Quadcopter

O drona cu 4 elice controlata prin WiFi

Proiectul reprezinta o drona de mici dimensiuni, cu 4 elice, controlata prin WiFI. Aceasta serveste drept platforma de baza pentru posibile imbunatatiri, precum adaugarea unei camere video, efectuarea de miscari acrobatice sau transportul unor obiecte.

Ideea de la care am pornit este aceea de a crea o versiune simpla a unei drone (unica limitare fiind puterea motoarelor, ca rezultat al constrangerilor de buget) pentru a explora nivelul de dificultate a unui astfel de proiect. Astfel, proiectul acesta constituie un “canvas gol” pentru potentiale upgrade-uri care ii pot extinde drastic functionalitatile.

Descriere generală

Microcontrollerul folosit este un ESP8266, care va fi alimentat de o baterie a carui curent va fi reglat de un converter DC-DC. Datele culese de senzori (modul de accelerometru + giroscop, respectiv senzor cu ultrasunete) vor fi transmise prin I2C, iar miscarile vor fi controlate de pe telefon prin WiFi. Comunicarea dintre microcontroller cu motoarele se realizeaza prin mosfeturi, scopul lor fiind de a regla curentrul transmis la motoarele coreless, deci, implicit, turatia lor.

Schema bloc a proiectului:

Hardware Design

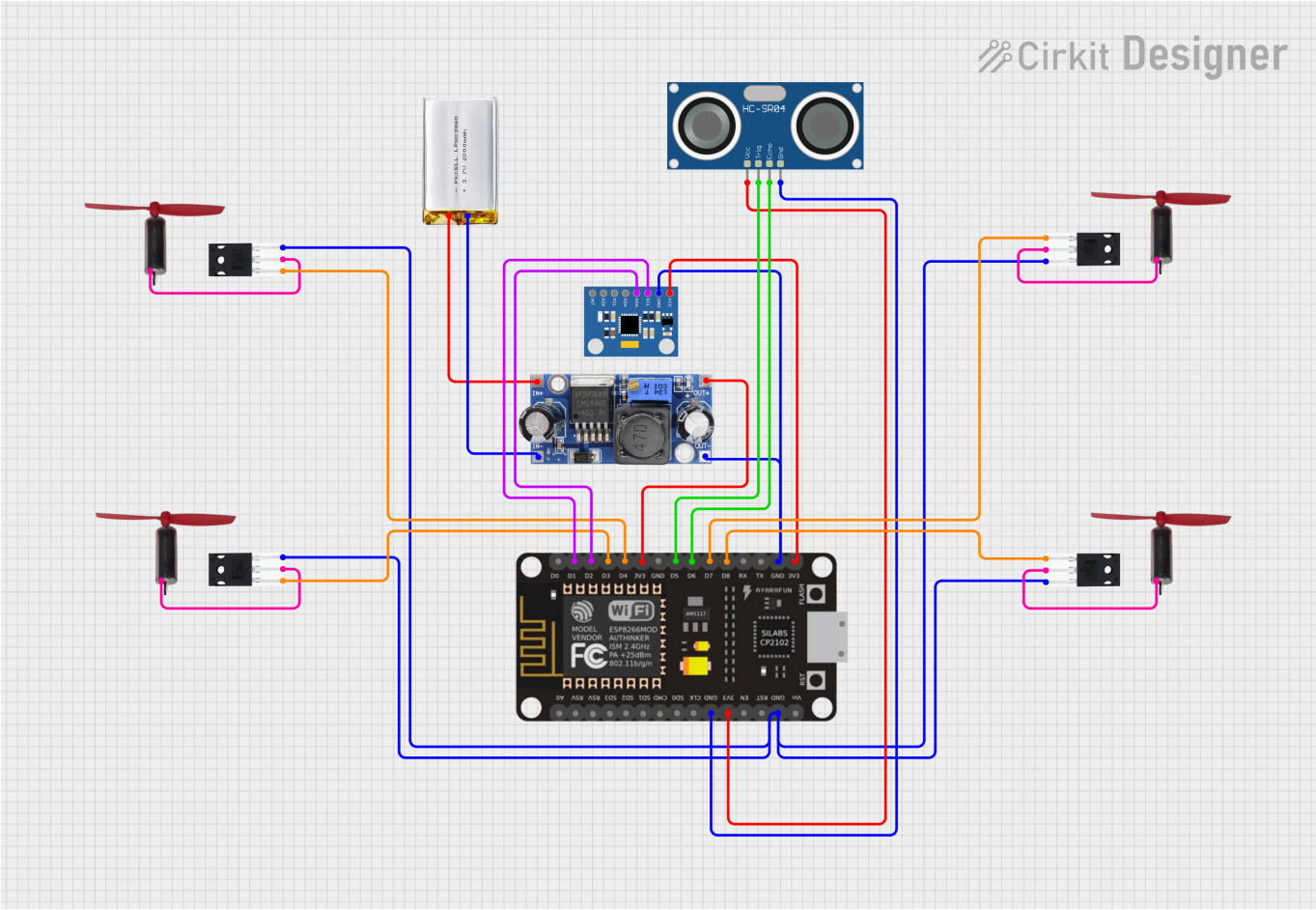
Elemente de hardware design:

- Microcontroller: **ESP8266 NodeMCU**
- Accelerometru + giroscop: **MPU6050**
- **DC-DC Buck Converter**
- 4 x **Motor Coreless 8520**
- 4 x **Elice 65mm cu 2 pale**
- 4 x **AO3400 MOSFET**

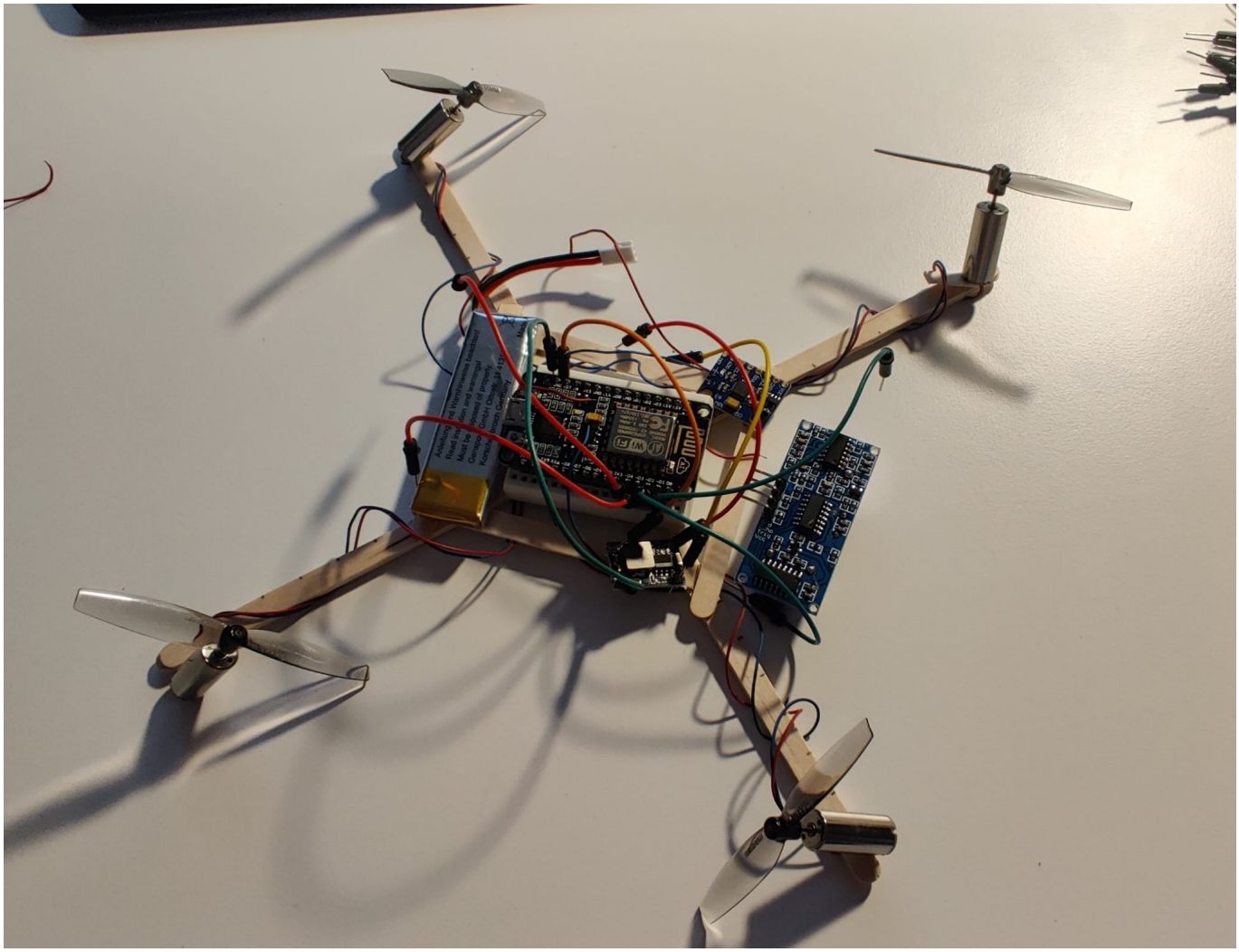
- **Fire conectoare**
- Baterie: **Tattu 600mAh 3.7V 30C**
- Senzor ultrasonic: **HC-SR04**

