

## Rc Car - Moldoveanu Paul-Adrian

### Introducere

Proiectul meu este o mașinuță RC controlată de pe telefon, dotată cu faruri, stopuri și senzor de parcare spate. Scopul este să simuleze funcțiile de bază ale unei mașini reale într-un model miniatural. Ideea a pornit din dorința de a combina controlul wireless cu senzori și iluminare, într-un proiect practic și distractiv. Este util atât pentru mine, ca mod de a învăța electronica aplicată, cât și pentru alții interesați de automatizări sau prototipuri auto.

Controlul se va face prin Bluetooth, iar senzorul de parcare va detecta obstacolele și va alerta utilizatorul. Farurile și stopurile vor funcționa în funcție de direcția de mers.

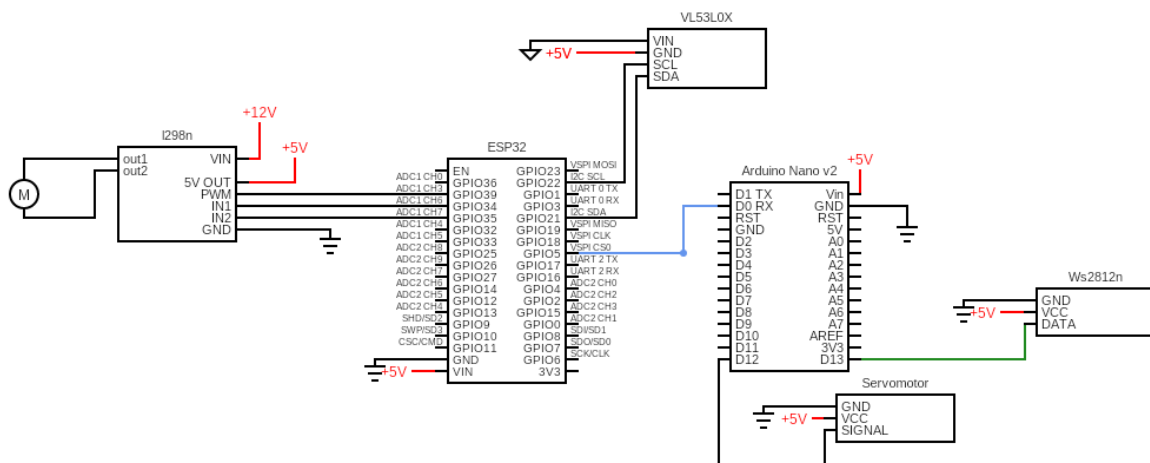
### Descriere generală

1. ESP32 Este creierul sistemului – primește comenzi de la aplicația de pe telefon prin Bluetooth sau WiFi și controlează restul componentelor (motoare, lumini, senzori).
2. Battery Asigură alimentarea întregului sistem. Este sursa principală de energie pentru toate modulele.
3. Arduino Nano – execută comenzile primite de la ESP32 pentru a controla servomotorul și banda LED.
4. DC Motor + DC Motor Control Unit Motoarele responsabile de deplasarea mașinuței. Unitatea de control (driver motor) primește semnale de la ESP32 pentru a controla viteza și direcția de mers.
5. Servo Motor Este folosit pentru direcție. ESP32 controlează unghiul său de rotație pentru a vira roțile mașinuței stânga/dreapta.
6. Led Strip Simulează farurile și stopurile mașinii. Controlate de ESP32 în funcție de direcția de mers (ex: aprindere la mers înainte sau frânare).
7. Senzor Distanță Este poziționat în spatele mașinuței și detectează obstacolele. Trimite date către ESP32, care le poate afișa în aplicație sau activa semnale vizuale/acustice.

Utilizatorul controlează mașinuța RC direct de pe telefon, prin intermediul unei aplicații mobile ce comunică cu ESP32 prin Bluetooth. Din aplicație, utilizatorul poate:

- Controla direcția de mers (înainte/înapoi/stânga/dreapta) prin butoane sau joystick virtual. Comenzile sunt trimise către ESP32, care ajustează motoarele în consecință.
- Vizualiza distanța față de obstacole atunci când mașinuța merge înapoi, datorită senzorului de parcare, astfel evitând coliziunile.
- Aprinde sau stinge farurile și stopurile – acestea sunt controlate de ESP32 și pot fi activate automat în funcție de direcția de mers sau manual din aplicație.
- Primi feedback vizual dacă un obstacol este detectat la o distanță mică.

