

# Masina cu telecomanda □

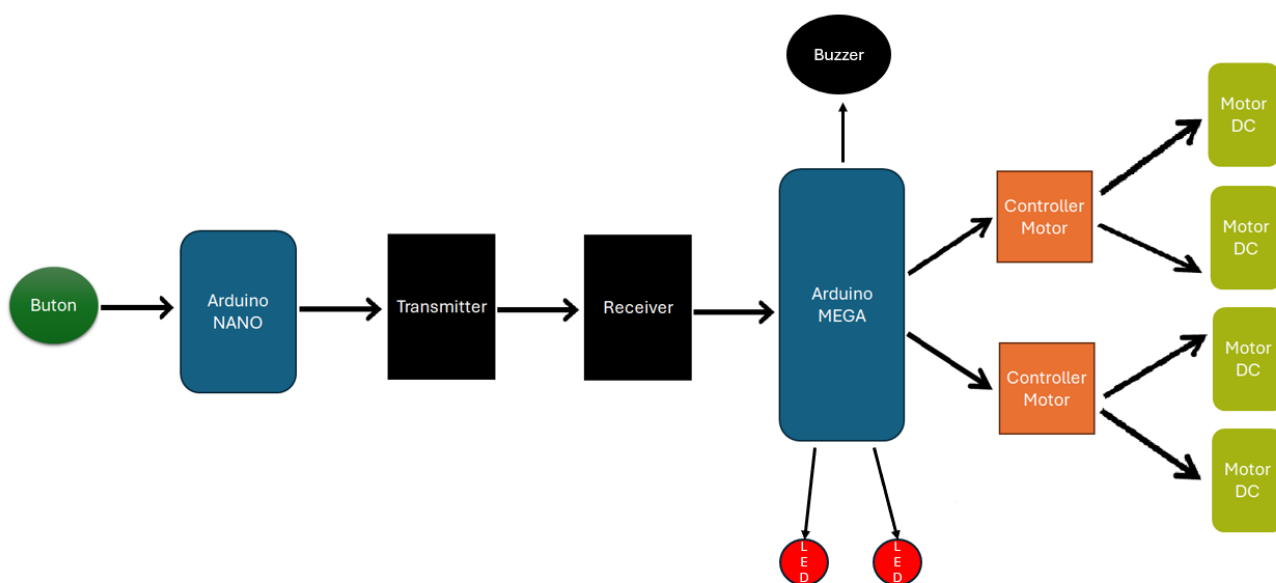
Movileanu Raluca - 331CB

## Introducere

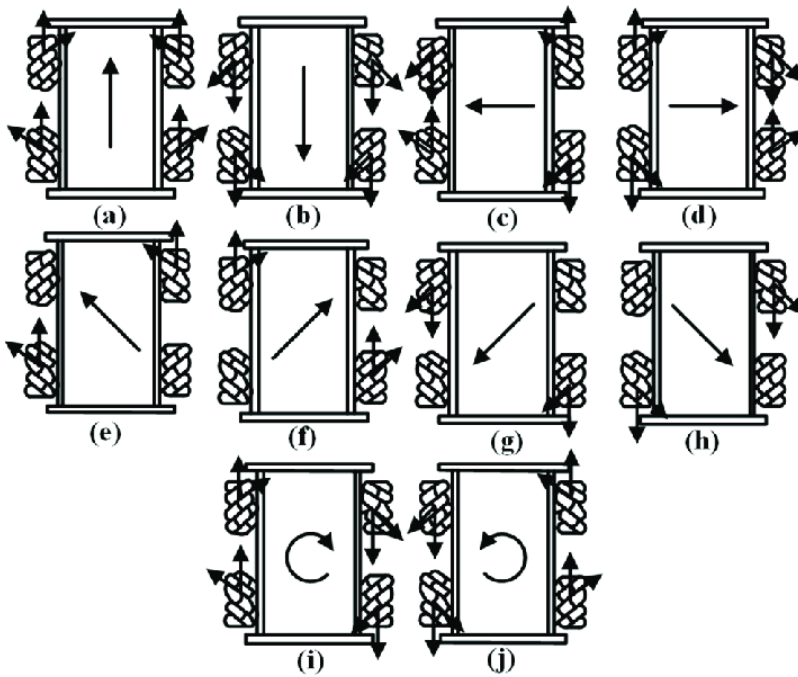
Proiectul consta intr-o masina care primeste comenzi de la o telecomanda, comenzile fiind mesaje cu instructiuni pentru deplasarea masinii: fata, spate, stanga, dreapta, diagonale si rotiri (rotire stanga, rotire dreapta). De asemenea, masina va avea montat si un girofar cu sirena. Scopul proiectului este realizarea unei masini care poate fi comandata de la distanta folosind module radio. Utilitatea sa vine din faptul ca proiectul meu este de fapt o jucarie cu care ma pot distra atat singura cat si cu prietenii.

