

Robot pentru Cartografiere 2D cu LiDAR

Introducere

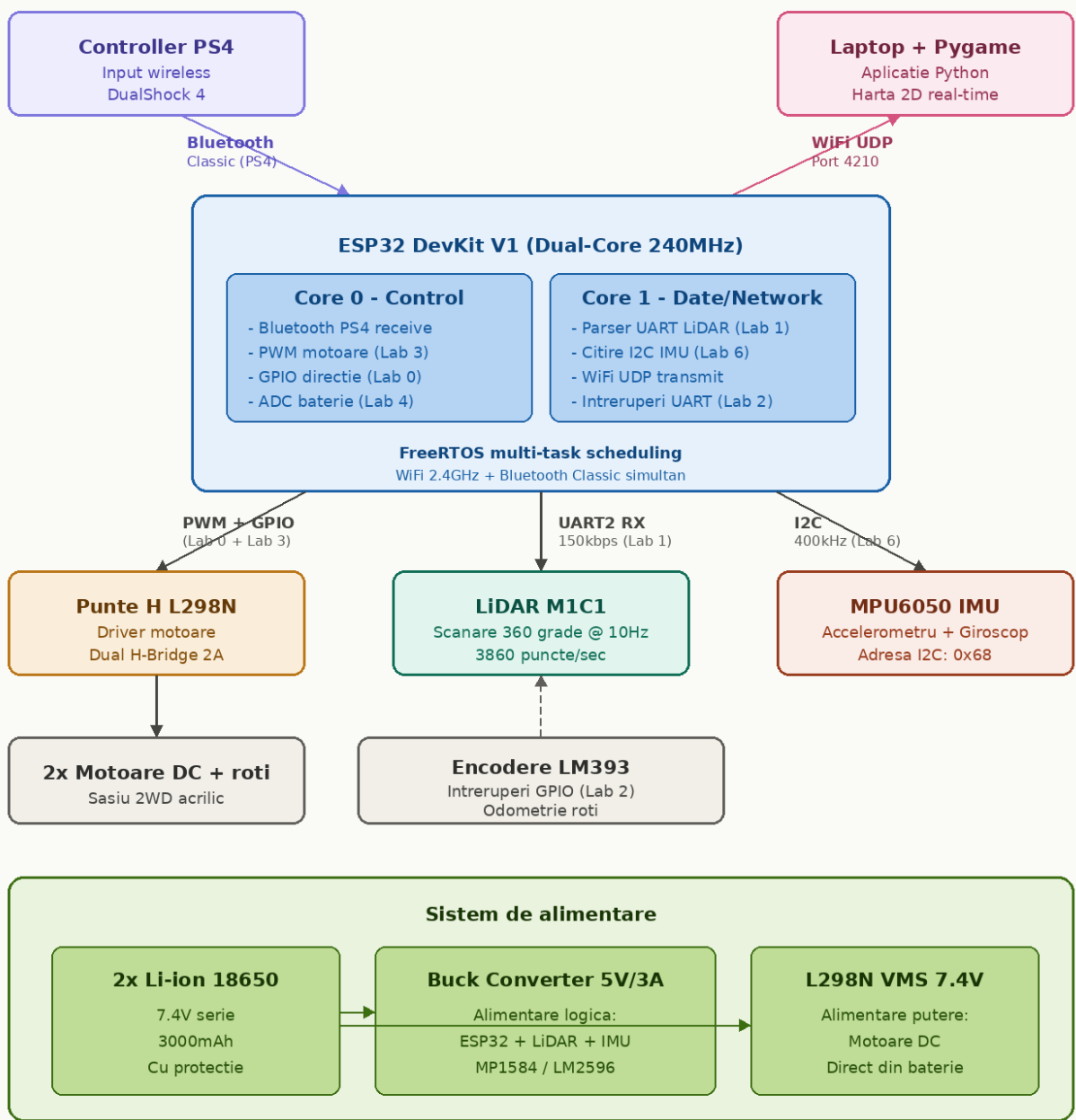
Proiectul constă în realizarea unui robot mobil pe roți, controlat wireless printr-un controller PS4 (Bluetooth), capabil să cartografieze în timp real spațiul unei camere folosind un senzor LiDAR cu scanare 360°. Robotul transmite datele de scanare prin **WiFi (UDP)** către o aplicație software rulată pe laptop, care construiește incremental o hartă 2D a mediului înconjurător.

