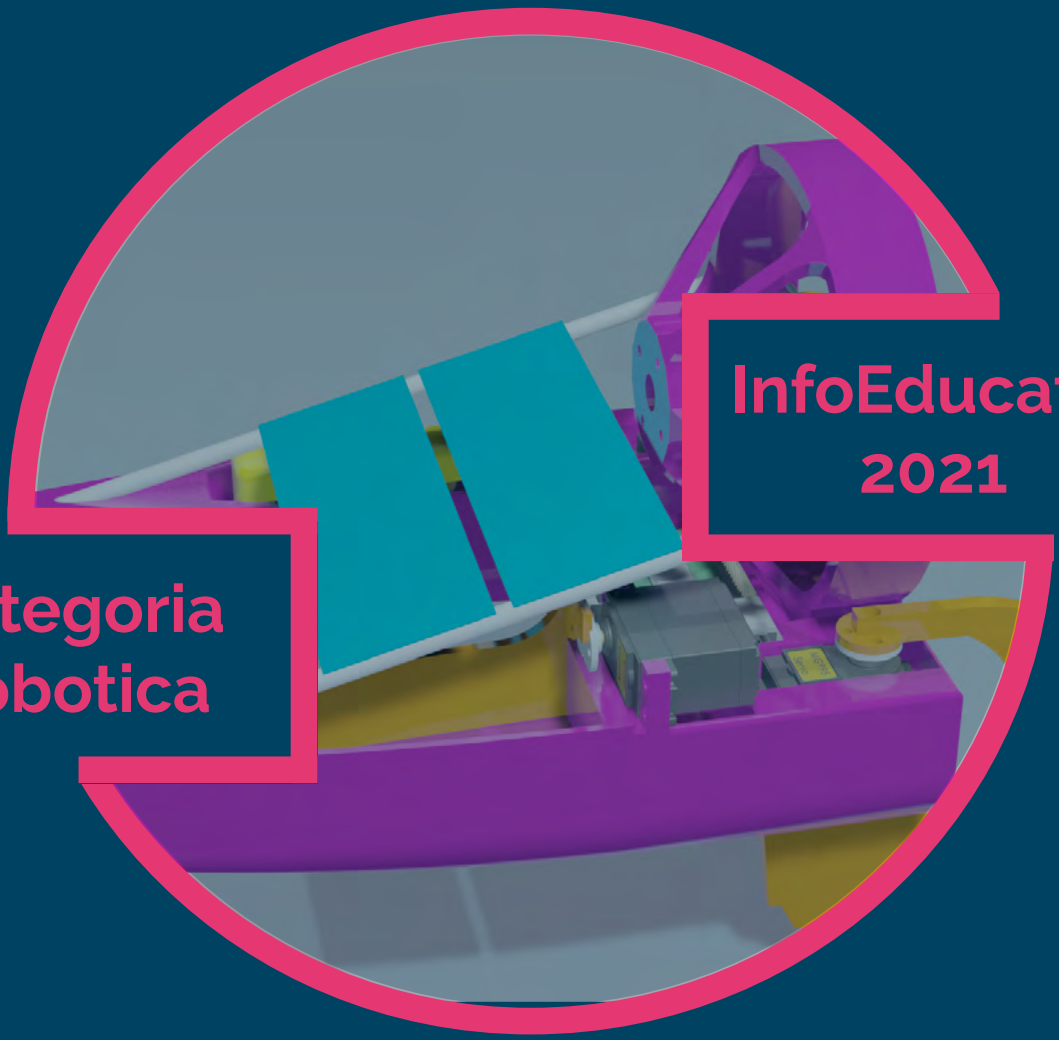
Barcainatorul



InfoEducatie 2021

Categoria Robotica

Noi usuram pescuitul intr-un mod sigur si inteligent

Cuprins documentatie

Barcainatorul

Proiectul a fost creat pentru concursul InfoEducatie - Categoria Robotica
GitHub: cod

Alexandru Oprea
clasa a 11-a

Oana Datcu
clasa a 10-a

Ce este Barcainatorul?

Barcainatorul este un proiect realizat pentru Olimpiada de Inovare și Creație digitală - INFOEDUCAȚIE 2021, categoria Roboti. Acest proiect are ca scop principal garantarea pescuirii eficiente, reducerea cantitatii de energie și de efort și a impactului asupra apei in contrast cu echipamentele tradiționale de momit.

Scop

Intrebarea secolului, ce face un pescar pe o balta? Raspunsul pare unul evident in prima instanta, pescuieste, dar este mult mai mult de atat, putem spune ca este si o arta si un sport simultan. Speranta, pierdedre, frustrare si vointa sunt cuvintele care descriu cel mai bine pescarul.

Barcainatorul este eroul povestii. Vine de nicaieri (doar selectezi traseul din aplicatie), se va deplasa rapid la punctul final si va arunca momeala din cuva pozitionata strategic, ademenind pana si cel mai feroce peste, usurand treaba pescarilor.

