



Documentation technique - BATONOME

Table des matières

Documentation technique - BATONOME	1
Introduction	3
Composant	3
Carte Mère du bateau	3
Communication Bateau - Interface Graphique	3
Interface graphique	4
Principe	4
Settings	5
Navigation	6
Girouette	6
Principe :	6
Schéma	7
IMU et détermination du CAP	8
Introduction	8
Présentation du module	8
Utilisation du module pour obtenir le cap	10
Algorithme de navigation	12
Principes de base	12
Choix du cap	12
Commandes du bateau	14
Réalisation	14
Carte de sûreté switch	15
1.Objectif	15
2. Conception	15
3. Tests et validation	17
4. Conclusion et axes d'améliorations	17

Introduction

Le présent document est consacré à l'élaboration détaillée du projet réalisé dans le cadre de nos cours : la transformation d'un voilier radiocommandé de classe 1m en un bateau autonome.

Pour faciliter la compréhension de notre travail, chaque étape de notre projet a été documentée de manière détaillée. Cela va de la modification physique du voilier, en passant par l'installation et la configuration de l'équipement électronique nécessaire, jusqu'à la mise en place du logiciel d'autopilotage.

Notre objectif principal avec cette documentation est de fournir une ressource complète pour quiconque s'intéresse à notre projet, que ce soit pour la reproduction, l'adaptation ou simplement pour approfondir leur compréhension de la technologie sous-jacente.

Tous les codes, schémas, plans et autres ressources liées à ce projet sont accessibles via notre dépôt GitHub. Vous pouvez accéder à ce dépôt à l'adresse suivante :

<https://github.com/Sacha-Cantat/Batonome>

En parcourant cette documentation, n'hésitez pas à consulter notre dépôt GitHub pour une compréhension plus approfondie de chaque composant du projet. Nous vous invitons également à partager vos idées, commentaires et suggestions, car nous croyons fermement à l'amélioration continue par le biais de la collaboration.

Bonne lecture, et bon voyage dans l'univers de la navigation autonome !

Composant

STM32 L432kc
Adaptateur Xbee USB 11293 (2)
Module Xbee série 2C XB24CZ7WIT-004 (2)
GPS-RTK-SMA ZED-F9P
IMU CMPS12
capteur électromagnétique AS5600.

Carte Mère du bateau

Communication Bateau - Interface Graphique

Le voilier communique avec un ordinateur au sol à l'aide d'une liaison Zigbee. Pour cela, on utilise les émetteurs Zigbee configurés en mode transparent.

Nous avons mis au point un protocole de communication qui fonctionne en polling. C'est la station au sol qui est à l'initiative de la communication.

A0	Polling PC
[Trame Hexa info Bateau]	ACk polling Batonome
A0	Polling PC

Le voilier renvoie dans cette trame, sa position, la valeur d'angle du vent, la valeur d'angle du cap.

D'autres commandes existent :

