

# Drona cu sistem de echilibrare automat

## Introducere

Proiectul **Drone Stabilizer** reprezintă construcția unei drone (quadcopter), bazată pe motoare cu perii (brushed), capabilă să își mențină echilibrul în aer în mod autonom. Nucleul proiectului din punct de vedere software este implementarea unui algoritm de control de tip **PID (Proportional-Integral-Derivative)** care procesează datele de la un senzor inerțial pentru a ajusta viteza motoarelor în timp real.

Funcționalități principale:

- stabilizarea automată pe axele Pitch, Roll și Yaw folosind giroscopul și accelerometrul
- recepția comenzilor de zbor (throttle, direcție) prin protocol wireless de 2.4GHz
- controlul turației celor 4 motoare prin semnale PWM transmise către punțile H
- monitorizarea orientării dronei în timp real
- gestionarea energiei pentru a asigura stabilitatea electronicii în timpul căderilor de tensiune ale motoarelor

**Laboratoare folosite:** UART (Lab 1), Interruperi (Lab 2), Timere/PWM (Lab 3), SPI (Lab 5 - NRF24), I2C (Lab 6 - MPU).

## Descriere generală

## Schema Bloc



### Module si interactiuni:

#### Drona:

- **ATmega328PB:** unitatea centrală de procesare. Citește datele de orientare de la MPU6050, calculează unghiurile de înclinare, rulează algoritmi PID de stabilizare și ajustează viteza motoarelor în timp real prin semnale PWM. Primește comenzi de la distanță prin modulul NRF24 și gestionează comenzile primite de la remote controller.
- **MPU6050:** senzorul IMU (Inertial Measurement Unit) care combină un giroscop pe 3 axe cu un accelerometru pe 3 axe. Giroscopul măsoară viteza unghiulară (grade/secundă), iar accelerometrul măsoară forțele liniare, inclusiv gravitația. La pornire, MCU-ul efectuează o calibrare pe 1000 de eșantioane pentru a elimina offset-urile statice. În zbor, datele brute sunt combinate printr-un filtru

complementar: 96% giroscop (precizie pe termen scurt) și 4% accelerometru (corecție drift pe termen lung), rezultând unghiurile de roll, pitch și yaw. Comunicare prin I2C. Timer2 generează intreruperi o data la o milisecunda, cand se ruleaza PID-ul pentru a se corecta directia dronei.

- **NRF24:** modulul de comunicație wireless care operează pe banda de 2.4GHz. Drona funcționează în modul receptor (listening), așteptând comenzi de la un remote controller. Prima comandă obligatorie este m — aceasta declanșează calibrarea MPU și permite pornirea motoarelor. Ulterior, avem comenzi de urcare, coborare, stanga, dreapta, fata si spate, dar si o comanda pentru oprire instanta a elicelor. Comunicare cu MCU-ul prin SPI.
- **Drivere MX1616H (x2):** drivere de motor cu punte H dublă, fiecare controlând câte două motoare brushed. Primesc semnalul PWM de la MCU prin registrele OCR (Output Compare Register) ale timerelor hardware (Timer0 pentru motoarele stânga, Timer1 pentru motoarele dreapta) și translatează comenzile digitale în tensiune aplicată direct pe bobinele motoarelor. Direcția de rotație (CW/CCW) este setată la inițializare prin pinii de direcție ai porturilor C și D.
- **4 motoare brushed:** dispuse în configurație quad — față-stânga (CCW), față-dreapta (CW), spate-stânga (CW), spate-dreapta (CCW). Sensurile alternante de rotație sunt esențiale pentru anularea momentului giroscopic și controlul yaw-ului. Viteza fiecărui motor este ajustată independent de algoritmul de stabilizare, câte 15 unități PWM per pas, clamped între 0 și 255.
- **LiPo 1S 3.7V:** alimentează direct driverele MX1616H și motoarele. Tensiunea de 3.7V este ridicată la 5V prin buck converter pentru alimentarea MCU-ului, MPU6050 și NRF24.

#### Telecomanda:

- **Arduino Nano V3:** joaca rolul de **Remote controller**, care este conectat la laptop si primeste comenzile din laptop prin protocolul **UART**, care de asemenea este conectata la un modul NRF24.

## Hardware Design

### Lista de piese

Nr.	Componentă	Cantitate	Rol în proiect
1	<b>ATmega328PB</b>	x1	Flight Controller — unitatea centrală de procesare
2	<b>MPU6050 — GY-521</b>	x1	IMU cu giroscop și accelerometru pe 3 axe
3	<b>MX1616H</b>	x2	Driver punte H — controlul motoarelor brushed
4	<b>Motor brushed 820</b>	x4	Propulsie — configurație quad
5	<b>NRF24</b>	x2	Transreceptor wireless 2.4GHz — primire, transmitere comenzi
6	<b>Buck converter</b>	x1	Ridicător tensiune 3.7V → 5V pentru partea logică
7	<b>LiPo 1S 3.7V 500mAh 95C</b>	x1	Sursă principală de energie
8	<b>Arduino Uno NANO</b>	x1	Remote controller