Nu v-am convins inca? Cititi mai departe pentru a afla de ce acest erou este protagonistul tuturor povestilor pescaresti.

Principii

Barcainatorul este un robot plutitor, de tip barca Catamaran. Modul principal de translatie este asigurat de un motor fara perii, pozitionat in spatele barcii folosind propulsia aerului cu ajutoul unei eleci de 20 cm. Carmele barcii sunt pozitionate coplanar cu planul elicii, aceste find actionate cu ajutorul a 2 servomotare. Unghiul de atac al barcii este controlat cu ajutorul unui alt servomotor modificand astfel directia fluxului de aer. Un al patrulea servomotor se ocupa cu deschiderea si inchiderea cuvei de mancare desfacand trapa. Toate acestea fiind alimntate cu ajutorul a doua surse, cea principala, un acumulator LiPo cu 3 celule de 3 000 mAh si un panou solar pentru regimul de urgenta.

Creierul barcii este reprezentat de un mini computer Raspberry Pi 3B+, care proceseaza datele colectate de senzori si informatiile primite de la statia de comanda. Principalul indice furnizat de senzori este pozitia curenta a robotului folosind un senzor GPS, acesta servind la stabilirea traseului barcii. Pentru a asigura stabilitate in timpul deplasarii, orientarea barcii este corectata cu ajutorul unui senzor de tip giroscopic. In final traseul este transmis Barcainatorului prin modulul wireless NRF24L01+.

Baza de control este compusa dintr-un microcontroler Seeeduino XIAO si un transmitaor NRF24L01+. Aceasta comunica cu aplicatia de pe telefon prin serial actionand ca o punte intre barca si telefon.

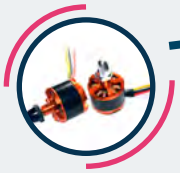
In ceea ce priveste alimentarea robotului tensiunea de 11,1 V a acumulatorului a fost transformata in 5 V folosind un modul step-down, iar cea a panoului solar, fiind variabila a fost mentinuta tot la 5 V de catre un modul step-up.x

PIESE FOLOSITE

Pentru a asigura o constructie rapida si sigura a robotului, am proiectat initial ideiile in Autodesk Inventor. Dupa mai multe prototipuri am ajuns la o versiune finala cu un design eficient si optim. Etapa urmatoare a constat in printatarea pieselor cu ajutorul unei imprimante 3D. Timpul total de printare a fost de aproximativ 60h, iar materialul folosit a fost in totalitate PETG. Pentru a diminua frecarea cu apa, astfel imbunatatind hidrodinamica fuselajului de tip catamaran, suprafta ce intra in contact cu apa a fost prelucrata cu chit si vopsea. In plus barca dispune de 2 tije de aluminiu oferindu-i rezistenta in timpul functionarii. In concluzie, designul barcii este unul industrial ce poate fi replicat in timp util si produs pe scara larga.