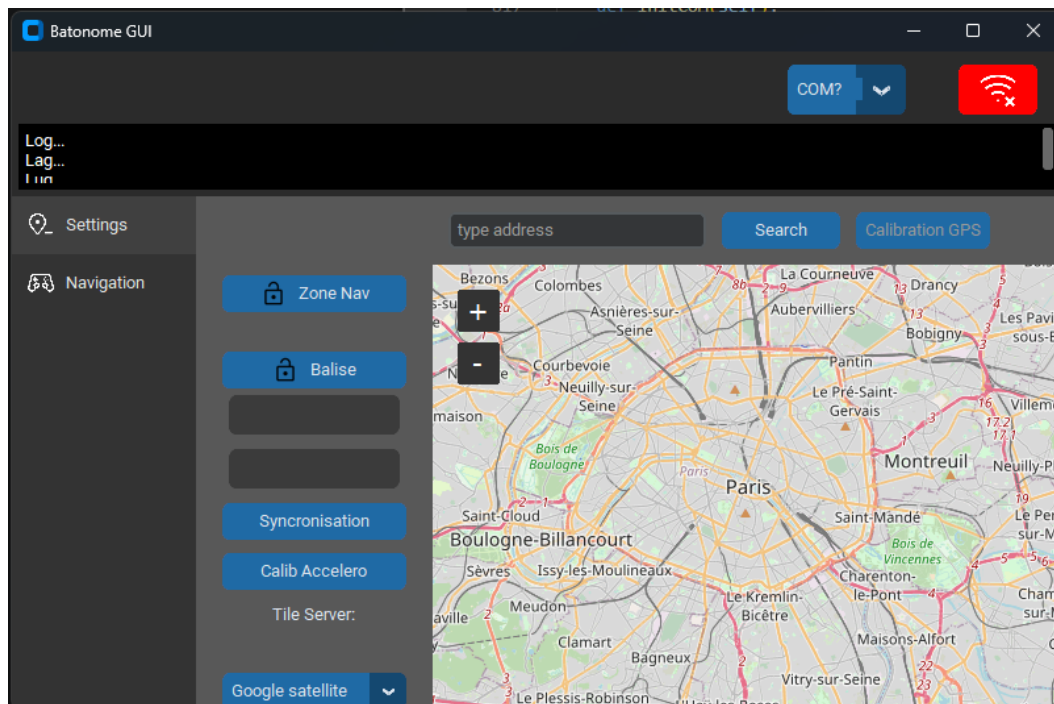
B0	Envoi des points GPS du périmètre
C0	Touche RIGHT appuyée sur le clavier
C1	Touche LEFT appuyée sur le clavier
D0	Touche UP appuyée sur le clavier
D1	Touche DOWN appuyée sur le clavier
C2	Touche SPACE appuyée sur le clavier
D2	Touche ENTER appuyée sur le clavier
E0	Touche T appuyée sur le clavier
F0	Demande de calibration IMU

Interface graphique

Principe

Nous avons mis en place une interface graphique. Cette interface graphique permet de donner la configuration au bateau et de suivre son fonctionnement pendant la navigation.

Cette interface graphique a été développée en python et se base sur le module Customtkinter.



On retrouve deux onglets :

- Settings, pour gérer la configuration
- Navigation, pour suivre le bateau en fonctionnement.