## Descriere generală

La apasarea unui buton din cele 10 (pentru fata, spate, stanga, dreapta, diagonale, rotire stanga, rotire dreapta) de pe telecomanda, placuta Arduino NANO va transmite un mesaj catre placuta Arduino MEGA de pe masina, prin intermediul transmitterului si al receiverului care comunica radio printr-o adresa data. In functie de mesajul primit de receiver, placuta MEGA va da comenzi controllerelor de motoare, care la randul lor vor actiona motoarele DC in modul dorit de catre utilizator. Voi avea si un buzzer si 2 leduri rosii ce se vor stinge si aprinde la un interval de timp.



- Masina se va deplasa pe urmatoarele directii conform pozei:



## Hardware Design

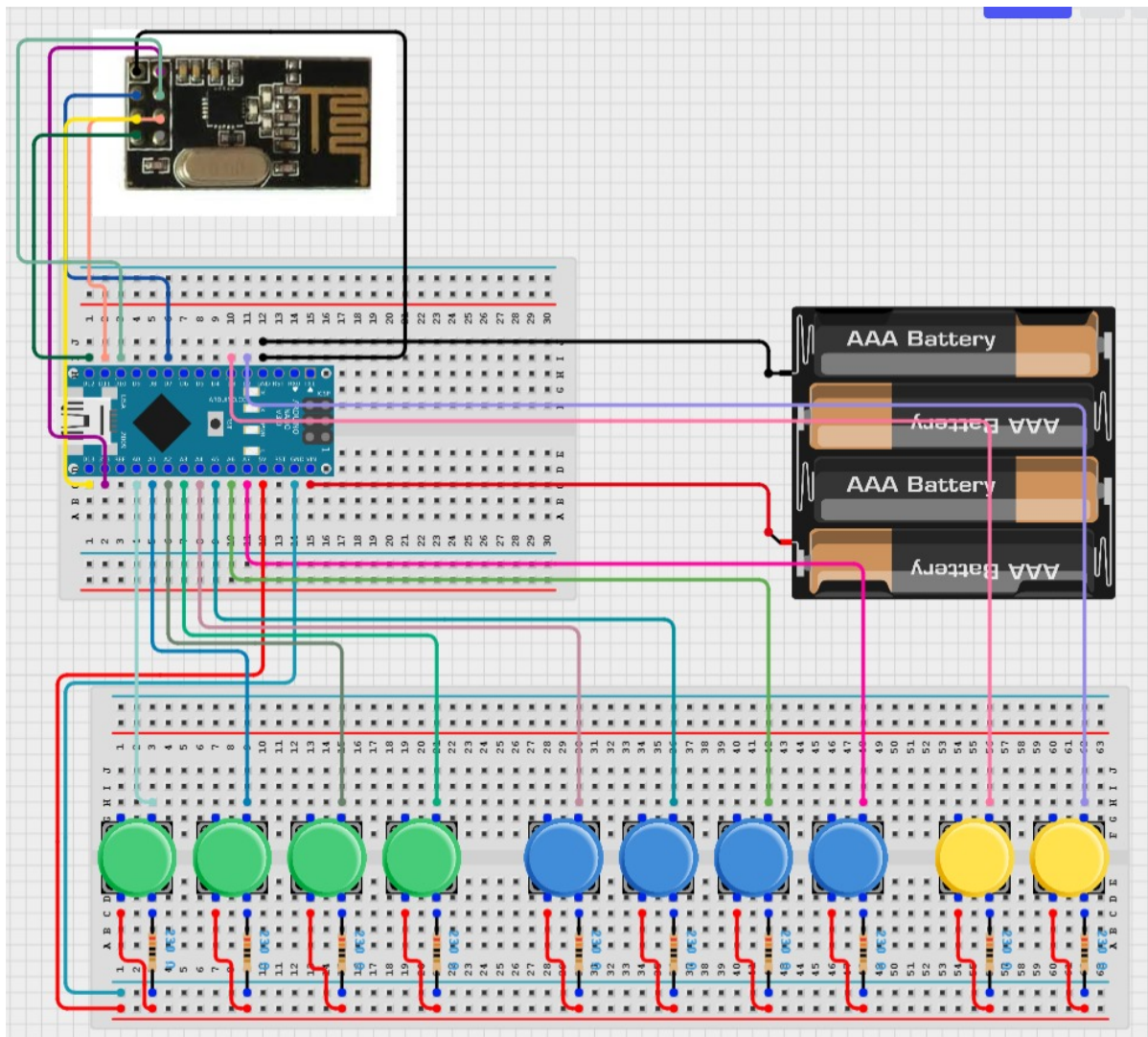
### Lista piese

- 1 X Arduino NANO
- 1 X Arduino MEGA
- 10 X buton
- 4 X roata
- 4 X motor DC 3V (MOTOR CU REDUCTOR 3-6V)
- 2 X controller motor (Modul cu Driver de Motoare Dual L298N)
- 2 X trans-receiver radio (NRF24L01 - 2.4G Wireless Transceiver Module)
- 2 X LED
- 1 X buzzer
- 3 X suport baterii
- 2 X suport baterii 9V
- 1 X breadboard mare
- 1 X breadboard mic
- 1 X breadboard mini
- 13 X rezistente
- fire

