

# Gyroscope controlled car

## Introducere

Proiectul „Gyroscope Controlled Car” urmărește dezvoltarea unui sistem de control wireless pentru un vehicul cu două roți, operat printr-o telecomandă care detectează mișcarea utilizatorului cu ajutorul unui giroscop. Acest sistem este conceput pentru a oferi un mod intuitiv și hands-free de a controla deplasarea unui robot mobil, doar prin înclinarea telecomenzii.

Proiectul îmbină mai multe concepte din domeniul sistemelor embedded, precum: comunicație wireless low-latency (ESP-NOW), controlul motoarelor prin PWM, interfațare I2C, UART pentru debug, GPIO, prelucrarea datelor senzorilor de mișcare, afișare în timp real și măsurare a tensiunii analogice(ADC). Soluția propusă este modulară și extensibilă, reprezentând o bază solidă pentru dezvoltarea de vehicule autonome sau controlate prin metode alternative.

## Descriere generală

Device-ul este compus din doua componente:

1. Masina
2. Telecomanda

Controlul vehiculului se bazează pe un senzor de mișcare MPU6050, montat pe o placă ESP32 aflată în telecomandă. Informațiile despre orientare (pitch, roll) sunt procesate și transmise în timp real către vehicul prin protocolul ESP-NOW, care permite o comunicare rapidă și directă între cele două microcontrolere ESP32, fără a necesita o rețea Wi-Fi.

Vehiculul, echipat cu un alt modul ESP32, primește comenzile, interpretează direcția dorită de deplasare și controlează două motoare prin intermediul unui driver L298N. Sistemul este completat de un ecran LCD care afișează în timp real starea bateriei, și de un senzor de distanță HC-SR04, ce permite detectarea obstacolelor și implementarea unei funcții de evitare a coliziunii..

## Hardware Design

### Lista de piese:

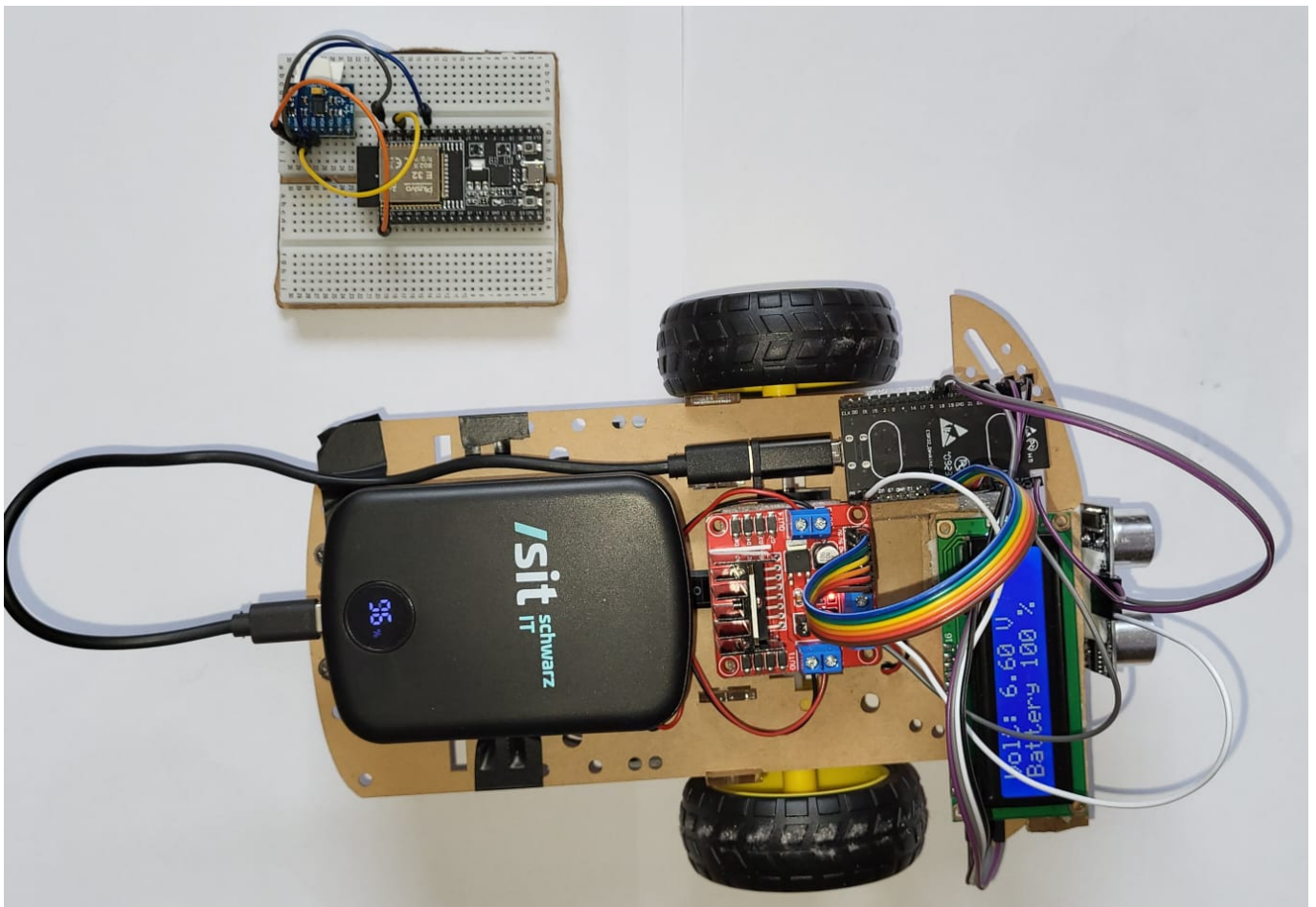
- [ESP32](#)
- [MPU6050](#)