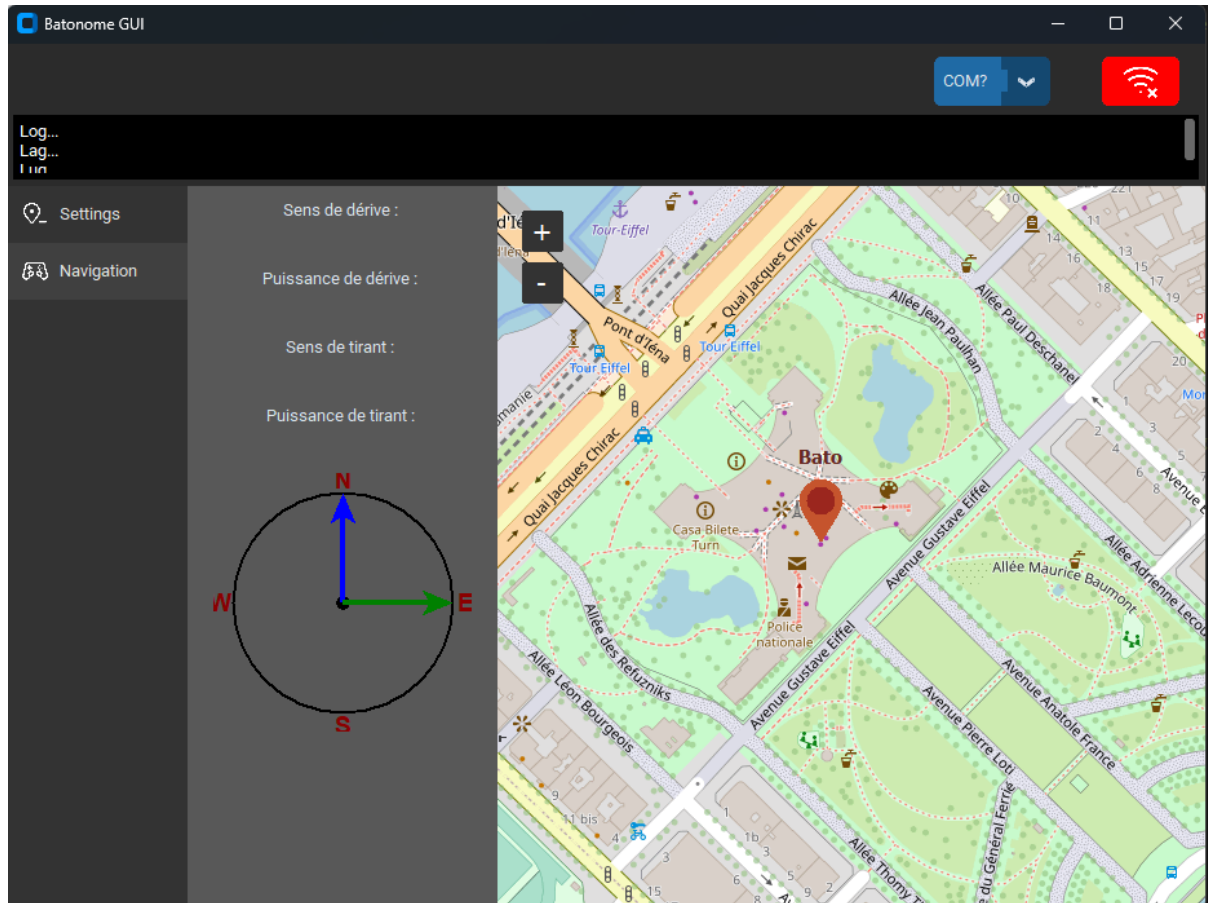
Settings

Dans cet onglet, nous pouvons :

- Initialiser la connexion au voilier en sélectionnant le port COM sur lequel est branché le module xbee.
- Définir le périmètre de navigation du voilier en cliquant avec le pointeur de la souris pour enregistrer les points GPS de 10 points. Cette fonctionnalité est fonctionnelle mais désactivée. On bloque le périmètre en cliquant sur le bouton Zone Nav

- Définir le point GPS de la balise en utilisant la souris de la même manière après avoir verrouillé la zone de navigation.
- Envoyer la configuration au voilier en appuyant sur le bouton de synchronisation.

Navigation



Dans cette onglet, nous pouvons :

- Visualiser la position GPS du voilier.
- Visualiser la position de la girouette et le cap du voilier

Girouette

Principe :

Une girouette est un dispositif utilisé pour déterminer la direction du vent. Dans notre cas, nous avons conçu une girouette imprimée en 3D avec un capteur électromagnétique AS5600. L'AS5600 est un capteur de position angulaire à 12 bits qui fonctionne par détection de champ magnétique. Il dispose d'une interface de communication I2C que nous

utilisons pour lire les données de position angulaire. Voici comment notre système fonctionne:

La girouette est montée de manière à pouvoir tourner librement autour d'un axe vertical. Nous avons utilisé des matériaux plastiques (ABS et PLA) de l'impression 3D pour construire la girouette, ce qui la rend légère et réactive aux vents même faibles.

Afin de stabiliser la girouette et de minimiser les mouvements parasites dus à la navigation, nous avons ajouté un contrepoids à la conception. Cela aide à maintenir la girouette dans une orientation stable et à éviter les fausses indications dues aux mouvements de tangage et de roulis du bateau.

Le capteur AS5600 est monté sur l'axe de rotation de la girouette. Il est basé sur la technologie Hall, qui permet de mesurer le champ magnétique. Dans notre cas, nous avons placé un petit aimant sur l'axe de rotation de la girouette, créant ainsi le champ magnétique que le capteur détecte.

