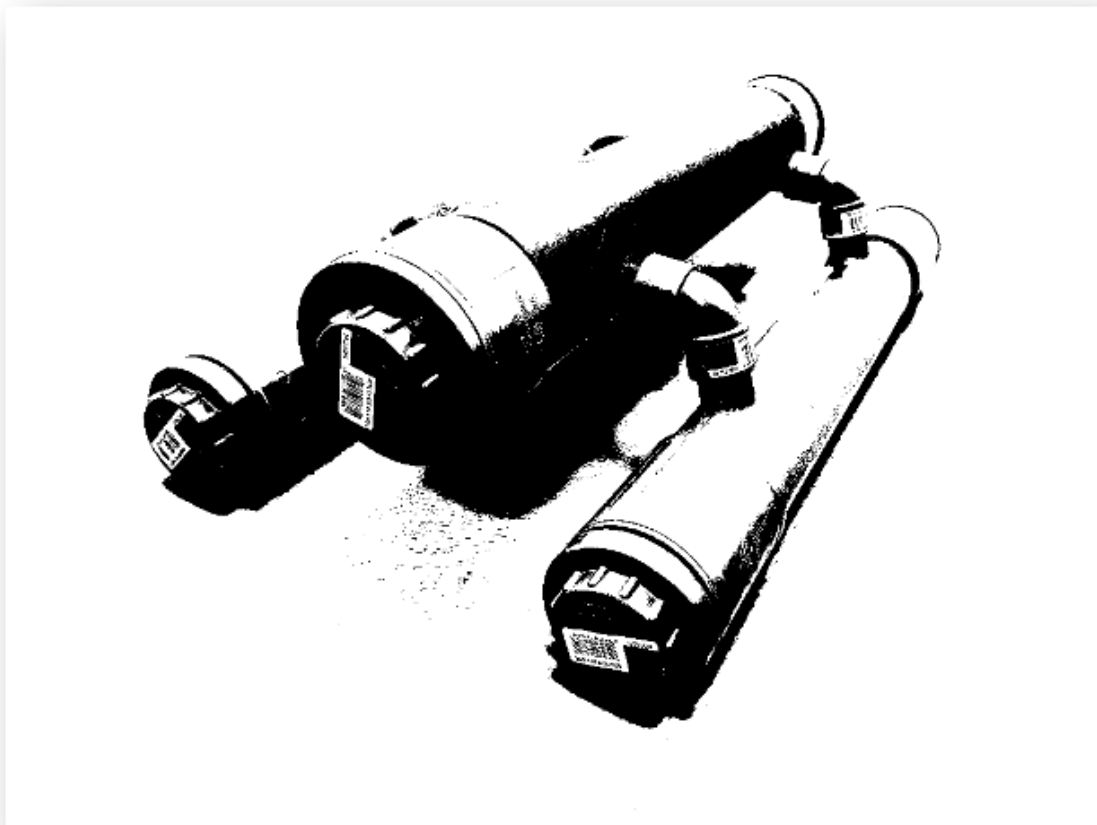


Projet Drone Sous-marin

Revue de projet Final



I.	Présentation technique du projet et de ses objectifs.	4
II.	Organisation matérielle du système	4
1.	Partie mécanique	4
2.	Moteurs	4
3.	Batterie et chargeur	4
4.	Les systèmes de traitement du drone sous-marin et de la télécommande.....	4
5.	Transmetteur HF	4
6.	Caméra	4
III.	Diagramme Sagittal	5
IV.	Description des objets techniques	5
V.	Analyse fonctionnelle niveau 1	6
1.	Schéma fonctionnel du drone sous-marin	6
a.	Partie drone sous-marin.....	6
b.	Description des fonctions et signaux.....	6
2.	Schéma fonctionnel de la télécommande.....	7
a.	Partie télécommande niveau 1	7
b.	Description des fonctions et des signaux.....	7
c.	Partie télécommande niveau 2	8
d.	Description des fonctions et signaux.....	8
VI.	Cahier des charges.....	9
VII.	Répartition des différentes réalisations	11
VIII.	Marc-Antoine Woodcock : Carte-mère et Châssis du drone sous-marin.....	12
1.	Carte-mère	12
2.	Choix réalisé	13
3.	Châssis	15
a.	Solution n°1	15
b.	Solution n°2	16
c.	Choix réalisé	16
IX.	Éric Rebillon : Alimentation et communication	17
1.	Diagramme fonctionnel Niveau 2 Communication	17
2.	Solution Communication	18
3.	Diagramme fonctionnel Niveau 2 Alimentation.....	18
4.	Solution Alimentation	18
5.	Solution pour les tensions inférieures à celle de la batterie	19
6.	Solution pour les tensions supérieures à celle de la batterie	19
7.	Validation du fonctionnement du module RF et procédure de test	19

