

M.U.S.C.A.

Introducere

Ce face proiectul: M.U.S.C.A. (Modular Uncrewed Self-Controlled Aircraft) este o drona cu 4 motoare controlata autonom prin intermediul unui ESP-32. In etapa curenta de dezvoltare, sistemul va fi capabil sa decoleze si sa efectueze un zbor la punct fix pentru un interval fixat de timp.

Scopul proiectului: Scopul este implementarea si validarea unui sistem de control pentru drone pe un microcontroller, pentru a fi folosit in urmatoarele etape de dezvoltare a proiectului.

Care a fost ideea: Ideea de la care am pornit a fost ca dronele existente de obicei au o arhitectura optimizata pentru un scop anume (cinema, militar, etc). Implementarea curenta este concentrata pe sistemul de control al dronei, scopul fiind obtinerea unui zbor stabil.

De ce este util: Proiectul propune o arhitectura de tip modular, ce le permite utilizatorilor sa schimbe payload-ul transportat de drona, folosind astfel o singura drona pentru multiple activitati, reducand astfel costurile necesare testarii unei varietati de modele diferite. In implementarea curenta, proiectul demonstreaza capacitatea de zbor stabil.

Descriere generală

Proiectul este construit pe baza unui microcontroller ESP-32, responsabil de a citi datele de la senzori, a calcula parametrii de zbor pentru drona si de a actiona motoarele.

Module: - o baterie LiPo 3s1p 2.1Ah (12V, 30C), din care sunt alimentate motoarele - 4 Motoare BLDC A2212, fiecare putand genera ~700g de thrust la capacitate maxima - 4 ESC-uri de 30A, fiecare primind semnal de la microcontroller prin GPIO si controland individual cate un motor - ICM-20602 este accelerometrul folosit pentru a determina inclinarea si acceleratia dronei, comunicand cu microcontroller-ul prin SPI - VL53L0X este un senzor de ToF, folosit pentru a estima distanta dronei pana la sol, care comunica cu microcontroller-ul prin IIC.



Hardware Design



Software Design

Mediu de dezvoltare

Firmware-ul a fost dezvoltat folosind **PlatformIO** (extensie VS Code), cu framework-ul **Arduino** pentru ESP-32. Configuratia proiectului se gaseste in `platformio.ini`, unde sunt specificate viteza de clock a CPU-ului (80 MHz), viteza flash-ului (40 MHz) si dependintele de biblioteci.

Biblioteci si surse 3rd-party

- **Arduino SPI** (built-in) — comunicatie cu accelerometrul ICM-20602; configurat la 1 MHz, MSB first, SPI Mode 3
- **Arduino Wire** (built-in) — linie IIC disponibila pentru senzorul ToF VL53L0X (neutilizata in implementarea curenta)
- **ESP32Servo** (madhephaestus/ESP32Servo@^3.0.5) — generare semnale PWM pentru controlul ESC-urilor; foloseste timer hardware dedicat si permite setarea perioadei la 50 Hz si a pulsului in microsecunde
- **INA226** (robtillaart/INA226@^0.5.2) — biblioteca pentru senzorul de monitorizare a tensiunii/curentului de pe magistrala de alimentare; senzorul fizic nu mai este functional, codul aferent este comentat

Algoritmi implementati

1. Citirea accelerometrului (SPI burst read)

La fiecare iteratie a buclei principale, microcontroller-ul initiaza o tranzactie SPI si citeste in mod secvential 6 octeti incepand de la registrul `ACCEL_XOUT_H (0x3B)`, cu bitul de citire setat (`OR 0x80`). Cei 6 octeti reprezinta valorile brute pe 16 biti ale acceleratiei pe axele X, Y si Z.

2. Calculul unghiurilor de inclinare (pitch si roll)

Din valorile brute ale accelerometrului se calculeaza unghiurile de inclinare ale dronei folosind functia `atan2`:

```
roll = atan2(ay, az) * 180 / PI
pitch = atan2(-ax, sqrt(ay*ay + az*az)) * 180 / PI
```

Aceste formule ofera o estimare stabila a inclinarii in conditii de acceleratie dominata de gravitatie.

3. Algoritm de control proportional (P-controller)

Semnalul PWM de baza pentru toate motoarele este de 1300 μ s. Fiecare motor primeste o corectie proportionala cu unghiurile de pitch si roll, cu gainul $k_p = 4.0$ pe ambele axe. Semnalele sunt

calculate astfel incat fiecare motor sa compenseze inclinarea in directia sa:

Motor	Formula semnal PWM
Fata-Stanga	$\text{base} + k_p \cdot (-\text{pitch}) + k_r \cdot \text{roll}$
Fata-Dreapta	$\text{base} + k_p \cdot (-\text{pitch}) + k_r \cdot (-\text{roll})$
Spate-Stanga	$\text{base} + k_p \cdot \text{pitch} + k_r \cdot \text{roll}$
Spate-Dreapta	$\text{base} + k_p \cdot \text{pitch} + k_r \cdot (-\text{roll})$

Semnalele sunt limitate (clamp) la intervalul [1000, 1500] μs pentru a nu depasi 50% din turatia maxima in aceasta faza de testare.

4. Failsafe

Daca unghiul de pitch sau roll depaseste 60° , sistemul intra in modul failsafe si trimite semnal de 1000 μs (oprire) tuturor motoarelor, prevenind un accident in caz de inclinare extrema.