Lorsque le vent souffle, il fait tourner la girouette. La rotation de l'aimant change le champ magnétique, ce que le capteur AS5600 détecte. Il convertit ensuite cette mesure en une valeur numérique représentant l'angle de rotation.

Nous utilisons l'interface I2C du capteur pour lire cette valeur numérique. L'interface I2C est un moyen standard de communication entre les composants électroniques qui permet d'échanger des données sur deux fils de signal.

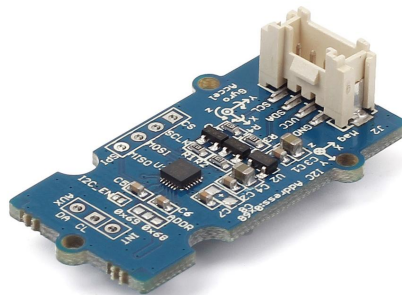
Cette sortie numérique nous donne l'angle de rotation de la girouette, que nous utilisons pour déterminer la direction du vent. Par exemple, nous pouvons configurer le système pour qu'il associe un angle de 0 degré à une direction vers le nord, et d'autres angles à d'autres directions.

Grâce à cette technologie, nous avons créé une girouette numérique précise qui peut détecter avec précision et rapidement les changements de direction du vent, tout en étant robuste face aux mouvements de navigation. C'est une excellente utilisation de la technologie de capteur à effet Hall, de l'impression 3D et de la communication I2C.

IMU et détermination du CAP

Introduction

Le module IMU a été changé en cours de route, à la fin du projet. En effet, nous utilisons le module IMU grove, fonctionnant en I2C, et basé sur le MPU9250 de Invensense, un capteur combinant un accéléromètre, un magnétomètre et un gyroscope.



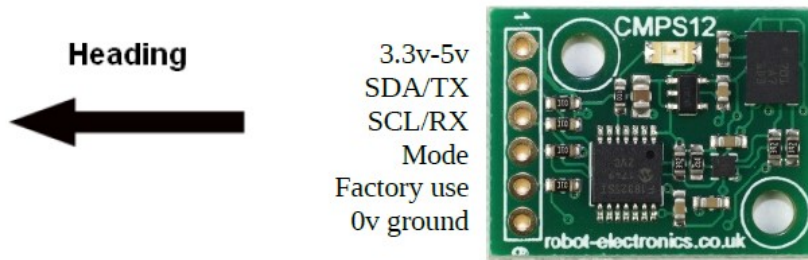
Pour déterminer la direction du bateau, l'opération consiste à fusionner les données de ces différents capteurs pour obtenir une dérive en degré par rapport au nord magnétique terrestre et ainsi se servir de cette donnée dans l'algorithme de navigation.

Néanmoins, faire une telle stratégie de fusion de capteur est très compliqué (calcul des angles d'euler, quaternion, et corriger les différentes dérives propre à chaque capteur (accélération, vitesse angulaire, champ magnétique)).

On est donc passé sur un module CMPS12, qui effectue cette fusion de capteurs directement sur le module.

Présentation du module

Ce module fonctionne en I2C et en liaison série. Nous l'avons donc utilisé en I2C.



L'adresse I2C permettant d'identifier le module sur le bus est l'adresse 0xC0.

Pour lire les valeurs du cap avec précision, nous préférons utiliser les registres 0x02 et 0x03, le registre 0x02 étant l'octet de poids fort. Ainsi nous récupérons 1 octet dans chaque registre

Ainsi, une simple opération va nous permettre de récupérer une valeur comprise entre 0 et 3600, grâce à la valeur des 2 octets récupérés.

Register	Function
0x00	Command register (write) / Software version (read)
0x01	Compass Bearing 8 bit, i.e. 0-255 for a full circle
0x02, 0x03	Compass Bearing 16 bit, i.e. 0-3599, representing 0-359.9 degrees. register 2 being the high byte. This is calculated by the processor from quaternion outputs of the BNO055

Le module se calibre automatiquement à l'allumage. Néanmoins on peut vérifier sa calibration en venant interroger le registre 0x1E pour récupérer l'état de la calibration.