Ce face proiectul:

- Se deplasează prin cameră sub control wireless (controller PS4 prin Bluetooth)
- Scanează continuu mediul cu LiDAR la 10Hz (~3860 puncte/secundă)
- Transmite datele LiDAR prin **WiFi UDP** către aplicația de pe laptop
- Aplicația Python pe laptop transformă datele polare în coordonate carteziene și desenează harta 2D a camerei în timp real

Descriere generală

Schema bloc - Robot autonom cu LiDAR si mapare 2D



Linii continue: comunicatie date | Linii intrerupte: feedback senzori | Verde: alimentare

Module hardware și interacțiunea lor:

- **ESP32 DevKit V1** - microcontroler dual-core care orchestrează tot sistemul. Folosește FreeRTOS pentru task-uri paralele pe cele două nuclee. Dispune de WiFi 2.4GHz și Bluetooth Classic integrate, esențiale pentru comunicarea dublă wireless.
- **LiDAR M1C1** - scanează 360° la 10Hz, trimite pachete binare prin UART la 150000 baud către ESP32.
- **MPU6050 (IMU)** - oferă date de orientare prin I2C, folosite pentru a corecta harta când robotul se rotește.
- **Driver L298N** - punte H care primește semnale PWM (3.3V) de la ESP32 și controlează motoarele DC cu putere mare din baterie.
- **Buck Converter 5V/3A** - regulator separat pentru a asigura alimentare stabilă pentru LiDAR și

ESP32, izolată de zgomotul motoarelor.

- **Controller PS4** - input wireless prin Bluetooth Classic.
- **Laptop (Python + Pygame)** - aplicație care primește pachete UDP prin WiFi de la ESP32 și desenează harta 2D în timp real.

Fluxul de date end-to-end:

1. **Input control:** Joystick PS4 → Bluetooth → ESP32 (Core 0)
2. **Procesare comenzi:** ESP32 generează semnale PWM → L298N → Motoare DC
3. **Achiziție senzori:** LiDAR (UART) + IMU (I2C) → ESP32 (Core 1)
4. **Buffer & parsare:** Core 1 procesează pachetele binare LiDAR
5. **Transmisie WiFi:** Core 1 împachetează datele în UDP datagrame → WiFi 2.4GHz → Laptop
6. **Vizualizare:** Aplicație Python primește UDP, transformă coordonate polare în carteziane, desenează cu Pygame

Hardware Design

Listă de piese

| Componentă | Model | Cantitate | Rol |
|-----------------|--------------------------------------|-----------|----------------------------------|
| Microcontroller | ESP32 DevKit V1 | 1 | Procesare centrală + WiFi/BT |
| Senzor LiDAR | M1C1 Mini 360° | 1 | Scanare mediu |
| Cablu adaptor | JST-PH 1.25mm 4-pin → Female Dupont | 1 | Conexiune LiDAR-ESP32 |
| Șasiu robot | Kit 2WD acril cu motoare și encodere | 1 | Mecanică |
| Driver motoare | L298N Dual H-Bridge | 1 | Control putere motoare |
| Convertor DC-DC | Buck MP1584 sau LM2596 (5V/3A) | 1 | Alimentare stabilă logică |
| IMU | MPU6050 (GY-521) | 1 | Orientare robot (Lab I2C) |
| Acumulatori | Li-ion 18650 3000mAh | 2 | Sursă energie |
| Suport baterii | 2x 18650 serial cu fire | 1 | Asamblare baterii |
| Încărcător | TP4056 sau extern dual | 1 | Reîncărcare |
| Senzori encoder | LM393 IR slot | 2 | Odometrie roți (Lab Întreruperi) |
| Switch | ON/OFF rocker | 1 | Pornire sistem |
| Fire Dupont | Set Male-Male, Male-Female | 1 set | Conectică |
| Breadboard | 400 puncte | 1 | Prototipare |
| Controller PS4 | Sony DualShock 4 (sau clonă) | 1 | Input wireless |

Pinout ESP32 DevKit V1

În lucru - Pinout-ul exact urmează să fie stabilit în timpul fazei de implementare hardware. Tabelul de mai jos este preliminar și poate suferi modificări în funcție de testarea practică.

| Pin ESP32 | Conexiune | Funcție | Lab |
|---------------|-----------------|---------------------|--------------|
| GPIO 16 (RX2) | LiDAR M1C1 - TX | Recepție date LiDAR | Lab 1 (UART) |

| | | | |
|---------------|--------------------------|----------------------------|--------------------|
| GPIO 17 (TX2) | LiDAR M1C1 - RX | Comandă LiDAR (opțional) | Lab 1 (UART) |
| GPIO 21 (SDA) | MPU6050 - SDA | Date I2C | Lab 6 (I2C) |
| GPIO 22 (SCL) | MPU6050 - SCL | Clock I2C | Lab 6 (I2C) |
| GPIO 18 | L298N - IN1 | Direcție motor stânga (1) | Lab 0 (GPIO) |
| GPIO 19 | L298N - IN2 | Direcție motor stânga (2) | Lab 0 (GPIO) |
| GPIO 23 | L298N - IN3 | Direcție motor dreapta (1) | Lab 0 (GPIO) |
| GPIO 5 | L298N - IN4 | Direcție motor dreapta (2) | Lab 0 (GPIO) |
| GPIO 25 (PWM) | L298N - ENA | Viteză motor stânga | Lab 3 (PWM) |
| GPIO 26 (PWM) | L298N - ENB | Viteză motor dreapta | Lab 3 (PWM) |
| GPIO 34 (ADC) | Divizor rezistiv baterie | Monitor tensiune baterie | Lab 4 (ADC) |
| GPIO 32 | Encoder roată stânga | Înterupere odometrie | Lab 2 (Înteruperi) |
| GPIO 33 | Encoder roată dreapta | Înterupere odometrie | Lab 2 (Înteruperi) |
| GPIO 2 | LED status WiFi | Indicator conexiune | Lab 0 (GPIO) |
| VIN (5V) | Buck converter OUT | Alimentare placă | - |
| GND | GND comun | Masa | - |