### Hardware Design



Lista Piese :

1. Esp32
2. Arduino Nano
3. Servomotor SG90
4. WS2812B led
5. Senzor de Distanță VL53L0X
6. Modul driver motor L298N
7. Motor DC

Software Design

Stadiul actual al implementării software

Implementarea software a fost realizată cu succes. În prezent, sistemul este controlabil prin rețea Wi-Fi, expunând o interfață de tip API către aplicația de mobil. ESP32 configurează un punct de acces local (`Rc\_Car`) și rulează un server web care gestionează comenzi pentru:

- mișcare (viteză și direcție)
- controlul direcției prin UART
- iluminare RGB cu LED-uri WS2812B
- în curând: senzor de distanță laser pentru parcare

Funcționalități implementate:

- Control motor cu PWM (viteza)
- Direcție controlată prin comenzi UART
- LED-uri RGB controlate cu FastLED
- API-uri REST pentru control din aplicația de mobil
- Server Wi-Fi + DNS redirect la `rccar/`
- Comunicare serială între microcontrollere

Funcționalități în dezvoltare:

- Integrarea senzorului de distanță cu laser (pe SDA)
- Aplicația mobilă pentru control complet

Motivația alegerii bibliotecilor

- **WiFi/WebServer/DNSServer** - pentru conectivitate locală și expunerea API-urilor fără internet
- **FastLED** - pentru control eficient și flexibil al LED-urilor RGB
- **Arduino standard** - compatibilitate garantată cu ESP32 pentru PWM, UART etc.

☐ Element de noutate

- Controlul se face prin API REST - gata pentru aplicații mobile
- Server Wi-Fi cu DNS intern - acces ușor la interfață (`rccar/`)
- Separarea logicii de control și acționare prin **UART**
- Senzor de distanță cu laser adăugat pentru funcție de parcare inteligentă

☐ Justificarea utilizării funcționalităților din laborator

Concept din laborator	Utilizare în proiect
PWM	Controlul vitezei motorului prin `ledcWrite()`
UART	Comunicare între ESP32 și microcontroller secundar pentru direcție/lumini
I2C (SDA)	Utilizat pentru senzorul de distanță cu laser

☐ Structura proiectului și interacțiuni



- ESP32 gestionează toate comenzile API și controlează motorul
- Comenzile de direcție și lumină sunt trimise prin UART
- LED-urile RGB sunt gestionate prin FastLED
- Aplicația mobilă va trimite comenzi către serverul ESP32
- Senzorul de distanță (pe I2C) va trimite feedback pentru manevre de parcare

☐ Optimizări realizate

- Separare pe thread (core) diferit pentru server, evitând blocaje
- Comenzi UART trimise în mod eficient prin singleton (`SerialSender`)
- PWM configurat cu frecvență și rezoluție adaptate nevoilor motorului
- Mesajele API sunt compacte și rapide de procesat

Rezultate Obținute

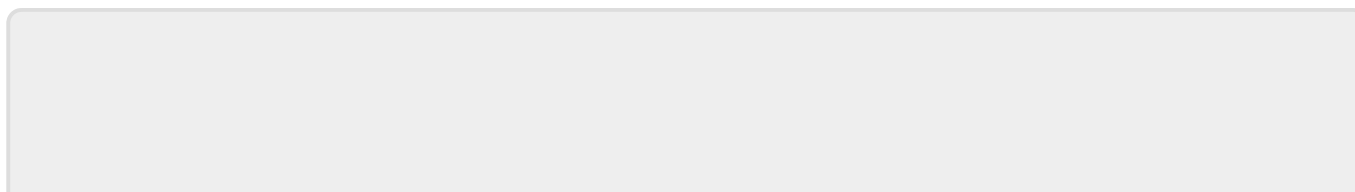
Demo video: [https://www.youtube.com/watch?v=RYxBSBJadco&ab\\_channel=Moldo](https://www.youtube.com/watch?v=RYxBSBJadco&ab_channel=Moldo)

Concluzii

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/prj2025/vstoica/paul.moldoveanu>



Last update: **2025/05/29 18:38**