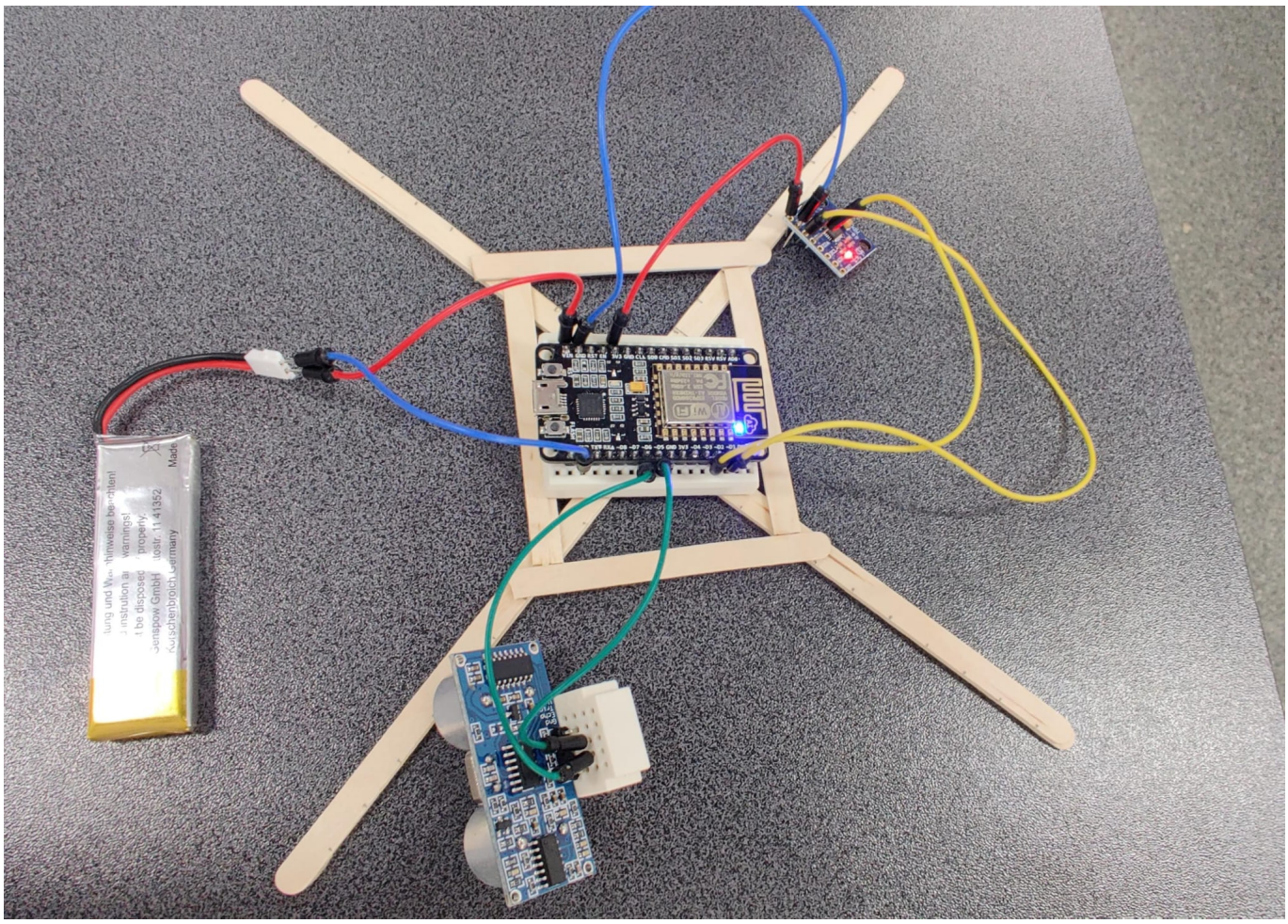
Bateria are un curent de 3.7V insa regulatorul de tensiune al microcontroller-ului ESP8266 il coboara la 3.3V. Astfel, polul pozitiv al bateriei va fi conectat la pinul VCC al microcontrollerului, iar cel negativ la GND. In ceea ce priveste senzorii, atat MPU-ul cat si HC-SR04 au elementele de alimentare: VCC respectiv GND conectate la ESP prin 3V3 respectiv GND. Senzorul ultrasonic are nevoie de tensiune de 5V, motiv pentru care se foloseste amplificatorul de tensiune XL6009E1. Comunicarea I2C a acestora cu ESP-ul se realizeaza astfel: MPU-ul are pinul SCL conectat la D1 si SDA la D2 pe microcontroller, in timp ce senzorul ultrasonic are pinul Trig conectat la D5 si Echo la D6. Mai raman motoarele coreless, conectate pe diagonala secundara cu polaritate normala la polul pozitiv al bateriei, iar pe diagonala secundara cu polaritate opusa, pentru a obtine o rotatie in sensul acelor de ceasornic pe diagonala secundara si trigonometric pe cea principala. Celelalte fire se conecteaza la pinul Drain al MOSFET-urilor, pinii GND la GND-ul de pe ESP, iar pinii Source sunt legati de pinii de pe D3 (dreapta fata), D4 (stanga fata), D5 (dreapta spate), D6 (stanga spate) de pe ESP.