- [L298N](#)
- [LCD](#)
- [Kit Masina](#)
- [HC-SR04](#)

## Diagrama bloc



## Montajul final



## Schema electrica



**Comunicare ESP32 - MPU6050**

MPU6050	ESP32	Funcție
VCC	3V3	Alimentare 3.3V
GND	GND	GND
SDA	GPIO 21	I2C - Date
SCL	GPIO 22	I2C - Ceas

**Masina****Comunicare ESP32 - L298N**

ESP32 Pin	L298N Pin	Funcție
GPIO 32	ENA	Viteză motor stânga (PWM)
GPIO 33	IN1	Direcție motor stânga
GPIO 25	IN2	Direcție motor stânga
GPIO 26	IN3	Direcție motor dreapta
GPIO 27	IN4	Direcție motor dreapta
GPIO 14	ENB	Viteză motor dreapta (PWM)
ADC 34	VCC	Măsurare tensiune baterie

**Comunicare ESP32 - LCD**

ESP32 Pin	LCD Pin (modul I2C)	Funcție
GPIO 21	SDA	Date I2C
GPIO 22	SCL	Ceas I2C
5V (Vin)	VCC	Alimentare
GND	GND	GND

**Comunicare ESP32 - HC-SR04**

ESP32 Pin	HC-SR04 Pin	Funcție
GPIO 12	Trig	Transmitere impuls ultrasonic
GPIO 14	Echo	Recepție ecou (măsurare timp întârziere)
5V	VCC	Alimentare senzor
GND	GND	GND

**Software Design**

Mediul de dezvoltare este ArduinoIDE.

## Biblioteci utilizate

- **esp\_now.h** - Această bibliotecă permite comunicarea wireless directă între dispozitive ESP32 folosind protocolul ESP-NOW (dezvoltat de Espressif). Este utilă pentru transmisii rapide și fără un router Wi-Fi.
- **WiFi.h** - Este necesară pentru a inițializa modulul Wi-Fi al ESP32. Deși ESP-NOW nu necesită o conexiune la rețea, modulul Wi-Fi trebuie pornit pentru a funcționa.
- **I2Cdev.h** și **MPU6050\_6Axis\_MotionApps20.h** - Biblioteci open-source pentru comunicarea cu senzorul MPU6050.
- **vector**
- **Wire.h** - Este biblioteca standard pentru comunicația I2C. Necesară pentru a comunica cu periferice I2C.
- **LiquidCrystal\_I2C.h** - Bibliotecă open-source care permite afișarea datelor pe LCD cu interfață I2C.