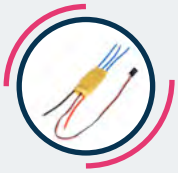
ACUMULATOR

Scop:
Sursa principala de alimentare

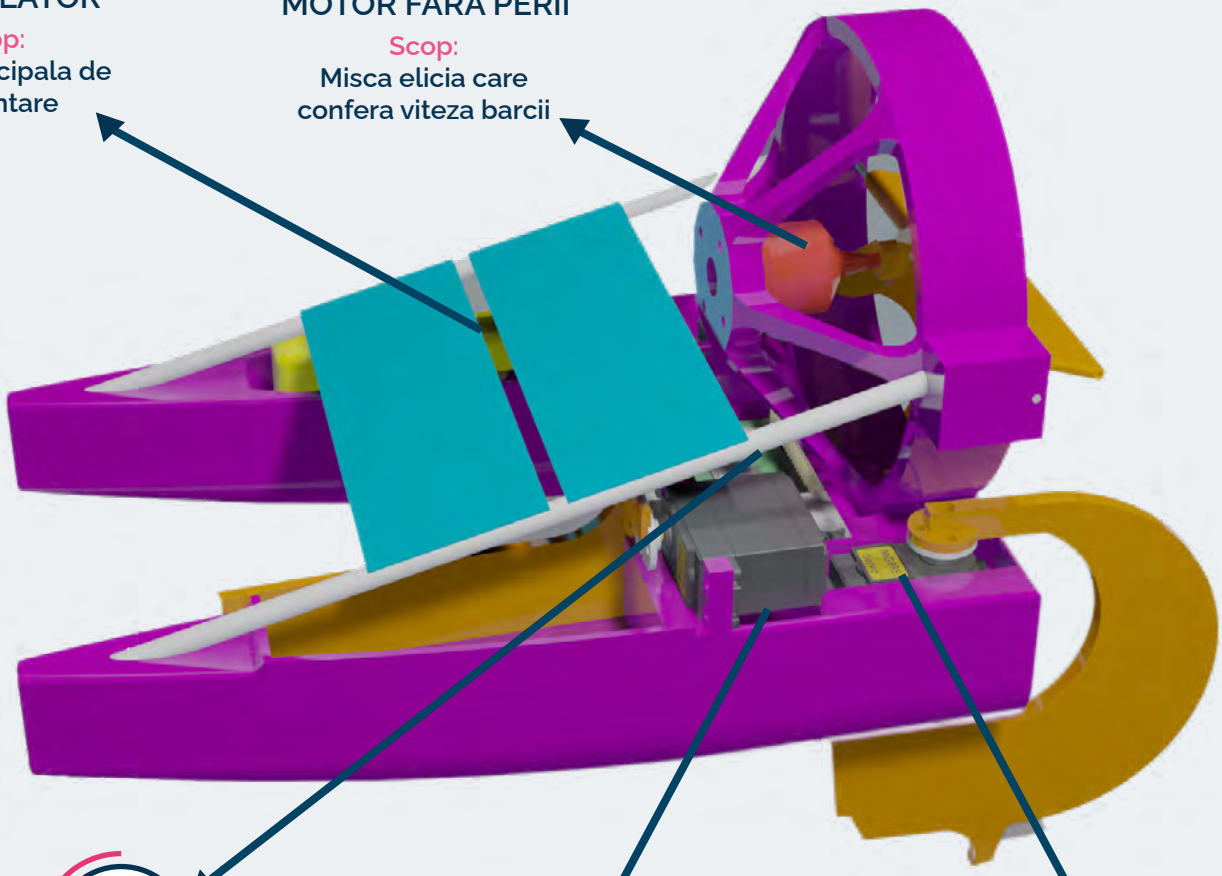


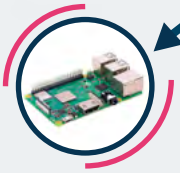
MOTOR FARA PERII

Scop:
Misca elicia care confera viteza barcii



ESC 30A - PT MOTOR(ESC)





