

# Drona ESP32

Nume: Gheorghiu Adrian  
Grupa: 332CC

## Introducere

### Ce face

Proiectul este o drona de tip quadcopter, bazata pe un modul ESP32-WROOM-32. Drona poate fi controlata cu ajutorul telefonului mobil prin WiFi, folosind o aplicatie pe Android. Astfel, utilizatorul se va conecta la reseaua WiFi a modulului ESP, iar apoi va avea posibilitatea de a seta din aplicatie IP-ul si port-ul dronei (sau se pot utiliza setarile default). Dupa aceea, utilizatorul va avea posibilitatea de a controla drona folosind doua joystick-uri cu ajutorul carora se pot modifica distanta fata de sol, rotatia fata de axa verticala si inclinatia in orice directie fata de planul orizontal. De asemenea, utilizatorul va avea optiunea de a controla inclinatia dronei prin inclinarea telefonului.

### Scopul proiectului

Scopul proiectului este de a oferi o drona care poate fi controlata cu usurinta de orice persoana care are un telefon mobil cu Android, fara a fi nevoie de telecomenzi radio scumpe.

### Ideea de la care am pornit

M-a atras de mult ideea de a imi construi propria drona si de a imi scrie propriul firmware care sa o echilibreze si de asemenea o aplicatie de mobil cu care sa o pot controla cu usurinta. Am considerat proiectul la PM ca fiind o oportunitate buna de realiza acest proiect.

### De ce este util?

Proiectul este util atat pentru mine cat si pentru altii deoarece drona se poate controla prin WiFi printr-un protocol simplu de tip text. Astfel, utilizatorul poate sa schimbe modul in care controleaza drona, prin implementarea protocolului folosit de drona. De asemenea, aplicatia cu care vine drona poate fi si ea folosita pentru a controla alte drone, daca este posibila modificarea firmware-ului acestora pentru a raspunde la comenzile prin WiFi ale aplicatiei. In acelasi timp, pe mine proiectul ma

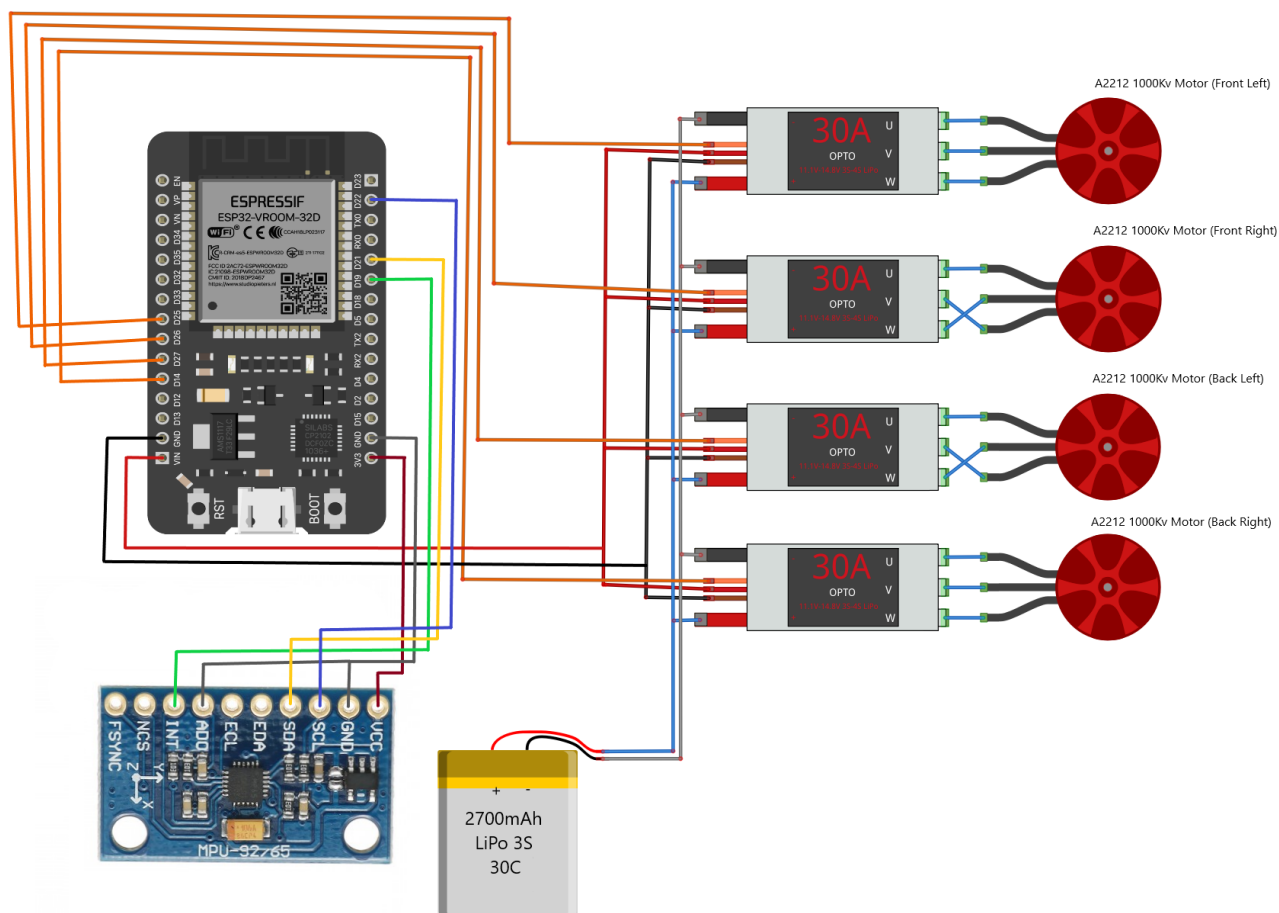
ajuta sa inteleg mai bine domeniile (Sisteme de control, Networking, Programare paralela) si tehnologiile (Arduino, Android) care trebuie combinate pentru a rezulta ceva functional.

