

<Sistem mobil de control bazat pe microcontroler pentru controlul stingerea incendiilor>

Documentul de proiectare

Cuprins

1. Introducere	3
1.1 Scopul documentului	3
2. Prezentare generală și abordări de proiectare	4
2.1 Prezentare generală	4
2.2 Presupuneri/ Constrângeri/ Riscuri	4
2.2.1 Presupuneri	4
2.2.2 Constrângeri	4
2.2.3 Riscuri	5
3. Considerații de proiectare.....	6
3.1 Obiective și linii directoare (ghiduri)	6
3.2 Metode de dezvoltare	6
3.3 Strategii de arhitectură.....	7
4. Arhitectura Sistemului și Proiectarea Arhitecturii.....	8
4.1 Vedere logică	8
4.2 Arhitectură hardware	8
4.3 Arhitectură software.....	9
4.4 Arhitectura informațiilor	9
4.5 Arhitectura de comunicații interne.....	9
5. Scenarii de utilizare	10
6. Proiectare de detaliu.....	11
6.1 Proiectare hardware de detaliu	11
6.2 Proiectare software de deatliu	11
6.3 Proiectare detaliată de securitate.....	12
6.4 Proiectare de detaliu pentru performanța sistemului.....	12
6.5 Proiectare detaliată a comunicațiilor interne (între componente)	12
7. Controale pentru verificarea integrității sistemului.....	13

1. Introducere

Acest document are rolul de a descrie în detaliu arhitectura și proiectarea unui sistem integrat bazat pe microcontroler, dezvoltat pentru detecția și stingerea incendiilor în medii închise, de dimensiuni reduse. Sistemul este construit sub forma unui robot mobil autonom, capabil să detecteze o sursă de foc folosind senzori specializați, să navigheze spre aceasta evitând obstacole și să activeze o pompă de apă pentru stingere.

Proiectul integrează componente hardware (senzori, servomotoare, pompă, șasiu 4WD) și software (cod Arduino, control ESP32), coordonate printr-o arhitectură modulară. Microcontrolerul central este Arduino Mega 2560, iar comunicația wireless se realizează prin modulul ESP32. Sistemul este controlabil în două moduri: autonom (decizii luate de robot) și manual, unde utilizatorul poate interveni direct prin comenzi transmise prin Wi-Fi.

Acest document este parte din documentația de licență și vine în completarea Documentului de Cerințe (SRS), concentrându-se pe arhitectura sistemului, strategiile de proiectare, interfețele și scenariile tehnice de funcționare. Se adresează atât cadrelor didactice, cât și celor interesați de replicarea sau îmbunătățirea sistemului.

1.1 Scopul documentului

Scopul acestui document este de a detalia modul în care cerințele definite în etapa de analiză au fost transformate în soluții concrete de proiectare.

Documentul explică:

- cum este organizat sistemul hardware (componente, conexiuni, alimentare),
- cum este structurat software-ul (modular, bazat pe funcții și stări),
- ce interacțiuni există între părți (comunicare senzor–Arduino–ESP32),
- și cum se integrează toate acestea într-un produs funcțional și testabil.

În plus, documentul poate servi drept ghid tehnic pentru viitoare adaptări sau extinderi ale sistemului, oferind o bază solidă pentru înțelegerea completă a arhitecturii implementate.

2. Prezentare generală și abordări de proiectare

2.1 Prezentare generală

Proiectul constă în realizarea unui robot mobil autonom, capabil să detecteze flăcări, să se deplaseze autonom către sursa de foc și să acționeze o pompă de apă pentru stingere. Robotul funcționează pe baza unor componente hardware conectate și coordonate printr-un program scris în limbaj C/C++ pentru Arduino. Sistemul este gândit modular, astfel încât fiecare parte (detectie, navigare, stingere, comunicare) să poată fi testată și extinsă separat.

Funcționalitatea generală este împărțită în:

- Modul autonom, în care robotul ia decizii fără intervenție umană
- Modul manual, în care operatorul controlează acțiunile robotului în timp real prin ESP32 și interfață de monitorizare

2.2 Presupuneri/ Constrângeri/ Riscuri

2.2.1 Presupuneri

- Sistemul va fi utilizat în spații interioare, fără curenți de aer sau lumină solară directă.
- Operatorul va avea acces la o rețea Wi-Fi locală pentru a trimite comenzi și a primi feedback.
- Toate componentele hardware (senzori, module) vor fi alimentate corect cu 5V sau 3.3V prin convertorul LM2596.
- Codul va fi rulat pe un singur microcontroller (Arduino Mega 2560), fără sisteme distribuite.
- Flacăra detectată va fi vizibilă direct pentru senzori, fără obstacole sau fum între robot și sursă.