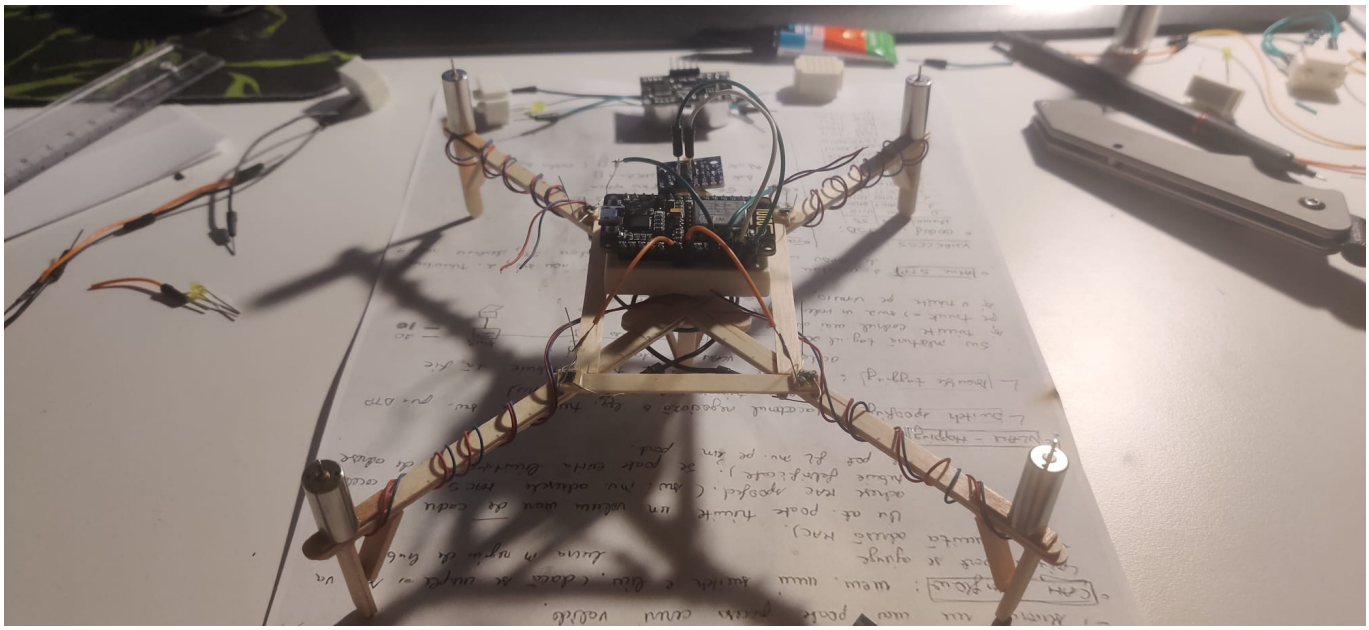
Schema electrica a proiectului:



Implementare hardware a proiectului:







Software Design

Programul ruleaza pe un ESP8266 si ofera atat stabilizare automata, cat si pilotaj manual printr-o

pagina web accesibila de pe telefon. La pornire, microcontrolerul se conecteaza la primul SSID definit; daca nu reuseste in intervalul de siguranta trece automat pe o a doua retea. Serverul HTTP integrat, disponibil pe portul 80, serveste o singura pagina HTML care contine patru slidere: Throttle, Pitch, Roll si Yaw. Sliderul de putere ramane pe valoarea setata, in timp ce celelalte trei revin la zero imediat ce utilizatorul nu le mai atinge, imitand revenirea unui stick cu arc.

Pentru comanda motoarelor sunt folosite patru MOSFET-uri low-side legate la pinii D3-D6. ESP8266 genereaza semnal PWM cu rezolutie de zece biti, ridicat la 16 kHz astfel incat zgomotul sa iasa din banda audibila a motoarelor coreless. Pentru a preveni blocarea rotorului la turatii prea mici exista o valoare “idle”; sub acest prag motoarele sunt oprite complet, peste el nu scad niciodata sub aproximativ doisprezece procente din ciclu.

Codul verifica prezenta unui senzor MPU6050 pe magistrala I2C. Daca senzorul raspunde, se face o calibrare scurta a giroscoapelor, iar apoi se ruleaza continuu un filtru complementar care combina giroscopul (98 %) cu accelerometrul (2 %), rezultand unghiurile Pitch si Roll. Daca senzorul nu este detectat programul continua totusi; variabila “mpuOK” ramane falsa si stabilizarea este omisa, drona functionand strict pe comenzile transmise din telefon.