Placuta **Arduino NANO** se afla pe telecomanda, ea primeste comenzi de la **butoane** si le transmite mai departe la placuta **Arduino Mega** de pe masina prin intermediul **transmitterului**(de pe telecomanda) si al **receiverului**(de pe masina). Placuta Arduino NANO este alimentata de **4 baterii AA**. Placuta **Arduino MEGA** primeste comanda, trimite semnale la cele doua drivere pentru motoare(

**Modul cu Driver de Motoare Dual L298N**) care actioneaza motoarele(**MOTOR CU REDUCTOR 3-6V**) la o viteza stabilita in cod, aceasta fiind setata folosind pini PWM. Placuta Arduino MEGA este alimentata de o **baterie de 9V** iar driverele sunt alimentate de cate **4 baterii AA**. Pe placuta **Arduino MEGA** se mai afla doua **leduri** si un **buzzer** care alterneaza intre starile aprins si stins, acestea servind rolurile de girofar, respectiv sirena.

## Telecomanda



Arduino NANO:

- A0: conectare buton 1 (fata)
- A1: conectare buton 2 (spate)
- A2: conectare buton 3 (stanga)
- A3: conectare buton 4 (dreapta)
- A4: conectare buton 5 (fata stanga)
- A5: conectare buton 6 (fata dreapta)
- A6: conectare buton 7 (spate stanga)
- A7: conectare buton 8 (spate dreapta)