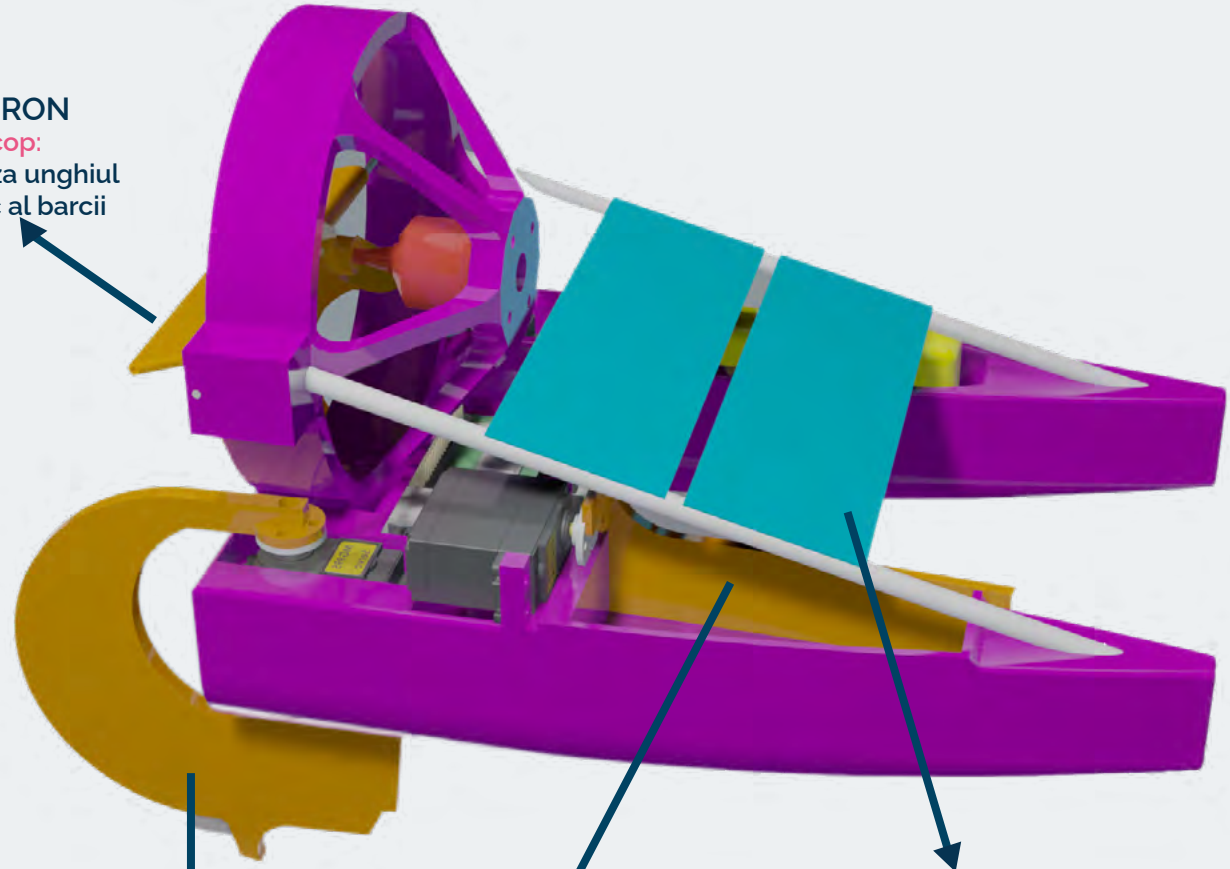
RASPBERRY PI 3 MODEL B+

Scop:
Placa de control a barcii

SERVOMOTOR

Scop:
Deschide si inchide cuva de mancare

Scop:
Setare pozitie carme

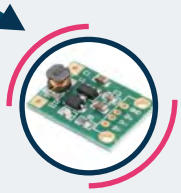


ELERON
Scop:
Ajusteaza unghiul de atac al barcii

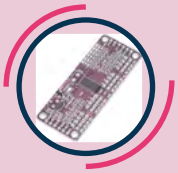
CARMA DREAPTA
Scop:
Viraje

PERETE CUVA
Scop:
Inchide si deschide cuva de mancare

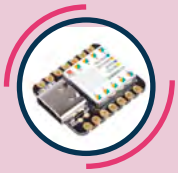
PANOU SOLAR
Scop:
Sursa secundara de putere



SURSA PANOURI SOLARE

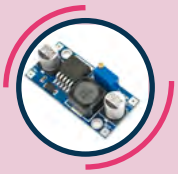


MODUL CONTROL PWM PRIN I2C



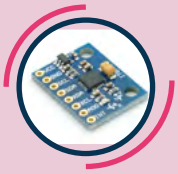
MICROCONTROLER PT TRANSMITAROR

MODUL COMUNICARE



SURSA PT RASPBERRY PI PS

GPS



GIROSCOP SI ACCELEROMETRU

CUM AM ALES PIESELE?

	AVANTAJE	DEZAVANTAJE
BATERIE LIPO 3S1P	- capacitate ridicata - 3000mAh - timp scazut de incarcare - 1 ora la rata de incarcare 1C - curent de scurt-circuit ridicat - 60A - tensiune nominala de 11.1V - densitate energetica mare	- greutate ridicata - apx 350g - pericol de incendiu in caz de scurt-circuit
ESC 30A	- curent maxim de 30A - radiator inclus - control PWM - sursa BEC de 5V inclusa in modul	- nu dispune de protocoale speciale (cum ar fi BLHeli sau DShot)
PWM MULTIPLEXER	- 16 canale disponibile - comunicare prin I2C - frecventa PWM ajustabila	- dimensiuni mari
SEEEDUINO XIAO	- dimensiuni foarte mici - consum redus de energie - performanța ridicată datorită microcontrolerului pe baza de ARM Cortex-M0+ - suport pentru protocoale populare (I2C, UART, SPI, PWM, Analog, Digital) - foloseste framework-ul Arduino	- nu include găuri pentru fixat

- frecventa de citire de 10 Hz - comunicare prin UART-Serial - dimensiuni reduse - raport pret/calitate bun	GPS	- precizie de doar 1 m - barca trebuie sa fie in camp deschis - senzorul trebuie amplasat cu vedere spre cer - performanta variaza cu numarul de sateliti detectati
- comunicare prin SPI distanța de acoperire de pana la 1 km - permite o programare facilă - atena omnidirectioanala - pret scazut	MODUL DE COMUNICARE WIRELESS NRF24L01+	- dimensiuni mari in comparatie cu module similare - comsum mai ridicat de enrgie in modul TX - lucreaza la frecventa de 2.4Ghz => permite comunicare doar in linie dreapta
- comunicare prin I2C - frecventa de citire ridicată - imbina trei senzori intr-un singur modul	GIROSCOP	- precizie medie - datele citite variaza cu temperatura
- eficienta de 96% - reglare automata a tensiunii de iesire (5V) - dimensiuni mici - putere maxima de 5W	SURSA PANOURI SOLARE	- necesita filtraj suplimentar (condesatori) pentru a obtine o tesniune stabila
- eficienta 96% - tensiune ajustabila - curent maxim de iesire 5A	SURSA RASP	- dimensiuni mari - degaja caldura in consum ridicat

- durata de viata ridicata - viteza de rotatie mare - cuplu mare - motor fara perii	MOTOR BR2212	- necesita curent trifazic pentru a putea fi acționat - corpul motorului este chiar rotorul
- rezistenta mecanica in timp - rezistenta la UV - rezistenta la temperatura	FILAMENT PETG	- se printeaza la temperatura mai ridicată (240 grade C) - necesita pat incalzit cu suprafata de sticla
- eficienta mai mare in comparatie cu elicii mai mici - pret scazut, se poate inlocui usor	ELICE 8060	- nu rezita la mai mult de un impact/oprire brusca
- cuplu si viteza ridicate	SERVOMOTOR MG996	- consum ridicat de energie
- performanta ridicata (4 nuclee la frecventa de 1.4 GHz) - suport pentru protocoale populare (I2C, UART, SPI, PWM, Analog, Digital) - sistemul de operare pe baza de Linux	RASPBERRY PI 3B+	- consum de energie ridicat (in medie 5W) - dimensiuni mari