0x1E	Calibration state, bits 0 and 1 reflect the calibration status (0 un-calibrated, 3 fully calibrated)
------	--

L'octet récupéré se décompose comme ceci:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cal status	System calibration		Gyro calibration		Accelerometer calibration		Magnetometer calibration	

Voici comment aider la calibration des différents capteurs de l'IMU à l'allumage:

Gyro – calibrated by the CMPS12 being in a stationary state.

Accelerometer – calibrated by tilting the module to roughly 45 and 90 degrees on one axis.

Magnetometer – a few random movements easily calibrates the CMPS12

Utilisation du module pour obtenir le cap

Ainsi pour utiliser l'IMU et retrouver le cap, nous commençons par déclarer l'adresse I2C du module:

```
#define IMU_ADDRESS 0xC0
```

Puis nous déclarons nos registres:

```
#define IMU_1 0x02
```

```
#define IMU_2 0x03
```

Nous déclarons nos buffers qui vont contenir la valeur contenue dans nos 2 registres d'angles.

```
uint8_t high_byte[1], low_byte[1];
```

Nous déclarons également notre variable qui va contenir notre cap:

```
uint16_t cap = 0;
```

Ensuite nous venons lire la valeur des registres (bien mettre la bonne adresse I2C du module - ici notre module I2C est connecté sur l'I2C n°1 du microcontrôleur):

```
HAL_I2C_Mem_Read(&hi2c1,IMU_ADDRESS,IMU_1,1,high_byte, 1,HAL_MAX_DELAY);
```

```
HAL_I2C_Mem_Read(&hi2c1,IMU_ADDRESS,IMU_2,1,low_byte, 1, HAL_MAX_DELAY);
```

Ensuite une opération constituée d'un décalage logique nous permet de retrouver notre valeur entre 0 et 3600.

```
cap = ((uint16_t)high_byte[0] << 8) | low_byte[0];
```

Nous divisons ensuite cette variable par 10, pour obtenir un cap entre 0 et 360°:

```
cap = cap/10;
```

Notre variable cap contient maintenant notre orientation entre 0 et 360. 0 étant le nord magnétique.

Attention, il est préférable de faire une calibration software du nord magnétique. Nous avons remarqué que les variations de mouvements étaient relativement bien gérés par l'IMU, toutefois la détection du nord magnétique peut être imprécise. Il est donc conseillé de positionner l'IMU face au nord à l'aide d'une boussole externe et de prendre cette référence comme nord magnétique, en l'ajoutant comme un offset à la valeur du cap déjà établi.

Algorithme de navigation

Principes de base

Face à une contrainte de temps en phase finale de notre projet, nous avons opté pour un algorithme de navigation plus simplifié. Cette stratégie nous a permis d'assurer un déploiement aisé et une correction rapide et efficace dès lors que notre navire était opérationnel.

Choix du cap

Nous avons défini trois scénarios de navigation distincts :

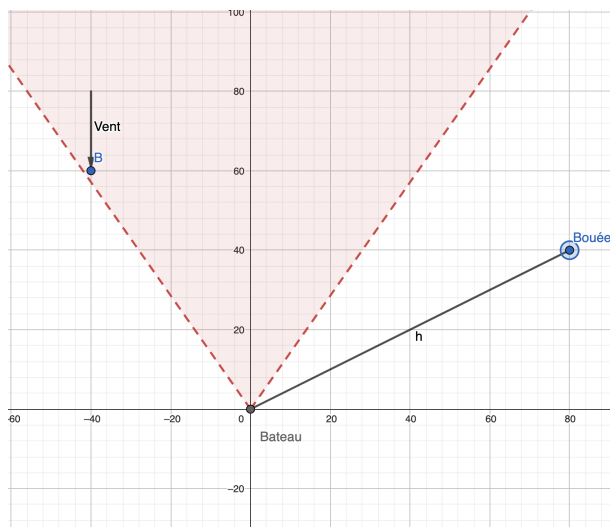
- Face du vent
- Dos au vent
- Autres situations.

