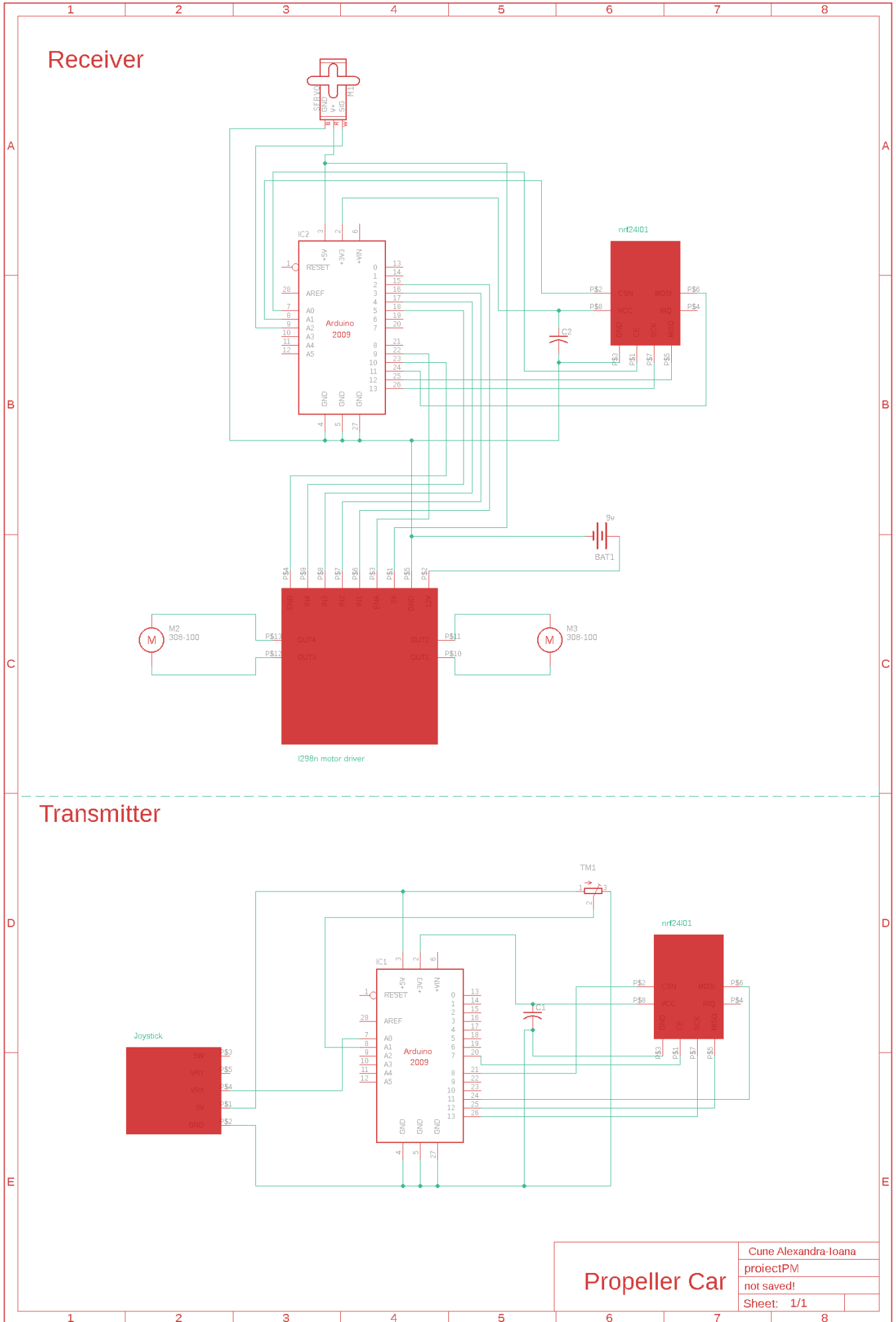
Hardware Design

Lista de piese:

1. Pentru masina:

- Arduino UNO
 - Motor Driver L298N
 - Servomotor
 - Breadboard
 - Motor cu elice X2
 - Roata X3
 - Modul radio nRF24L01
 - Condensator 47 μ F
 - Fire de legatura/ mama-tata
2. Pentru telecomanda:
- Arduino UNO
 - Joystick
 - Potentiometru
 - Modul radio nRF24L01
 - Condensator 47 μ F
 - Fire de legatura/ mama-tata

Schema electrica



Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare Arduino IDE
- librării și surse 3rd-party : SPI.h, nRF24L01.h, RF24.h, Servo.h.
- github: <https://github.com/AlexandraCune/Propeller-Car>

Algoritmul ce sta la baza implementarii acestui proiect este urmatorul:

In transmitter se citesc date de la joystick si potentiometru utilizand functia ***analogRead()*** . Aceste date sunt stocate intr-o structura 'Data_Package'

```
struct Data_Package {  
    int dir = 0;  
    int power = 0;  
};
```

care va fi transmisa prin modulul radio, receiver-ului. Utilizand biblioteca RF24.h, am dispus de numeroase functii de manipulare a semnalelor radio. Astfel, transmitter-ul l-am setat pe modul: ***stopListening***, folosind functia corespunzatoare. In loop() acesta are rolul de a citi datele mentionate mai sus si de a le transmite mai departe prin comanda :

```
radio.write(&data, sizeof(Data_Package));
```

In receiver se seteaza la inceput, in functia de setup, pinul pe care functioneaza servo-ul, starea modulului radio: ***startListening*** , dar si pinii prin care sunt controlate cele 2 motoare si vitezele lor.

In loop se verifica conectivitatea cu celalalt modul radio ***if (radio.available())*** , iar daca acestea pot comunica, receiverul citeste datele trimise si le stocheaza intr-o structura ca cea mentionata mai sus.

```
radio.read(&data, sizeof(Data_Package));
```

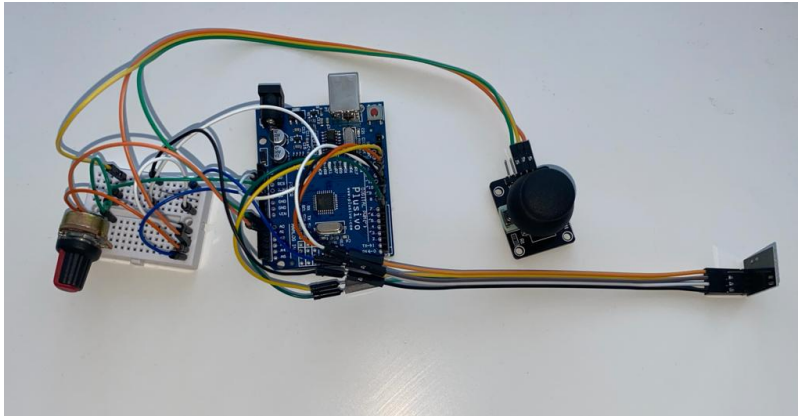
Cu datele obtinute, se modifica prin *data.dir* directia servo motorului, si prin *data.power* viteza de rotatie a motoarelor. Tot aici sunt pornite motoarele si li se atribuie un sens setand cate un pin pe HIGH si celalalt pe LOW.

Rezultate obtinute

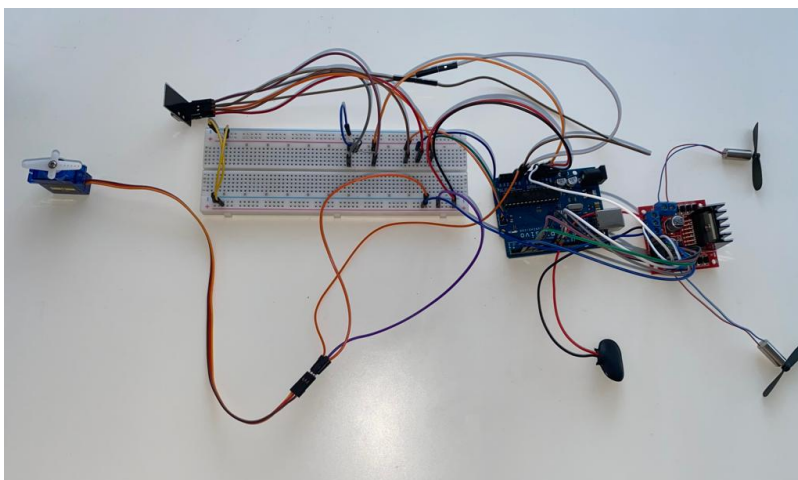
[Link Demo](#)