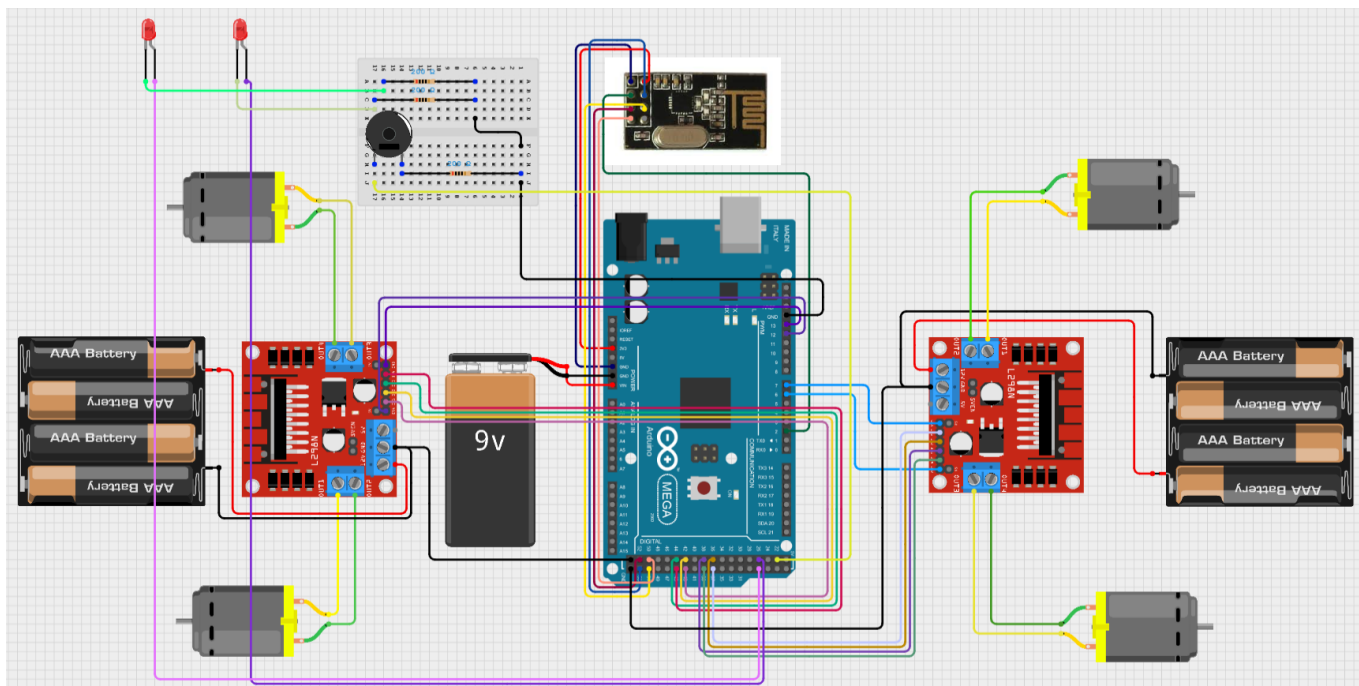
butoanele analogice au fost folosite pentru butoane deoarece nu sunt disponibile destui pini digitali pentru a conecta toate butoanele la digital deoarece modulul nRF24L01 ocupa 5 pini digitali din 12

- D3: conectare buton 9 (rotire stanga)
- D2: conectare buton 10 (rotire dreapta)

am folosit pentru ultimele doua butoane pini digitali deoarece nu au mai fost disponibili pini analogici pentru a conecta asemenea celorlalte butoane de pe telecomanda

- GND: conectare GND de la modulul nRF24L01, firul negru al carcasi de baterii si butoanele
- 3V3: conectare VCC modul nRF24L01
- 5V: conectare butoate
- VIN: conectare fir rosu al carcasi de baterii
- D7: conectare pin CE al modulului nRF24L01
- D10: conectare pin CSN al modulului nRF24L01
- D11: conectare pin MOSI al modulului nRF24L01
- D12: conectare pin MISO al modulului nRF24L01
- D13: conectare pin SCK al modulului nRF24L01

## Masina



### Arduino MEGA

- 3v3: alimentare modul NRF24L01
- GND: pin GND modul NRF24L01, fir negru alimentare placuta Arduino MEGA, fir negru Buzzer si LED-uri rosii, GND drivere motoare L298N
- VIN: alimentare baterie 9V la placuta Arduino MEGA
- D13(PWM): pin ENB de pe driverul de motoare L298N (1)

- D12(PWM): pin ENA de pe driverul de motoare L298N (1)
- D7(PWM): pin ENA de pe driverul de motoare L298N (2)
- D6(PWM): pin ENB de pe driverul de motoare L298N (2)
- D22: Buzzer
- D26: LED 1
- D27: LED 2
- D36: pin IN1 de pe driverul de motoare L298N (2)
- D37: pin IN2 de pe driverul de motoare L298N (2)
- D38: pin IN3 de pe driverul de motoare L298N (2)
- D39: pin IN4 de pe driverul de motoare L298N (2)
- D42: pin IN2 de pe driverul de motoare L298N (1)
- D43: pin IN1 de pe driverul de motoare L298N (1)
- D44: pin IN3 de pe driverul de motoare L298N (1)
- D45: pin IN4 de pe driverul de motoare L298N (1)
- D2: pin CE al modulului NRF24L01
- D53: pin CSN al modulului NRF24L01
- D51: pin MOSI al modulului NRF24L01
- D52: pin SCK al modulului NRF24L01
- D50: pin MISO al modulului NRF24L01