### Schema electrica



## Pini folositi

Componentă	Pin componentă	Pin ATmega328PB	Explicație
MPU6050	VCC	5V	Modulul este alimentat la 5V
MPU6050	GND	GND	Masă comună
MPU6050	SDA	PC4 / SDA	Pin standard pentru date I2C
MPU6050	SCL	PC5 / SCL	Pin standard pentru clock I2C
NRF24	VCC	3.3V	Modulul necesită alimentare la 3.3V
NRF24	GND	GND	Masă comună
NRF24	CE	PD3	Chip Enable — activare modul RX/TX
NRF24	CSN	PD7	Chip Select — selecție SPI
NRF24	SCK	PB5 / SCK	Clock SPI
NRF24	MOSI	PB3 / MOSI	Date SPI master → slave
NRF24	MISO	PB4 / MISO	Date SPI slave → master
Motor față-stânga (CCW)	IN1	PD2	Control direcție — Driver MX1616H
Motor față-stânga (CCW)	IN2 / PWM	PD5	Semnal PWM — viteză motor
Motor față-dreapta (CW)	IN1	PC0	Control direcție — Driver MX1616H
Motor față-dreapta (CW)	IN2 / PWM	PB1	Semnal PWM — viteză motor
Motor spate-stânga (CW)	IN1	PD4	Control direcție — Driver MX1616H
Motor spate-stânga (CW)	IN2 / PWM	PD6	Semnal PWM — viteză motor
Motor spate-dreapta (CCW)	IN1	PC1	Control direcție — Driver MX1616H
Motor spate-dreapta (CCW)	IN2 / PWM	PB2	Semnal PWM — viteză motor
Buck converter	VIN	VBAT (LiPo)	Intrare 3.7V de la baterie
Buck converter	VOOUT	VCC placă	Ieșire 5V pentru partea logică

Pinii PC4 și PC5 au fost aleși pentru MPU6050 deoarece sunt pinii dedicați pentru I2C pe ATmega328PB. Comunicarea se face la 400kHz (fast mode) pentru a minimiza latența citirii datelor IMU.

Pentru NRF24L01+ am folosit pinii hardware SPI (PB3, PB4, PB5) deoarece SPI hardware este semnificativ mai rapid decât bit-banging. CE pe PD3 și CSN pe PD7 sunt pini digitali obișnuiți, aleși să nu intre în conflict cu timerele folosite pentru motoare.

Timer2 este configurat în mod CTC cu un prescaler de 64 și OCR2A = 249, ceea ce generează o întrerupere la fiecare 1ms. În ISR-ul acestui timer se incrementează contorul checkMpu. În bucla principală, la fiecare 50 de incrementări (adică la fiecare 50ms) se citesc datele de la MPU6050, se calculează unghiurile prin filtrul complementar și se rulează cei 3 controlleri PID. Timer2 a fost ales pentru această sarcină tocmai pentru că Timer0 și Timer1 sunt ocupați cu PWM-ul motoarelor.

Timer0 este configurat în mod Fast PWM cu prescaler 8, generând semnal PWM pe OC0A (PD6) și OC0B (PD5) — folosiți pentru motoarele din stânga. Registrele OCR0A și OCR0B controlează direct ciclul de lucru, adică viteza fiecărui motor.

Timer1 este configurat similar în mod Fast PWM pe 8 biți cu prescaler 64, generând semnal PWM pe OC1A (PB1) și OC1B (PB2) — folosiți pentru motoarele din dreapta. Registrele OCR1AL și OCR1BL sunt cele scrise de algoritmul de stabilizare la fiecare iterație PID.

Direcția de rotație a fiecărui motor este setată o singură dată la inițializare prin pinii de direcție ai driverelor MX1616H, nu prin PWM. Algoritmul PID modifică doar viteza (registrele OCR), nu și direcția — motoarele brushed pe o dronă nu își inversează sensul în zbor normal.

## Software Design

Descrierea codului aplicației (firmware):


- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).

**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

# Jurnal

## Etapa 1 de Hardware

- Am lipit pe o placa de prototipare buck converterul si driverele ca sa aibe gnd si vcc comun de la baterie.
- Am lipit pini pe o placa de prototipare pe care am bagat-o in placuta pentru a putea lipi in continuare fire.
- Am terminat de lipit tot ce era acolo dupa aproximativ 30 ore oneste de munca.
- Am reusit sa ansamblez drona si sa cuprind tot inaustrul ei.
- Momentan am reusit sa setez elicele in sensurile bune, deci se ridica de la sol, dar nu pentru mult timp, trebuie reglata software mult.





## Bibliografie/Resurse

## Resurse hardware

- **ATmega328PB Xplained Mini** documentatie Microchip:  
<https://www.microchip.com/en-us/development-tool/ATMEGA328PB-XMINI>
- **Modul GY-521 (MPU6050):**  
<https://sigmanortec.ro/Modul-girosopic-si-accelerometru-3-axe-GY-521-p126016326>
- **Driver MX1616H:** De la Victor din laborator
- **Motor coreless 8x20mm 3.7V:**  
<https://sigmanortec.ro/en/mini-coreless-motor-20x85-1mm-shaft-50000rpm-37v>
- **Modul NRF24:** <https://www.sigmanortec.ro/en/nrf24l01-24ghz-wireless-transceiver-module>
- **Modul ridicător tensiune 3.7V → 5V:** <https://sigmanortec.ro/en/step-up-modules>
- **Baterie LiPo 3.7V 95C 500mah:**  
<https://www.emag.ro/baterie-r-line-gens-ace-tattu-500-mah-3-7-v-95-c-multicolor-taa5001s95jsl/pd/DP84FRMBM/>

## Resurse software / scheme

- EasyEDA - realizare schema electrica: <https://easyeda.com/>
- Arduino - citire accurate a sezorilor:  
<https://forum.arduino.cc/t/mpu6050-gyroscope-readings-are-drifting/1113217/10>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/stefan.oprea1711>



Last update: **2026/05/16 23:04**