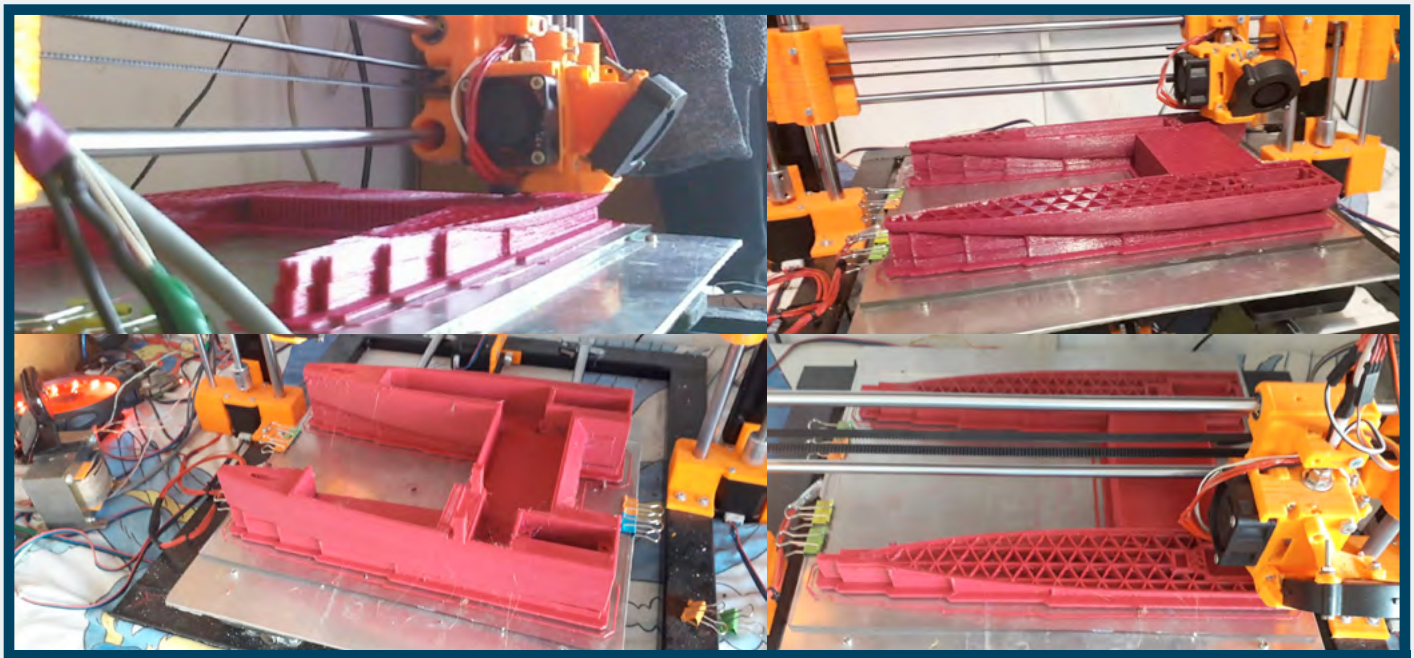
Etapele parcurse

Partea de constructie a fost cea mai distractiva - a trebuit sa printam toate componentele realizate in etapa de proiectare,iar dupa aceea am rezolvat probleme de etansare a componentelor parcurgand un procedeu industrial in repetate randuri.

Printare 3D

Prima etapa a construirii Barcainatorului a fost printarea 3D a corpului, suportului de motor si de elice, carmeleor, a eleronului si a capacului cuvei de mancare. Dupa o printare de aproximativ 60 de ore am putut spune ca barcuta noastra era gata de atac.

PS: Printarea a fost facuta cu ajutorul unei imprimate 3D homemade cu filament PETG.