Dans le cas où le bateau est confronté à une situation autre que face ou dos au vent, nous ajustons simplement le cap du bateau directement vers la bouée.

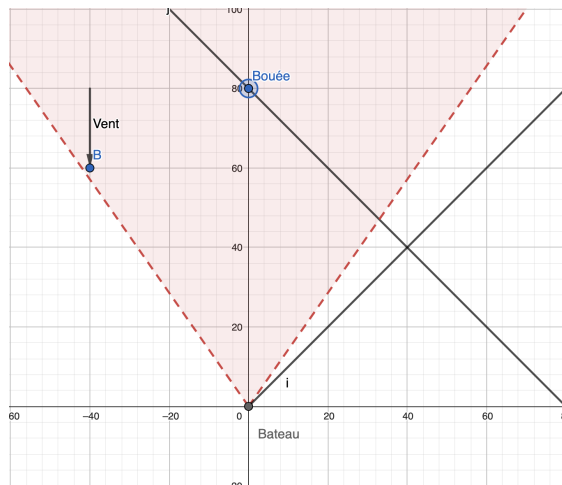
Lorsque le bateau fait face au vent, notre stratégie est de définir le cap à : vent $+45^\circ$. Cette manœuvre, appelée "prendre un bon près", permet au bateau de maintenir une bonne vitesse. Une fois que le bateau a parcouru une distance suffisante, de sorte que la bouée soit désormais à 90° de lui, il change de cap pour se diriger vers la bouée, en maintenant toujours le vent à 45° .

Dans le cas d'un vent arrière, nous appliquons une stratégie similaire à celle du vent de face, à la différence près que le cap est défini à : vent $+135^\circ$.

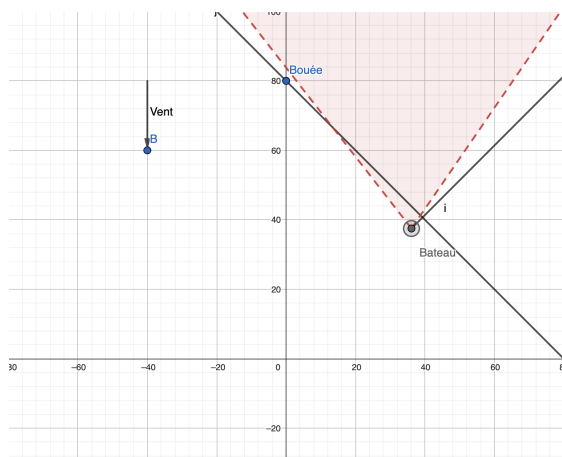
Quelques images pour illustrer :



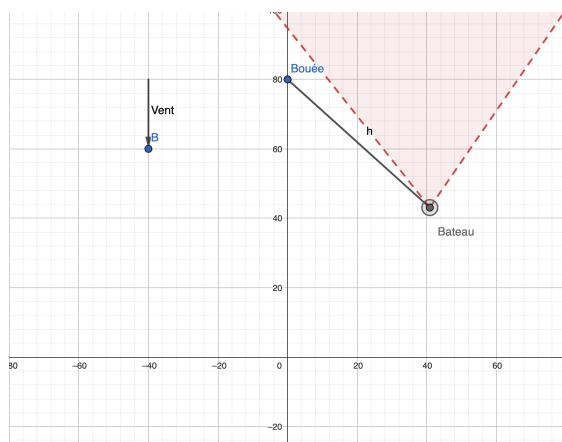
Dans le cas présent, lorsque le vent n'est pas de face, le bateau se dirige directement vers la bouée.



Dans cette situation, le bateau est face au vent.
Par conséquent, il va naviguer à un angle de 45° par rapport à la direction du vent.