## Descriere generală



Aplicatia de pe android va comunica prin WiFi cu microcontrollerul ESP32, folosind un protocol tip text simplu pentru a transmite informatii precum puterea de referinta a motoarelor (throttle), rotatia fata de axa verticala si inclinatia fata de planul orizontal. Microcontrollerul va interpreta aceste comenzi si va seta corespunzator unghiurile dorite fata de cele 3 axe (Yaw, Pitch, Roll). Pentru aflarea unghiurilor curente ale dronei, se va folosi modulul DMP(Digital Motion Processor) din cadrul acestuia, comunicarea realizandu-se cu ajutorul I<sup>2</sup>C. In final, microcontrollerul comunica prin PWM cu cele 4 controllere care vor controla viteza de rotatie a motoarelor.

## Hardware Design



## Lista de piese

- 4 x Motor Brushless A2212 1000Kv 13T
- 4 x Controller Motor ESC30A
- 1 x Microcontroller ESP32-WROOM-32
- 1 x Senzor MPU6500
- 1 x Acumulator LiPo 3S 2700mAh 30C
- 1 x Cadru Drona F450
- 2 x Breadboard

## Software Design

### Schema logica



### Medii de dezvoltare si biblioteci

- Arduino IDE (firmware ESP32)
- Android Studio (aplicatie Android)
- Biblioteca MPU6050 by Electronic Cats (pentru citire date de pe senzorul MPU6500)
- Biblioteca FreeRTOS ESP32 (pentru creare si controlare a taskurilor)

### Descrierea firmwareului, algoritmi si structuri

Firmwareul de pe arduino va rula un algoritm de tip PID pentru a echilibra drona. Acest algoritm ruleaza ca un task separat FreeRTOS ce se executa pe intrerupere de timer la un timp predefinit. Astfel, la fiecare intrerupere se va semnala taskului sa continue prin intermediul functiilor de sincronizare FreeRTOS. Timpul la care ruleaza codul de echilibrare, numit si looptime, este ales ca fiind acelasi ca frecventa la care procesorul DMP din cadrul senzorului MPU6050 furnizeaza datele, respectiv 200Hz. In paralel cu codul de echilibrare, ruleaza bucla principala a programului care verifica daca sunt date de citit de pe retea si le citeste, dupa care verifica daca sunt date de citit de pe senzor. Datele de pe senzor se citesc la intreruperea externa data de senzorul MPU6050, microcontrollerul citind prin I<sup>2</sup>C datele in urma intreruperii.

Firmwareul se poate gasi la urmatorul link de Github, impreuna cu un alt firmware folosit pentru calibrarea controllerelor ESC:

<https://github.com/RaySteak/ESP32-Drone>

De asemenea, codul aplicatiei Android folosita pentru a controla drona se gaseste la urmatorul link:

<https://github.com/RaySteak/DroneControlApp>

## Rezultate Obținute

Video type not supported by browser

Daca videoclipul de mai sus nu se incarca sau are probleme, las de asemenea un link catre YouTube (care se vede mai putin bine din cauza compresiei):

<https://www.youtube.com/watch?v=LCQZVNR1ETE>

## Concluzii

Desi am testat o multitudine de parametri PID si am verificat codul de echilibrare sa fie corect, drona este destul de instabila. Atunci cand tin drona in mana, echilibrarea pare sa mearga, orice miscare intr-o anumita directie activand corespunzator motoarele pentru a aduce inapoi drona in pozitia plana. Din ce m-am documentat, instabilitatea pe care o prezinta drona este data de controllerele de motoare, care nu sunt de cea mai buna calitate. Astfel, raspunsul intarziat al controllerelor combinata cu faptul ca acestea isi pierd configurarea rezulta in supra-reglaj si instabilitate. Bineinteles, sunt si alti factori care pot face drona mai stabila: montarea senzorului giroscopic astfel incat sa fie cat mai putin afectat de vibratiile motoarelor si cat mai fix, dar si faptul ca elicele pot fi si ele echilibrate prin smirgheluire. Acesti factori au insa un impact mai minor.

## Download

[adrian\\_gheorghiu\\_332cc\\_drona\\_esp32.zip](#)

Arhiva contine: codul firmware principal pentru drona, codul firmware pentru calibrarea controllerelor ESC, apk-ul pentru aplicatia de Android si schemele.

## Bibliografie/Resurse

- Documentatia FreeRTOS ESP32:  
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>
- Exemple folosire biblioteca MPU6050:  
<https://github.com/ElectronicCats/mpu6050/tree/master/examples>
- Datasheet ESC30A:  
[https://www.optimusdigital.ro/index.php?controller=attachment&id\\_attachment=451](https://www.optimusdigital.ro/index.php?controller=attachment&id_attachment=451)
- Explicatii algoritm PID (dar si multe alte informatii utile despre drone DIY, prezente in toate videoclipurile de pe acest canal): <https://www.youtube.com/watch?v=JBvnB0279-Q>

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2022/ndrogeanu/drona\\_esp32](http://ocw.cs.pub.ro/courses/pm/prj2022/ndrogeanu/drona_esp32)



Last update: **2022/06/01 19:06**