#### Driver motor L298N (1)

- OUT4: conectare motor 1
- OUT3: conectare motor 1
- OUT1: conectare motor 2
- OUT2: conectare motor 2
- +12V: conectare fir rosu carcasa baterii AA
- GND: conectare fir negru carcasa baterii AA, conectare GND-GND cu placuta Arduino MEGA

#### Driver motor L298N (2)

- OUT4: conectare motor 3
- OUT3: conectare motor 3
- OUT2: conectare motor 4
- OUT1: conectare motor 4
- +12V: conectare fir rosu carcasa baterii AA
- GND: conectare fir negru carcasa baterii AA, conectare GND-GND cu placuta Arduino MEGA

## Software Design

- Mediu de dezvoltare: Arduino IDE
- Librarii și surse 3rd-party: SPI.h, nRF24L01.h, RF24.h

LINK GITHUB: [https://github.com/Raluck20/Masina\\_cu\\_telecomanda](https://github.com/Raluck20/Masina_cu_telecomanda)

## **Telecomanda:**

Initializare:

- Biblioteca SPI este inclusa pentru a permite comunicarea prin protocolul SPI.
- Biblioteca nRF24L01 si RF24 sunt incluse pentru a gestiona modulul de comunicatie wireless RF24.
- Se defineste un obiect RF24 pentru modulul radio, specificand pinii 7 si 10 pentru conexiune.
- Se defineste adresa de scriere a datelor pentru modulul RF24.
- Se definesc pinii pentru citirea comenzilor (analogici si digitali) de la butoane.

Setup:

- Se initializeaza comunicatia seriala pentru debugging la o viteza de 9600 bps.
- Se initializeaza modulul radio si se configureaza pentru scrierea datelor la adresa specificata.
- Se seteaza nivelul de putere al modulului radio si se opreste ascultarea (receptia) pentru a permite transmiterea datelor.

Loop:

- In bucla principala, se citesc valorile analogice de la pinii definiti (A0-A7) care corespund directiilor de control: fata, spate, stanga, dreapta, fataStanga, fataDreapta, spateStanga, spateDreapta.
- Daca valorile citite depasesc un anumit prag (1015), se trimite o comanda specifica prin intermediul modulului radio. Comenzile posibile sunt: "fata", "spate", "stanga", "dreapta", "fata stanga", "fata dreapta", "spate stanga", "spate dreapta".
- In plus, se citesc valorile digitale de la pinii 2 si 3 pentru rotatia stanga si dreapta. Daca aceste butoane sunt apasate (valoare HIGH), se trimite comanda corespunzatoare.
- Dupa fiecare comanda trimisa, exista o intarziere de 500 ms pentru a preveni trimiterea continua de comenzi si pentru a oferi un timp de raspuns adecvat.

Functia 'comanda':

- Aceasta functie construiesc un mesaj de 32 de caractere, care contine comanda de transmisie. Mesajul este transmis folosind functia radio.write prin intermediul modulului RF24.

## **In codul pentru telecomanda am folosit urmatoarele concepte:**

- **SPI:** Utilizat pentru comunicarea cu modulul RF24 pentru transmisie wireless.
- **GPIO:** Utilizat pentru citirea starii pinilor digitali (rotireStanga si rotireDreapta).
- **ADC:** Utilizat pentru citirea valorilor analogice de la pinii A0-A7 (fata, spate, stanga, dreapta, etc.).
- **UART:** Utilizat pentru comunicarea seriala pentru debug (comunicare cu monitorul serial).

## **Masina:**

Initializare:

- SPI.h, nRF24L01.h, RF24.h sunt incluse pentru a permite comunicarea SPI si utilizarea modulului

#### RF24.

- Se definesc pinii de control pentru motoare (inainte si inapoi pentru fiecare din cele 4 motoare).
- Se definesc comenzile posibile (fata, spate, stanga, dreapta, etc.).
- Se definesc pinii pentru buzzer si LED-uri.
- Se initializeaza modulul RF24 pe pinii 2 si 53 si se seteaza adresa de citire.
- Pinii motoarelor, buzzer-ului si LED-urilor sunt setati ca iesiri.