8.	Fonction Lora.....	20
9.	Validation et test de la transmission et réception	20
a.	Test de communication sans carte de développement	21
b.	Communication depuis la stm32.....	21
10.	Code envoie/réception.....	22
11.	Envoie par RJ45 Filaire.....	22
12.	Validation du fonctionnement des modules d'alimentation	23
13.	Test en tension	23
14.	Test en courant.....	23
15.	Test en tension mesure	23
X.	Damien NOUHAN : Télécommande	25
1.	Choix possible :.....	25
2.	Les caractéristiques de la télécommande :.....	26
3.	Le développement.....	26
XI.	Clément MILLEREUX : Déplacements.....	30
1.	Choix à effectuer :	30
2.	Configuration retenue :	31
3.	Tests	33
XII.	Thomas COUTANT : Mise en place du flux vidéo et de l'alimentation de la télécommande ...	35
1.	L'utilisation d'une Raspberry :.....	35
2.	Choix réalisé et test :	37
XIII.	Conclusion	40

Figure 1 : Diagramme Sagittal du système	5
Figure 2 : Schéma fonctionnel de l'aéroglesseur de niveau 1	6
Figure 3 : Schéma fonctionnel de la télécommande de niveau 1	7
Figure 4 : Schéma fonctionnel de la télécommande de niveau 2	8
Figure 5: Ordonnancement du drone sous-marin.....	14
Figure 6 : Solution n°01	15
Figure 7 : Solution n°01 réel	15
Figure 8 : Solution n°02	16
Figure 9 : Schéma Fonctionnel communication niveau 2	17
Figure 10: Solution Communication Sous-l'eau	18
Figure 11: Schéma Fonctionnel Alimentation niveau 2	18
Figure 12: Solution Alimentation	19
Figure 13 : Solution Alimentation.....	19
Figure 14 intérieur du module Lora	20
Figure 15 Test transmission et réception	20
Figure 16 Terminal Test communication.....	21
Figure 17 transceiver.....	22
Figure 18 : Module Abaisseur.....	23
Figure 19 : Module élévateur	23
Figure 20 : Schéma de test en tension	23
Figure 21 : Schéma test en courant.....	23
Figure 22 : Oscilloscope tension Module abaisseur réglages 5V.	24
Figure 23: Oscilloscope tension Module abaisseur réglages 3.3V.	24
Figure 24: Télécommande Face° 1	25
Figure 25: Télécommande Face 2.....	25
Figure 26 : Dessin de la manette	27
Figure 27 : Pièce arrière Figure 28 : Pièce avant	27
Figure 29: Schématique du PCB télécommande	28
Figure 30 Capture PWM dynamique	34
Figure 31: Bureau Raspbian sur l'écran Raspberry	35
Figure 32: Photo prise par la caméra Raspberry	36
Figure 33: Retour vidéo caméra Raspberry.....	36
Figure 34: Alimentation Raspberry	37
Figure 35 : Splash screen de démarrage	38
Figure 36 : Retour vidéo	38

I. Présentation technique du projet et de ses objectifs.

Le but de projet est de développer un drone sous-marin « from scratch ». Ce dernier doit être simple d'utilisation et de mise au point afin d'être accessible à toute personne voulant le piloter.

Ce projet comprend la conception d'un drone sous-marin ainsi qu'une télécommande ce qui nous permettra d'augmenter nos compétences en conception assistée par ordinateur (CAO) ainsi qu'en programmation, mais également le travail en équipe.

II. Organisation matérielle du système

Le drone sous-marin est un véhicule capable de se mouvoir dans toutes les directions dans l'eau et de garantir une stabilité exemplaire. Il est dirigé par une télécommande sur laquelle les actionneurs seront à définir par l'équipe dans le respect des exigences initiales : simplicité / efficacité.