2.2.2 Constrângeri

- Tensiunea de alimentare este limitată la 7.4V (2x Li-ion), ceea ce impune utilizarea unor convertoare de tensiune.
- ESP32 este folosit exclusiv pentru comunicație, nu rulează logica de decizie.
- Sistemul funcționează pe o platformă mobilă simplă (4WD), fără suspensii, deci este optim doar pentru suprafețe plane.
- Nu este folosită nici o bază de date sau stocare persistentă – toate deciziile se iau în timp real.
- Sistemul este testat și calibrat manual, fără senzori de feedback avansați (ex: giroscop, IMU).

2.2.3 Riscuri

- ESP32 pierde conexiunea Mediu Reset hardware/manual; fallback pe mod autonom.
- Flacăra nu este detectată corect Ridicat Testare și poziționare corectă a senzorilor.
- Pompa consumă prea mult curent Ridicat Adăugare tranzistor de control și protecție.
- Obstacle detectat prea târziu Mediu Măsurare frecvență + servo cu scanare continuă.
- Comenzi eronate de la utilizator Scăzut Validare mesaje și limitarea comenzilor permise.

3. Considerații de proiectare

3.1 Obiective și linii directoare (ghiduri)

Proiectarea sistemului a avut în vedere următoarele obiective principale:

- Fiabilitate – robotul trebuie să funcționeze fără erori în timpul unei sesiuni complete de test (minim 20 min)
- Modularitate – fiecare componentă (detecție, navigare, stingere, comunicație) trebuie să poată fi testată și modificată individual
- Simplitate – sistemul trebuie să fie ușor de înțeles, întreținut și reprodus cu resurse minime
- Control dual – trebuie să existe atât mod autonom, cât și mod manual de intervenție
- Timp real – reacțiile robotului la stimuli trebuie să fie rapide (sub 500 ms pentru detecție + acțiune)

Liniile directoare au inclus folosirea unor convenții clare în cod (nomenclatură, comentarii), testare incrementală a modulelor hardware, și prioritizarea siguranței în interacțiunea cu focul real (nivel redus, controlat).

3.2 Metode de dezvoltare

Sistemul a fost dezvoltat incremental folosind o abordare modulară și orientată pe componente hardware. Dezvoltarea a urmat o logică de tip testare pe piese → integrare → testare totală.

Metodologia Scrum a fost aplicată pentru organizarea muncii în 4 sprinturi:

- Sprint 1: detecție
- Sprint 2: navigare
- Sprint 3: stingere
- Sprint 4: interfață și testare

Diagrama de secvență, cazurile de utilizare și arhitectura sistemului au fost construite folosind notatii UML. Limbajul de programare folosit este C++ (Arduino), în mediul Arduino IDE.

3.3 Strategii de arhitectură

Arhitectura sistemului este una centralizată:

- Arduino Mega 2560 este unitatea de control care gestionează toate deciziile și comenziile hardware
- ESP32 este conectat prin UART și se ocupă doar de transmiterea și recepționarea comenzilor
- Componentele hardware (senzori, pompe, motoare) sunt tratate ca modul hardware dedicat, fiecare controlat prin pin digital sau PWM
- Motivele acestei alegeri:
- Arduino Mega are suficiente pini digitali pentru a controla toate modulele fără expansiuni
- Separarea logicii de comunicație de logica de decizie crește fiabilitatea
- Sistemul poate funcționa și fără ESP32, în caz de eșec Wi-Fi → fallback pe mod autonom
- Alte alternative luate în calcul (ex: Raspberry Pi ca master) au fost respinse din cauza consumului mai mare, complexității și timpului de dezvoltare extins.

4. Arhitectura Sistemului și Proiectarea Arhitecturii

4.1 Vedere logică

Sistemul este împărțit în patru module logice:

1. Modul de detecție
 - 3 senzori de flacără (analog + digital)
 - 1 senzor ultrasonic HC-SR04 montat pe servo pentru scanare
2. Modul de decizie și control
 - Arduino Mega 2560
 - Cod ce analizează datele senzorilor și comandă componentele active
3. Modul de stingere
 - Servo motor pentru orientarea duzei
 - Pompă controlată prin tranzistor + driver logic
4. Modul de comunicare
 - ESP32 conectat prin UART
 - Primește comenzi (manual), trimite feedback (autonom)

Aceste module colaborează în fluxul: Detecție → Decizie → Acțiune → Feedback.