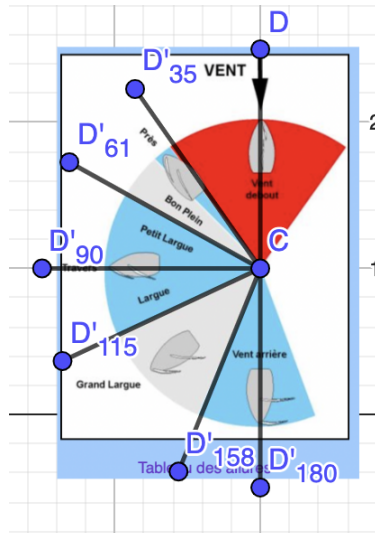
Après avoir navigué sur une certaine distance, la direction vers la bouée ne sera plus face au vent.



Nous pouvons donc naviguer directement vers la bouée.

Commandes du bateau

Pour ce qui est de la commande des voiles, notre approche est de définir simplement un angle de voile en fonction de la direction du vent. Voir ci-dessous :



En ce qui concerne la commande du safran, il est clair que nous devons revoir et améliorer cette partie. Pour l'instant, notre code calcule la différence entre le cap désiré et le cap résultant de l'algorithme de sélection du cap. Plus cette différence est grande, plus le bateau effectue une rotation importante.

Réalisation

Je vous invite à consulter directement le code. Il est soigneusement commenté et subdivisé en fonctions facilement compréhensibles pour une meilleure lisibilité et une compréhension optimale.

Carte de sûreté switch

1. Objectif

L'objectif de la carte switch va être de switcher entre le mode manuel (pilotage du bateau à la télécommande) et le mode automatique (pilotage autonome du bateau) suite à l'activation du switch CH6 de la télécommande.

2. Conception

Nous avons choisi de faire ce système de manière purement analogique dans un souci de sûreté de fonctionnement. En effet, en cas de défaillance du microcontrôleur, on veut pouvoir repasser dans le mode manuel afin de reprendre le contrôle du bateau. Nous avons donc développé une carte physiquement séparée de la carte mère et fonctionnant uniquement avec des composants analogiques .

Cette carte va devoir interagir et traiter 5 signaux :

3 PWM de la télécommande :

- PWM Interrupteur
- PWM Manuel 1 (voile)
- PWM Manuel 2 (safran)

2 PWM du μC

- PWM automatique 1 (voile)
- PWM automatique 2 (safran)

Afin de fournir deux signaux de sortie PWM voile et PWM safran.

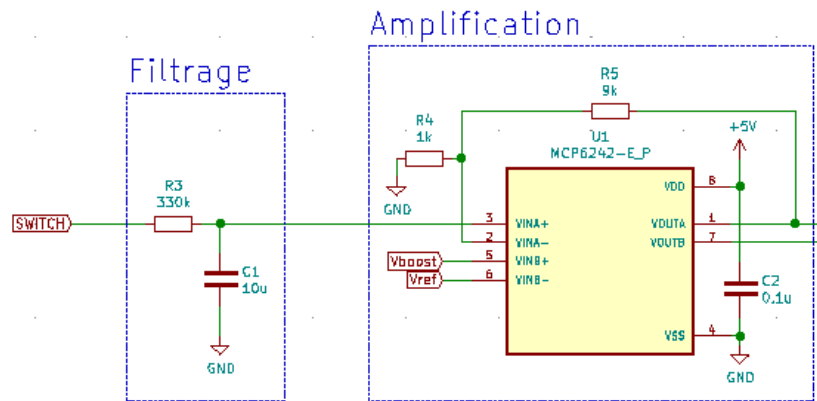
Il va donc s'agir d'en un premier temps de récupérer et traiter le signal switch de CH6 de la télécommande.

Ce signal de sélection entre mode AUTO/MANUEL (CH6) a une amplitude de 3,3V et une période de 20 ms soit 50HZ. Quand CH6 est en haut le signal est à l'état haut vaut 2ms et quand CH6 est à l'état bas il vaut 1ms.