Proiectul neamplasat pe macheta:

Telecomanda:

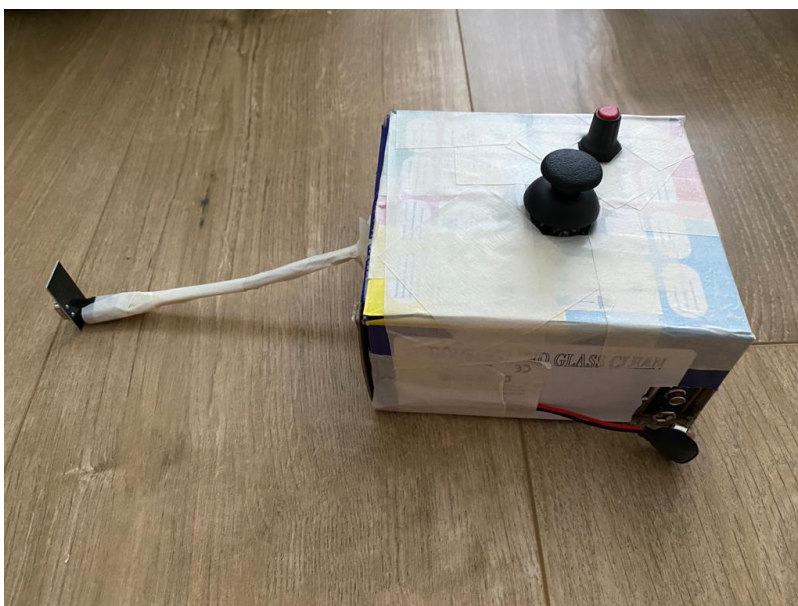


Masina:

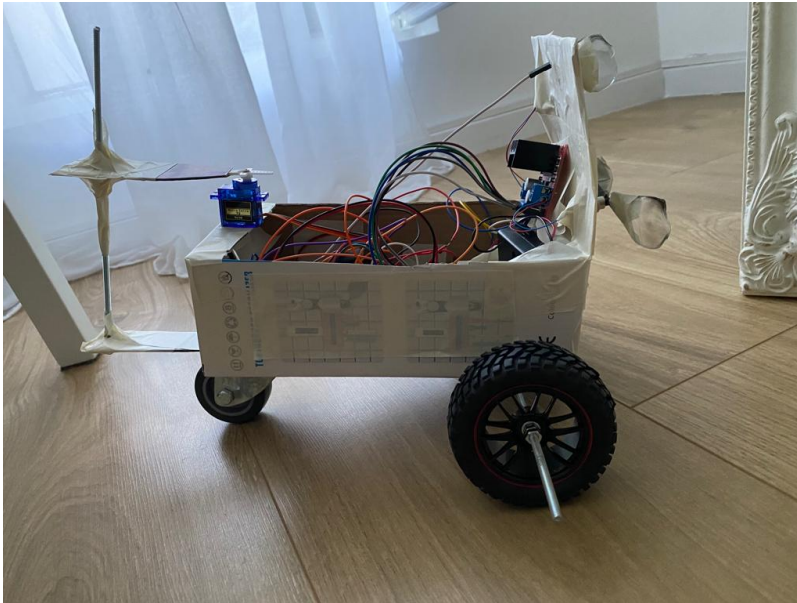


Macheta proiect:

Telecomanda:



Masina:



Download

[Arhiva proiect](#)

Bibliografie/Resurse

- <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>
- <https://create.arduino.cc/projecthub/ryanchan/how-to-use-the-l298n-motor-driver-b124c5>
- <https://circuitdigest.com/microcontroller-projects/dc-motor-speed-control-using-arduino-and-potentiometer>
- <https://www.youtube.com/watch?v=C6Mbs27iFrA>

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/propeller_car



Last update: **2022/06/02 21:52**