4.2 Arhitectură hardware

Componente principale:

- Arduino Mega 2560 – controler central
- ESP32 (ESP-WROOM-32) – comunicare wireless
- 3x Senzori flacără – detectare flacără și direcție
- 1x HC-SR04 + Servo SG90 – scanare obstacole
- 4x Motoare DC – deplasare 4WD
- L298N Driver – control motoare
- Servo SG90 pentru Duza + Pompă apă 3–5V – acțiune stingere
- 2x Baterii Li-ion 3.7V + LM2596 – alimentare

Este un sistem centralizat, cu toate semnalele și comenzile gestionate de Arduino.

4.3 Arhitectură software

Codul este structurat în următoarele componente logice:

- setup() – inițializare senzori, pini, UART
- loop() – buclă principală cu funcții:
 - citireSenzoriFlacara()
 - detectareObstacol()
 - controlMotoare()
 - stingereFoc()
 - transmitereStatusESP()

Modulul ESP32 primește comenzi seriale simple (mode>manual, pump=on) și răspunde cu mesaje (status: foc detectat).

4.4 Arhitectura informațiilor

Sistemul nu stochează date persistent. Informația circulă astfel:

- Senzorii trimit semnale (analog/digital) → Arduino procesează → decide acțiunea
- ESP32 primește doar mesaje în timp real (nu salvează nimic)

Toate datele sunt volatile, iar deciziile sunt reacții directe la stimuli de moment.

4.5 Arhitectura de comunicații interne

- ESP32 ↔ Arduino – UART (Serial)
- Senzori ↔ Arduino – pini digitali/analogici
- Motoare, pompă, servo – pini digitali/PWM

Comunicarea este una direcțională, pe fire fizice directe, fără protocoale complexe. ESP32 transmite și primește text simplu. Nu se utilizează protocoale IoT avansate (ex: MQTT).

5. Scenarii de utilizare

Scenariul 1: Funcționare în modul autonom

Operatorul pornește robotul, iar acesta funcționează fără intervenție umană. Scopul este detectarea și stingerea focului într-un spațiu controlat.

Flux de execuție:

1. Operatorul alimentează robotul și îl pornește fizic (switch ON).
2. ESP32 pornește în rețea, dar nu primește comenzi – modul implicit este autonom.
3. Arduino începe scanarea cu senzorii de flacără și ultrasonic.
4. Dacă este detectată o flacără:
 - se stabilește direcția cu ajutorul valorilor analogice;
 - se verifică distanțele (evitare obstacole);
 - robotul se deplasează spre foc.
5. La apropierea de foc:
 - duza este rotită automat;
 - pompa este activată pentru câteva secunde.
6. După stingere, Arduino trimite mesaj de feedback către ESP32 (ex: status: foc stins).
7. Robotul oprește pompa și revine în poziție de standby.

Scenariul 2: Funcționare în modul manual

Operatorul preia controlul robotului prin comenzi transmise de pe un dispozitiv conectat la ESP32.

Flux de execuție:

1. Operatorul pornește robotul și se conectează la ESP32 prin Wi-Fi.
2. Trimite comanda: mode=manual
3. Robotul trece în modul manual – senzorii sunt dezactivați.
4. Operatorul trimite comenzi text:
 - forward, left, right – pentru controlul motoarelor;
 - pump=on / off – pentru activarea pompei;
 - servo=rotate – pentru direcționarea duzei.
5. La final, operatorul poate trimite stop sau mode=autonom pentru a reveni la comportamentul automat.

6. Proiectare de detaliu

6.1. Proiectare hardware de detaliu

Sistemul este compus din următoarele componente hardware:

- Arduino Mega 2560 – microcontrollerul central care controlează toate celelalte componente.
- ESP32 (ESP-WROOM-32) – modul Wi-Fi pentru comunicarea dintre robot și operator.
- 3 senzori de flacără (AO + DO) – detectează prezența și direcția sursei de foc.
- Senzor ultrasonic HC-SR04 montat pe servo SG90 – scanează mediul și detectează obstacolele.
- 4 motoare DC + șasiu 4WD – permit deplasarea robotului în toate direcțiile.
- Driver motoare L298N – controlează alimentarea și sensul de rotație al motoarelor.
- Servomotor SG90 (duză) – rotește mecanic duza pentru a direcționa jetul de apă.
- Pompa de apă (3–5V) – activează jetul de apă pentru stingerea focului.
- LM2596 – convertor de tensiune care scade 7.4V la 5V pentru alimentarea componentelor sensibile.
- 2 baterii Li-ion 3.7V conectate în serie (7.4V total) – sursa de alimentare principală a sistemului.
- Conexiuni hardware:
- Senzorii sunt conectați pe pinii analogici și digitali ai Arduino.
- Servomotoarele sunt controlate prin semnal PWM.
- Pompa este comandată printr-un pin digital cu tranzistor de comutare.
- ESP32 comunică cu Arduino prin conexiune UART (Serial1).
- Driverul L298N este conectat între Arduino și motoare.
- Sistemul este centralizat, toate deciziile și comenzile fiind emise de Arduino.

6.2. Proiectare software de detaliu

Software-ul este scris în C/C++ folosind mediul Arduino IDE. Structura codului este modulară, bazată pe funcții care se ocupă individual de detecție, navigare, stingere și comunicare.

Funcțiile principale sunt:

- `detectare_foc()` – analizează valorile de la senzorii de flacără
- `evitare_obstacol()` – citește distanța și rotește senzorul ultrasonic
- `deplasare()` – comandă motoarele pentru a merge spre sursă
- `stingere_foc()` – controlează servo și activează pompa
- `ascultare_comenzi_manual()` – interpretează comenzile venite prin ESP32

Funcția `loop()` este folosită pentru rulare continuă, iar `setup()` pentru inițializări. Comenzile ESP32 sunt simple, de tipul `mode=manual`, `pump=on`, `status?`.

6.3. Proiectare detaliată de securitate

Sistemul nu lucrează cu date sensibile, dar s-au luat măsuri minime de siguranță:

- ESP32 se conectează doar la rețele Wi-Fi cunoscute
- Nu există baze de date sau stocări externe
- Comenzile primite sunt filtrate și validate
- Se poate adăuga un buton fizic pentru oprire de urgență

6.4. Proiectare de detaliu pentru performanța sistemului

Sistemul este optimizat pentru răspuns rapid:

- Reacția la detectarea focului este sub 0.5 secunde
- Viteza de deplasare este de aproximativ 20–30 cm/s
- Pompa este activată pentru 5 secunde în scenariul standard
- Autonomia totală este de aproximativ 15–20 minute

Nu există întârzieri vizibile în execuția comenzilor, iar sistemul procesează totul în timp real.

6.5. Proiectare detaliată a comunicațiilor interne (între componente)

Comunicarea se face local, între componente hardware:

- Arduino comunică cu ESP32 prin UART Serial
- Senzorii trimit date direct către pinii analogici/digitali ai Arduino
- Motoarele, servomotoarele și pompa sunt controlate de Arduino prin PWM sau semnale logice
- Nu se folosesc protocoale de rețea complexe, ci doar comunicație directă prin fire. ESP32 trimite și primește comenzi simple în format text.

7. Controale pentru verificarea integrității sistemului

Controalele aplicate în cadrul sistemului au fost gândite pentru a asigura funcționarea corectă, sigură și stabilă a robotului. Deoarece este un sistem hardware-software integrat, măsurile luate vizează atât protecția fizică a componentelor, cât și validarea fluxului de comenzi și date.

Controale implementate:

- Verificare senzor flacăra – dacă valorile analogice sunt nule constant, sistemul intră în modul standby.
- Validarea comenzilor ESP32 – se acceptă doar comenzi predefinite; cele necunoscute sunt ignorate automat.
- Feedback vizual și serial – fiecare acțiune (activare pompă, pornire motor, rotație servo) este confirmată în consolă și/sau prin LED-uri.
- Protecție alimentare – LM2596 reglează tensiunea pentru componentele de 5V; dacă tensiunea scade sub 6V, pompa și motoarele se dezactivează automat.
- Testare individuală a componentelor – fiecare modul hardware (senzori, pompă, servo, ESP32) a fost testat separat înainte de integrare finală.
- Reset manual – sistemul poate fi repornit în orice moment prin întreruperea alimentării (fizic sau cu buton de reset).

Sugestii suplimentare de integritate (posibil de implementat):

- Adăugarea unui buzzer de eroare pentru semnalizarea unor probleme de funcționare.
- Implementarea unui watchdog timer software pentru a reseta Arduino în caz de blocare a codului.
- Integrarea unui buton fizic de oprire de urgență în cazul unei funcționări necontrolate.