Software Design

Mediu de dezvoltare

- **Firmware ESP32:** Arduino IDE 2.x cu suport ESP32 (ESP32 Arduino Core), alternativ ESP-IDF
- **Aplicație PC:** Python 3.x cu Pygame pentru vizualizare, biblioteca standard socket pentru UDP

Librării și surse 3rd-party

- **WiFi.h** și **WiFiUdp.h** - librării standard ESP32 Arduino pentru rețea
- **Bluepad32** - librărie pentru gestionarea controllerului PS4 prin Bluetooth Classic pe ESP32 (<https://github.com/ricardoquesada/bluepad32>)
- **Wire.h** - librărie I2C standard Arduino pentru MPU6050
- **MPU6050 library** by Electronic Cats (sau driver custom)
- **socket** (Python stdlib) - comunicare UDP
- **Pygame** - vizualizare hartă 2D
- **NumPy** - calcule transformări coordonate

Notă: Driverul UART pentru LiDAR M1C1 va fi implementat de la zero, fără bibliotecă. Documentația protocolului va fi reconstruită din observarea fluxului de date și din resurse GitHub disponibile.

Rezultate Obținute

Această secțiune va fi completată progresiv pe măsură ce proiectul avansează. La momentul actual (Milestone Săpt 9) este în faza de proiectare și aprobare temă.

Etape planificate:

- **Săpt 10-11:** Asamblare hardware, testare individuală module (motoare, LiDAR, IMU, WiFi)
- **Săpt 12-13:** Integrare firmware, sincronizare task-uri, implementare protocol UDP
- **Săpt 14:** Aplicație Python receiver UDP + algoritm cartografiere
- **Săpt 15-16:** Testing end-to-end, optimizări, demonstrație finală

Concluzii

Va fi completat la finalizarea proiectului.

Download

Arhivele cu surse, scheme și documentație vor fi încărcate aici la finalizarea fiecărei etape.

Jurnal

- **Săpt 9 (curentă):** Alegerea temei, definitivare componente, schemă bloc, aprobare asistent.
- **Cercetare conector LiDAR:** Identificat că LiDAR M1C1 folosește JST-PH 1.25mm 4-pin (rar), găsit cablu adaptor pentru conexiune directă fără lipit.
- **Decizie alimentare:** Adăugat Buck Converter separat după ce am observat că regulatorul L298N are limită ~500mA, insuficient pentru ESP32 + LiDAR + IMU simultan.
- **Decizie microcontroler:** Ales ESP32 DevKit V1 standard pentru documentație extinsă, pini lipiți și WiFi/BT integrate.
- **Decizie protocol comunicare:** Ales UDP în locul TCP/USB Serial pentru a permite robot complet autonom (fără cablu) cu latență minimă pentru streaming LiDAR.

Bibliografie/Resurse

Resurse Hardware

- Datasheet ESP32: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- Datasheet L298N: <https://www.st.com/resource/en/datasheet/l298.pdf>
- Datasheet MPU6050: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- Pinout ESP32 DevKit V1: <https://lastminuteengineers.com/esp32-pinout-reference/>
- LiDAR M1C1 (community): căutare GitHub "M1C1 LIDAR protocol"

Resurse Software

- Bluepad32 (PS4 controller): <https://github.com/ricardoquesada/bluepad32>
- ESP32 Arduino Core: <https://github.com/espressif/arduino-esp32>
- FreeRTOS pe ESP32:
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>
- ESP32 WiFi UDP tutorial: <https://randomnerdtutorials.com/esp32-udp-client-arduino-ide/>
- Python socket module: <https://docs.python.org/3/library/socket.html>
- Pygame: <https://www.pygame.org/docs/>

Laboratoare PM utilizate

- Lab 0 - GPIO: <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2024>
- Lab 1 - UART: <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023>
- Lab 2 - Întreruperi: <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023>
- Lab 3 - Timere/PWM: <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>
- Lab 4 - ADC: <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024>
- Lab 6 - I2C: <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/cezar.zlatea/eduard.vasilescu>



Last update: **2026/05/09 20:26**