5. Sincronizare non-blocking

Bucula principala ruleaza fara delay-uri blocante, folosind timestamp-uri (`millis()`):

- semnalele PWM catre ESC-uri sunt actualizate la **50 Hz** (la fiecare 20 ms)
- telemetria seriala (roll, pitch, semnale motoare) este afisata la **5 Hz** (la fiecare 200 ms)

Surse si functii implementate

- `spi_write(byte reg, byte data)` — scrie un octet intr-un registru al IMU-ului prin SPI
- `clip(float raw, float min, float max)` — limiteaza o valoare in intervalul [min, max]
- `setup()` — initializeaza comunicatia SPI, scoate IMU-ul din sleep (`PWR_MGMT_1 = 0x00`), configureaza si armeaza cele 4 ESC-uri (semnal 1000 μs timp de 10 secunde pentru procedura de armare)
- `loop()` — bucla principala: citeste accelerometrul, calculeaza pitch/roll, calculeaza semnalele PWM pentru fiecare motor, scrie semnalele catre ESC-uri si afiseaza telemetria pe UART

Rezultate Obținute

In urma dezvoltarii proiectului, zborul propriu-zis nu a putut fi testat din cauza defectarii a doi senzori cheie pe parcursul lucrarilor: senzorul de distanta VL53L0X (ToF) si senzorul de monitorizare a curentului INA226. Fara senzorul ToF, sistemul nu are informatie despre altitudine, deci nu poate stabili drona pe verticala. Fara INA226, nu exista protectie la supracurent in software.

In schimb, a fost realizata si validata o **simulare de zbor pe banc**, in care drona este tinuta fixata si se testeaza raspunsul sistemului de control:

- Accelerometrul ICM-20602 citeste corect datele prin SPI si calculeaza unghiurile de pitch si roll in timp real
- La inclinarea manuala a placii (tilt), motoarele de pe partea ridicata cresc turatia, iar cele de pe partea coborata o reduc, ceea ce reprezinta comportamentul corect al unui controller proportional

pentru stabilizare

- Mecanismul de failsafe functioneaza: la inclinari mai mari de 60° pe orice axa, toate motoarele sunt oprite imediat
- Semnalele PWM sunt generate stabil la 50 Hz catre toate cele 4 ESC-uri
- Telemetria seriala afiseaza in timp real valorile de roll, pitch si semnalele trimise fiecarui motor

Concluzii

Proiectul a demonstrat ca un controller proportional de baza pentru stabilizarea unei drone poate fi implementat si validat pe un microcontroller ESP-32, chiar si in absenta unui zbor real. Logica de control raspunde corect la variatiile de inclinare, ceea ce confirma ca arhitectura software este functionala.

Principala lectie invatata este legata de robustetea hardware: senzorii periferici sunt vulnerabili in contextul unui sistem alimentat la 12V cu motoare BLDC, iar izolarea galvanica sau protectia la supratensiune pe liniile de semnal sunt necesare intr-o iteratie viitoare.

Ca directii de continuare:

- Inlocuirea senzorilor defecti si validarea zborului real
- Adaugarea datelor de la giroscop (disponibil tot pe ICM-20602, neutilizat momentan) pentru un controller PID complet, care sa elimine deriva acumulata in timp ce apare la controlul bazat exclusiv pe accelerometru
- Implementarea controlului de altitudine pe baza senzorului ToF

Download

[pm-musca.zip](#) — arhiva cu sursele firmware (src/main.cpp, platformio.ini) si schemele hardware ale proiectului.

Jurnal

- **Saptamana 1 (28 apr - 4 mai):** Sosirea componentelor, documentare datasheet-uri, definirea arhitecturii generale a sistemului.
- **Saptamana 2 (5 - 11 mai):** Primul test hardware: un motor conectat la un ESC, motorul a pornit si a raspuns corect la semnal PWM. Conectarea IMU-ului (ICM-20602) prin SPI, verificarea datelor brute prin monitor serial. Tentativa de conectare a senzorului ToF (VL53L0X) — defectat in urma unei conexiuni gresite de alimentare.
- **Saptamana 3 (12 - 18 mai):** Montarea tuturor celor 4 motoare pe cadru. Conectarea ESC-urilor si a senzorului de curent INA226; sistemul functional, senzorul raporta date corecte. Defectarea

INA226 prin descarcare electrostatica (ESD) la atingerea cu mana neizolata — codul aferent comentat. Dezvoltarea algoritmului de control proportional.

- **Saptamana 4 (19 - 25 mai):** Implementarea failsafe-ului si a sincronizarii non-blocking. Teste de simulare pe banc: tilt al IMU-ului si observarea raspunsului diferentiat al motoarelor. Ajustarea gainurilor. Finalizare firmware si documentatie.

Bibliografie/Resurse

- datasheet motor A2212:

https://cdn.robotshop.com/rbm/a00a7635-653b-4220-aac9-b0c23c5c5e2c/5/520795f9-301f-4e66-a0a3-019811d1f78b/222c434c_a2212-brushless-motor.pdf

- datasheet accelerometru:

<https://bluerobotics.com/wp-content/uploads/2022/05/ICM20602-DATASHEET.pdf>

- datasheet senzor ToF:

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout.pdf>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/nicolae.macovei1412>



Last update: **2026/05/25 04:20**