#### Setup:

- Se initializeaza comunicatia seriala.
- Se initializeaza modulul RF24.
- Se configureaza pinii de control ai celor 4 motoare ca iesiri.
- Se configureaza pinii pentru controlul vitezei motoarelor si al buzzer-ului si LED-urilor ca iesiri.

#### Loop:

- Seteaza viteza motoarelor prin iesirile PWM la un nivel constant (150 din 255).
- Schimba starea buzzer-ului si a LED-urilor la fiecare interval de milisekunde (3000 ms), creând un efect de intermitenta (1.5 secunde pornite, 1.5 secunde oprite).
- Verifica daca exista date disponibile pe modulul radio.
- Citeste mesajul primit si il afiseaza pe monitorul serial.
- In functie de mesajul primit (text), codul controleaza directia de miscare a motoarelor. Pentru fiecare comanda, se schimba starea pinilor corespunzatori motoarelor pentru a executa miscarea dorita (fata, spate, stanga, dreapta, etc.).
- Daca nu exista comenzi disponibile, toate motoarele sunt oprite pentru a preveni miscarea accidentala.

#### **In codul pentru masina am folosit urmatoarele concepte:**

- **SPI** (Serial Peripheral Interface): Utilizat pentru comunicarea cu modulul RF24 pentru receptia comenzilor wireless.
- **GPIO** (General Purpose Input/Output): Utilizat pentru controlul motoarelor si a LED-urilor. Pinii motoarelor (ex. motor1Inainte, motor1Inapoi, etc.) si LED-urilor (led1, led2) sunt setati ca iesiri.
- **UART** (Universal Asynchronous Receiver-Transmitter): Utilizat pentru comunicarea seriala pentru debugging (comunicare cu monitorul serial).
- **Timere**: Utilizat pentru a alterna starea buzzer-ului si a LED-urilor la fiecare 3 secunde.
- **PWM** (Pulse Width Modulation): Utilizat pentru a seta o valoare PWM pentru anumiti pinii (pentru controlul vitezei).

## Rezultate Obținute



## Concluzii

Am reusit sa realizez o masina cu telecomanda conform planului propus initial.

### Indicatori de performanta:

- Timpul dintre trimiterea unei comenzi de la telecomanda si executia acesteia de catre masina < 0.5 secunde.
- Rata de succes a transmiterii si receptiei comenzilor prin modulul RF24 este de 100%.
- Motoarele functioneaza la viteza dorita si mentin o performanta consistenta in timp.
- Nu exista miscare accidentala sau nedorita a masinii cand aceasta nu primeste comenzi.
- Corectitudinea si consistenta functionarii buzzer-ului si LED-urilor.

### Optimizari pe care le-as putea adauga:

- Atasarea motoarelor in alt mod pe masina.
- Folosirea a mai putinor baterii (eu am ales sa folosesc mai multe si pentru a adauga greutate masinii pentru aderenta).

## Download

[pm\\_project.zip](#)

## Jurnal

- 30 aprilie - procurarea pieselor
- 3 mai - realizare descriere proiect
- 13-14 mai - realizarea telecomenzii fizice
- 15-16 mai - realizarea masinii fizice
- 16 mai - proiectare scheme electrice (telecomanda+masina)
- 13-16 mai - am implementat si modificat codul in paralel cu realizarea componentelor fizice
- 23 mai - realizare caroserie masina



ETAPA/SAPTAMANA	SAPTAMANA1	SAPTAMANA 2	SAPTAMANA 3
Comandare componente	2 zile		
Realizare descriere proiect	1 ora		
Realizarea telecomenzii fizice		2 zile	
Realizarea masinii fizice		2 zile	
Proiectare scheme electrice (telecomanda + masina)		5 ore	
Realizarea codului		4 zile	
Realizare caroserie masina			7 ore

## Bibliografie/Resurse

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/prj2024/ccontasel/raluca.movileanu>

Last update: **2024/05/26 21:22**