## Descriere software

### Funcționarea modului transmițător

Modulul transmițător are rolul de a colecta date de la senzorul de mișcare MPU6050 și de a le transmite, prin intermediul protocolului ESP-NOW, către un modul receptor.

#### Etapele de inițializare:

1. Inițializarea conexiunii ESP-NOW
  - Modulul activează interfața Wi-Fi și configurează conexiunea ESP-NOW.
  - Se adaugă un nou peer (receptor) utilizând adresa sa MAC, pentru a permite transmiterea datelor.
2. Configurarea senzorului MPU6050
  - Senzorul MPU6050 este inițializat și calibrat cu ajutorul funcției `setupMPU()`.
  - Această funcție setează senzorul în modul DMP (Digital Motion Processor), permițând obținerea de date prelucrate (pitch, roll, yaw).

#### Funcționarea `loop()`:

- Modulul citește în mod repetat datele din coada senzorului (FIFO).
- Valorile sunt prelucrate și mapate în intervalul 0-254, pentru a fi compatibile cu formatul transmisiei ESP-NOW.
- Datele sunt apoi încapsulate într-o structură și transmise către receptor prin ESP-NOW.

### Funcționarea modului receptor

Modulul receptor are rolul de a primi datele transmise prin ESP-NOW de la senzorul MPU6050 (de pe transmiiător) și de a executa acțiuni corespunzătoare asupra motoarelor, în funcție de mișcările detectate.

### Etapele de inițializare:

1. Inițializarea protocolului ESP-NOW
  - Se configurează modulul Wi-Fi al plăcii ESP32 în modul stație și se pornește protocolul ESP-NOW.
2. Configurarea pinilor de control
  - Se setează modurile de funcționare pentru pinii digitali implicați în controlul motoarelor, prin apelul funcției `setUpPinModes()`.
3. Inițializarea afișajului LCD
  - Se pornește comunicația I2C și se inițializează afișajul LCD, care va afișa ulterior nivelul bateriei.

### Recepția și procesarea datelor:

- La primirea unui pachet de date prin ESP-NOW, este apelată funcția `OnDataRecv()`.
- Aceasta citește structura de date recepționată și, în funcție de valorile primite (corespunzătoare orientării sau accelerației), apelează funcția `processCarMovement()` cu parametrul corespunzător.
- Pentru evitarea coliziunii, cu ajutorul senzorului de distanță, primul pas din funcția `OnDataRecv()` este determinarea distanței până la obstacol, iar dacă distanța este mai mică de 20 cm, se va introduce o întârziere de o secundă, timp în care funcția de primire a datelor nu va mai procesa informațiile primite.
- Funcția `processCarMovement()` interpretează comanda și controlează direcția de deplasare a vehiculului, trimițând comenzi către motoare prin intermediul funcției `rotateMotor()`.

### Monitorizarea nivelului bateriei:

- Pentru a monitoriza tensiunea bateriei care alimentează motoarele, pinul VCC al modului L298N este conectat la pinul 34 al plăcii ESP32, care permite citiri analogice.
- Se realizează o medie a mai multor citiri analogice pentru a obține o valoare mai stabilă a tensiunii.
- Nivelul bateriei este calculat în funcție de această medie și este afișat pe LCD.

## Rezultate Obținute

Proiectul demonstrează eficiența comunicării ESP-NOW pentru aplicații low-latency și fiabilitatea senzorilor utilizați în contextul controlului robotic. Prin combinarea mai multor tehnologii (MPU6050, HC-SR04, ESP32, LCD I2C), s-a obținut un sistem complet, capabil să răspundă în timp real la comenzile utilizatorului. Acest proiect oferă o bază solidă pentru extinderea ulterioară către vehicule complet autonome sau controlate prin alte metode, precum aplicații mobile sau comenzi vocale.

[Demo](#)

## Download

[ilie\\_nicolai\\_332cb.zip](#)

## Bibliografie/Resurse

### Resurse Hardware

- [Instalare suport ESP32](#)
- [ESP32-DevKitC Getting Started Guide](#)
- [L298N](#)
- [MPU6050](#)

### Resurse Software

- [I2Cdevlib](#)
- [Arduino](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/ilie.nicolai>



Last update: **2025/05/26 15:16**