Pintare in derulare

Pregatirea pieselor

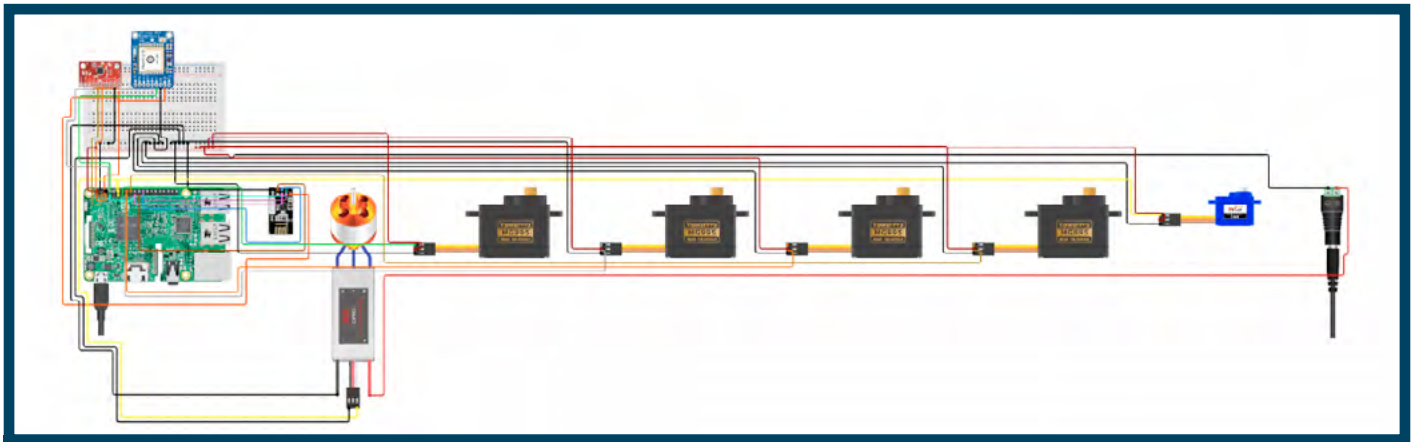
Dupa inlaturarea tuturor suportilor am slefuit corpul pentru a putea acoperii partea inferioara cu chit, astfel am redus considerabil riscul ca componentele din interior sa fie afectate din cauza infiltartilor apei. Dupa chituire am inalaturat surplusul cu ajutorul smirgherului si am repetat procedura de inca 3 ori, iar in final am adugat un strat de vopsea.