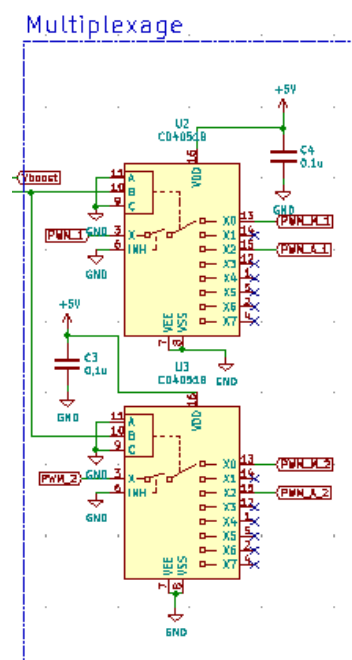
On mesure donc les valeurs moyennes de ce signal à l'état haut et à l'état bas :

- $V_{moy_bas} = 1ms \cdot 3,3v / 20ms = 0,165V$
- $V_{moy_haut} = 2ms \cdot 3,3v / 20ms = 0,33V$

Il serait trop risqué de travailler directement avec ces signaux brut (faible tension, risque de parasites...). On va donc filtrer le signal d'entrée pour en retirer la valeur moyenne puis le multiplier par 10 à l'aide d'un AOP. Ensuite à l'aide d'un comparateur nous allons comparer le signal avec une tension seuil. Cette dernière sera de 2,5V, créé à partir d'un pont diviseur de tension mis sur le 5V. Si la tension de CH6 (multipliée par 10) est inférieure à 2,5V alors V_{out} de l'AOP = 0V si elle est supérieure à 2,5V alors V_{out} = 5V.



Ce signal va ensuite interagir avec deux multiplexeurs permettant d'envoyer au bateau soit le signal manuel soit le signal automatique. Le câblage se fera comme suit :

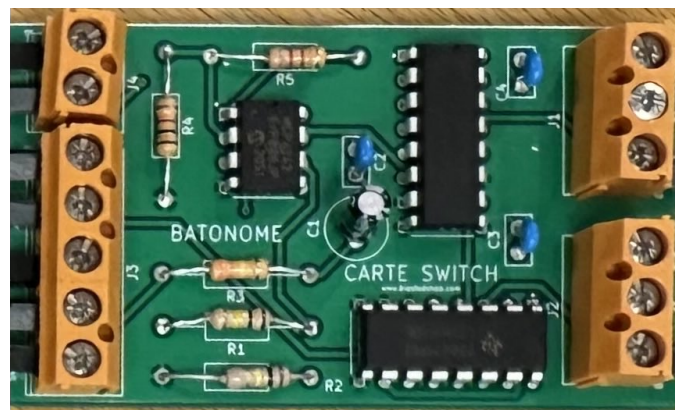


3. Tests et validation

Afin de tester ce système avant la réalisation du PCB on vient assembler les composants sur breadboard et vérifier que l'on arrive bien à commander le bateau en mode manuel ou autonome en fonction de ce qui a été sélectionné sur CH6.

Une fois cela validé on effectue le routage du PCB sur le logiciel kicad et on le lance en production sur JLCPCB.

On réalise à nouveau la batterie de tests. Une fois le PCB validé on peut l'intégrer au bateau.



4. Conclusion et axes d'améliorations

Le PCB final est fonctionnel et permet bien de switcher entre le mode manuel et automatique de manière sécurisée on peut donc valider cette partie

Pour ce qui pourrait être amélioré : actuellement il n'est pas possible d'alimenter de manière optimale les 2 servomoteurs à l'aide des pins 5V et GND de sortie du PCB. En effet, le 5V fournit à la carte switch (par le μ c) ne permet pas d'assurer le besoin en courant des 2 servos. Il serait donc pertinent de brancher le 5V de la batterie sur la carte switch afin qu'elle le redistribue aux 2 servos.

Afin d'optimiser le volume occupé par les cartes dans le bateau, il serait également pertinent de faire une seule et même carte pour la partie switch et carte mère.

Tous les documents nécessaires à la reprise de cette carte (fichiers kicad, gerber, bom) se trouvent dans le git.