La fiecare iteratie a buclei principale se citesc eventualele unghiuri stabilizate si se aduna cu valorile provenite din slidere. Se obtin astfel trei termeni: pTerm pentru fata-spate, rTerm pentru stanga-dreapta si yTerm pentru rotatie. Acesti termeni se combina cu throttle-ul intr-un mixer clasic de quadcopter: motoarele din spate primesc supliment de turatie cand se inclina in fata, motoarele de pe partea stanga cresc cand se cere deplasare la dreapta, iar perechile de motoare CW si CCW se aduna sau se scad pentru rotatie. Fiecare rezultat este trecut prin functia “mixMotor” care aplica regulile de idle si saturatie, dupa care se trimite catre pinul corespunzator.

Interfata web trimite cereri “fetch” catre doua rute. ”/throttle” modifica puterea principala, iar ”/ctrl” transmite simultan cele trei deviatii Pitch, Roll si Yaw. Logica se executa integral in browser, fara reimprospatarea paginii, iar pachetele HTTP pastreaza conexiunea deschisa pentru latenta minima. In partea de sus a paginii este afisat un mesaj care confirma daca stabilizarea este activa (“MPU OK – stabilisation on”) sau daca zborul este strict manual (“MPU OFFLINE – manual only”).

Auditarile de siguranta sunt simple: lipsa retelei nu blocheaza controlerul, ci pastreaza PWM-ul la ultima valoare; coborarea throttle-ului sub prag aduce toate motoarele la zero; iar esecul initializarii senzorului nu mai conduce la blocarea aplicatiei, doar la dezactivarea filtrului de stabilizare. In ansamblu, firmware-ul inglobeaza patru straturi – retea, senzor, control si hardware – intr-un singur binar capabil sa functioneze indiferent daca MPU-ul este prezent sau nu, facilitand testare rapida, zbor manual si, atunci cand hardware-ul permite, stabilizare automata.

Rezultate Obţinute

Continutul video care suprinde diferite etape din realizarea proiectului poate fi regasit pe pagina de **GitHub** de mai jos. In mod regretabil, transformatorul de curent nu a obtinut un flux constant de curent, variatiile acestuia fiind suficient de mari incat la una dintre testari s-au produs scantei, motiv pentru care am renuntat la el, nedorind ca proiectul sa devina un fire hazard.

Dintr-un motiv pe care nu mi-l pot explica, odata cu lipirea firelor motoarelor, la primul test, modulul

MPU 6050 a cedat, acesta nemaiputand fi detectat de microcontroller. In ciuda unor verificari extinse, a refacerii conexiunilor fizice si a activitatii vizibile ale ledului de la senzor, acesta a ramas inaccesibil de pe microcontroller. Acest lucru m-a determinat sa adaptez codul si la o versiune manuala de control, pentru a putea face un demo chiar si in lipsa stabilizarii.

In cadrul ultimului test, suprins de altfel si intr-unul din videoclipuri, se poate vedea cum un motoras a cedat, iar celalalt cedeaza in timpul filmarii. Asta se dovedeste a fi natura efemera a motoarelor coreless, care tind sa aiba o durata de viata extrem de scurta.

Concluzii

Concluzia acestui proiect, de altfel, pe atat de interesant pe cat a fost de ambitios, este ca, din pacate, socoteala de acasa nu e intotdeauna ca aceea din targ. In definitiv, atat lipsa de experienta, cat si limitarile hardware-ului si-au spus cuvantul, intrucat, in ciuda precautiilor si masurilor luate, am intampinat dificultati la concretizarea proiectului, dupa cum se poate vedea si in videoclipuri. Acesta a fost, insa, un prilej excelent de invatare.

Download + Jurnal

Fisierele proiectului si jurnalul progresului surprins in **commit-uri** sunt regasite pe urmatorul link de **GitHub**:

- **https://github.com/AlexChiorean/Quadcopter**.

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/alexandru.chiorean

Last update: **2025/05/30 00:30**