Netezire, chituire si vopsire

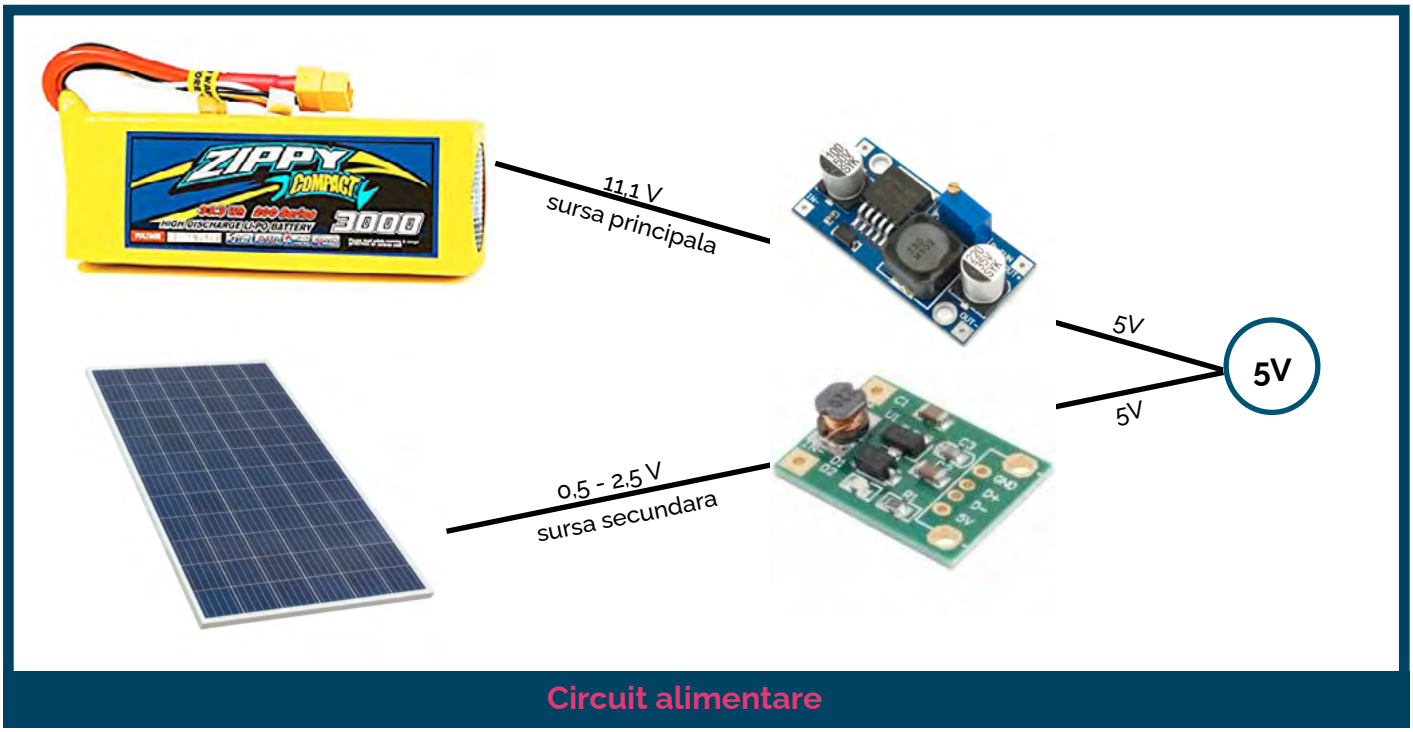
Electronica

Pentru a realiza sarcinile intr-un mod eficient, robotul dispune de mai multi senzori controlati de un microcomputer(Raspberry Pi 3B+). Un prim senzor este modulul GPS (BN220) care furnizeaza latitudinea si longitudinea curenta cu o precizie de 1 metru. Acesta este vital proiectului deoarece ne permite sa calculam directia si distanta pana la punctele din traseu. Totusi, pentru a ne orienta in directia respectiva, folosim un modul giroscopic (MPU6050) care ne permite sa determinam atat directia de inaintare cat si inclinatia barcii .

Comunicarea cu robotul este relizata printr-un set de transceivere (NRF24L01) ce au o raza de actiune de 1km, unul fiind montat pe barca, iar celalalt ramanand pe uscat la statia de control. Statia este reprezentata de un microcontroler ARM(Seeeduino XIAO) de mici dimenisuni, avand rolul de nod intre telefonul mobil si barca. Informatiile sunt generate cu ajutorul unei aplicatii dezvoltate in Android Studio folosind frameworkul Google Maps si mai apoi codificate si transmise prin USB serial.



Circuit

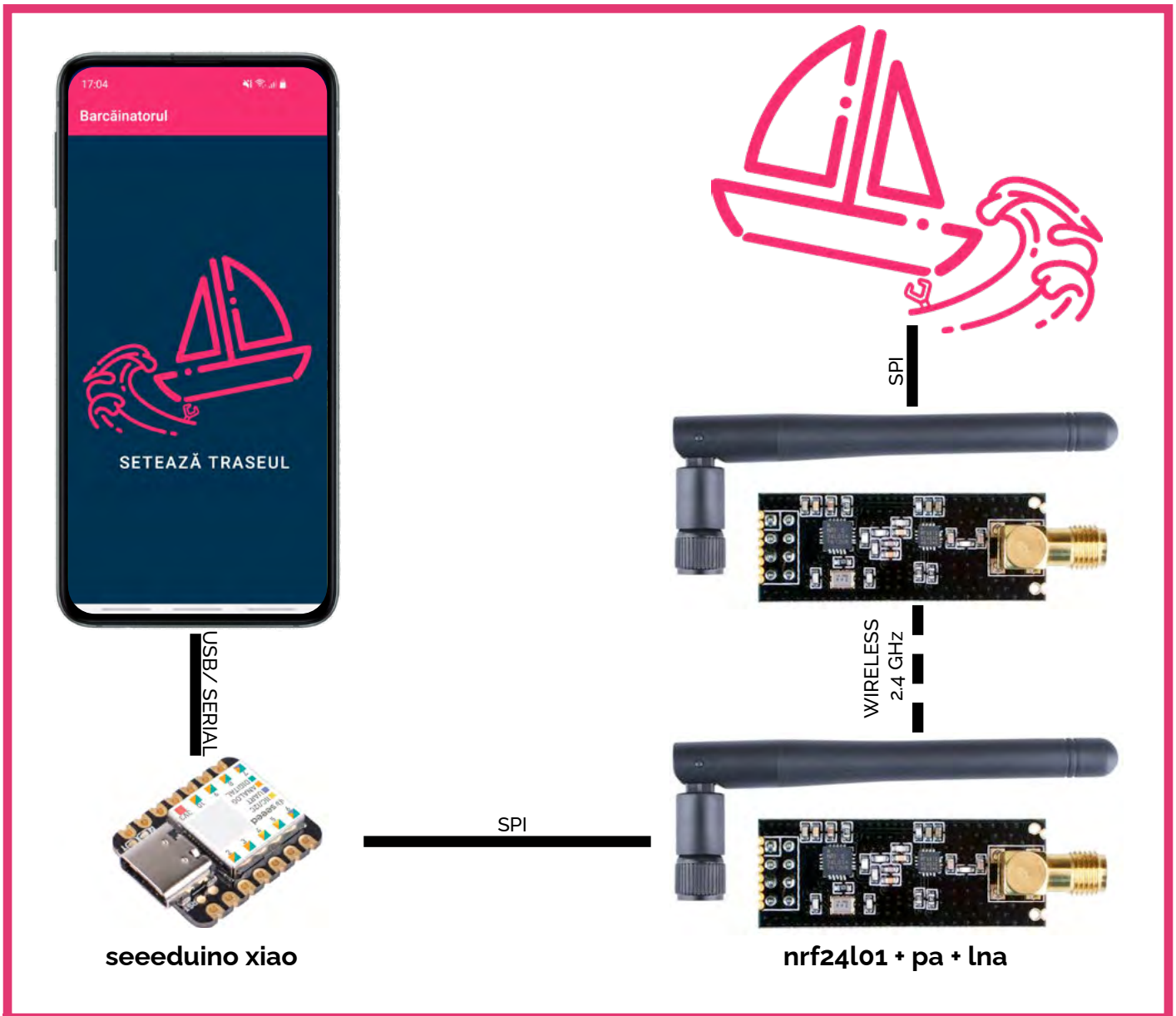


Circuit alimentare

Scrierea codului

Pentru a dezvolta codul am folosit 3 limbaje: java, c++ si python.

Codul a fost impartit in doua sectiuni: aplicatia de telefon prin care utilizatorul poate comunica cu barca si codul propiu-zis care transforma robotul dintr-un obiect static intr-o barca complet functionala si autonoma.



Modul de conumicare dintre aplicatie si barca

Pentru a dezvolta aplicatia, prin care utilizatorul poate selecta traseul barcii, am utilizat limbajul java si am folosit platforma AndroidStudio.

Cu ajutorul API-ul specific crearii de harti oferit de Google (google maps) am putut dezvolta interfata prin care adaugam puncte pe harta transmitand astfel coordonatele drumului dorit robotului.

Algoritm principal

Codul principal al robotului (main.py) este scris in python si se bazeaza pe librariile din colectia "adafruit" folosind framework-ul CircuitPython. Acestea contin functii pentru comunicarea cu senzorii (UART, SPI, i2C) dar si obiecte primitive reprezentand motoarele si servomotoarele facilitand astfel programarea. Algoritmul este impartit in mai multe etape. La lansare, se asteapta primirea unui traseu de la statia de control. Dupa primirea instructiunilor, se calculeaza pozitia curenta obtinuta de la senzori si se actioneaza ficare motor/servomotor in functie de raspunsurile algoritmului PID si a legilor de traslatie preprogramate. Totusi pentru a determina pozita si distanta pana la punctul tinta folosind GPS-ul a fost necesara implementarea legii Haversine (determinarea distantelor pe o sfera). Daca s-a ajuns la destinatie se actioneaza servo-ul cuvei si se seteaza pozitia initiala ca destinatie urmatoare, reluand astfel operatiile de mai sus. In final, cand barca ajunge de unde a plecat, se deconecteaza senzorii si se iese din program.

Baza de control este programata in C++ folosind framework-ul Arduino. Acest cod (main.cpp) actioneaza similar cu un repeter: se preiau informatiile primite prin serial de la aplicatie, se stocheaza intr-un buffer si se transmit mai departe catre barca.

In final, dorim sa precizam ca atat codul barcii cat si codul statiei de control respecta un stil modern de programare, avand explicatii si comentarii pe tot parcursul algoritmilor, iar bucatiile de cod care sunt apelate de mai multe ori pe parcursul programului au fost organizate in functii si clase, usurand astfel depanarea si modificarea programului.

PID

Care este primul cuvânt care îți apare în cap când auzi PID? Precizie? Excelență? O nevoie absolută? Pentru noi acest cuvânt este proporțional-integral-derivativ. Știm ca de fapt sunt trei cuvinte, dar aceasta sintagma este intreg algoritmul.
Principiul algoritmului este de a corecta o variabilă (de exemplu: unghi, poziție, viteza) pe baza feedback-ului pe care îl obține.
După cum sugerează și numele, magia se face în trei etape: proporționala, integrala și derivativa, de obicei într-o buclă de control. La sfarsitul fiecărui ciclu îi setam o valoare țintă și starea curentă a sistemului nostru. Pe baza acestor date, PID-ul returneaza modificările ce trebuie făcute pentru a ajunge la valoarea ținta. La început, calculează eroarea curentă a sistemului si o furnizeaza celor 3 stagii. La prima etapa proporțională, eroarea se înmulțește cu o constantă (denumită de obicei Kp), oferind astfel o corecție proporțională cu eroarea curentă. Etapa de integrare ia în considerare, pe lângă eroarea curentă, si eroarea în timp. Putem vizualiza acest efect ca un trend: dacă tendința este ca eroarea să crească, atunci trebuie să combatem acest lucru prin creșterea corecției, în caz contrar, reducem corecția pentru a nu ne depăși ținta. Impactul acestui stagiu este controlat cu ajutorul unei constante (de obicei Ki). Ultima etapă este cea derivativa. Acesta nu încearcă să elimine complet eroarea, dar încearcă să facă rata de schimbare a erorii (derivata acesteia) nula. De asemenea, impactul acesteia este gestionat de o constantă (Kd). În final, prin ajustarea acestor trei constante (Kp, Ki și Kd) putem obține un sistem fără oscilații.
Toate aceste calcule sunt executate pentru fiecare axa de control (inaintare, directie si unghi de atac), iar pentru a simplifica programarea am scris o clasa separata in python (pid.py) care contine informatiile necesare si functiile principale executarii programului.

BARCAINATORUL IN ACTIUNE