1. Partie mécanique

Les parties mécaniques du drone sous-marin sont à convenir. Elles pourront être conçues par le biais d'une imprimante 3D, un kit ou bien DIY (Do It Yourself).

Cette partie assure la sécurisation de toutes les cartes électroniques et équipements nécessaires au déplacement du drone sous-marin.

2. Moteurs

Le drone sous-marin doit être capable de se mouvoir dans l'eau dans toutes les directions possibles. Le nombre de moteurs et le type de moteur restent encore à définir.

3. Batterie et chargeur

Le drone sous-marin ainsi que la télécommande doivent être équipés de batteries rechargeables.

L'autonomie du drone sous-marin doit être de 30 minutes.

L'autonomie de la télécommande doit être de 2h.

4. Les systèmes de traitement du drone sous-marin et de la télécommande



Ces systèmes de commande sont architecturés autour de deux familles de microcontrôleur au choix :

- Microcontrôleur CISC de la famille Freescale (MC9S08QE ?? avec MC9S08JS16)
- Architecture ARM de la famille MikroElektronika (STM32 avec carte d'extension customisée)

5. Transmetteur HF

La communication sans fil est réalisée par un module émetteur/récepteur à 2.4 GHz.

La transmission doit se faire entre la télécommande et le drone sous-marin en évitant les contraintes du milieu : l'eau.

6. Caméra

Une caméra sera placée sur le drone sous-marin pour permettre une visualisation du fond marin.

III. Diagramme Sagittal

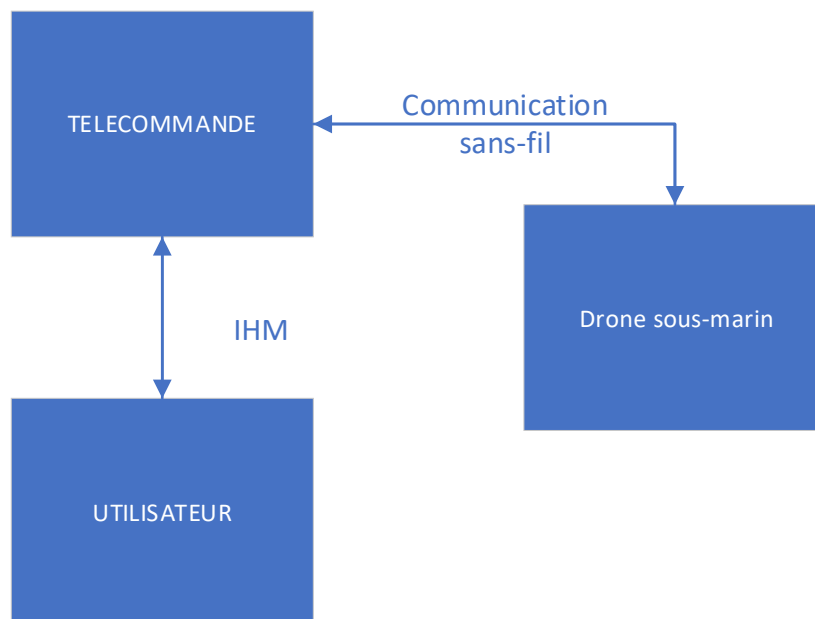


Figure 1 : Diagramme Sagittal du système

C'est un projet constitué d'un drone sous-marin et d'une télécommande dont le but est de communiquer entre eux.

IV. Description des objets techniques

Le drone sous-marin :

- Une unité de traitement des différents modules présents sur le drone sous-marin associée à un module sans fil bidirectionnelle capable de communiquer avec la télécommande.
- Un module de communication sans-fil bidirectionnel communiquant avec la télécommande.
- Un système de gestion des moteurs.
- Un système de positionnement spatial.
- Un superviseur de batterie.
- Un système de récupération en cas de problèmes.
- Une caméra.

La télécommande :

- Une unité de traitement des différents modules présents dans la télécommande associée à un module sans fil bidirectionnel capable de communiquer avec le drone.
- Un module de communication sans fil communiquant avec le drone sous-marin.
- Une interface Homme-Machine (IHM) composée d'une partie assurant le contrôle complet du drone sous-marin.
- Retour vidéo avec écran.

V. Analyse fonctionnelle niveau 1

1. Schéma fonctionnel du drone sous-marin

a. Partie drone sous-marin

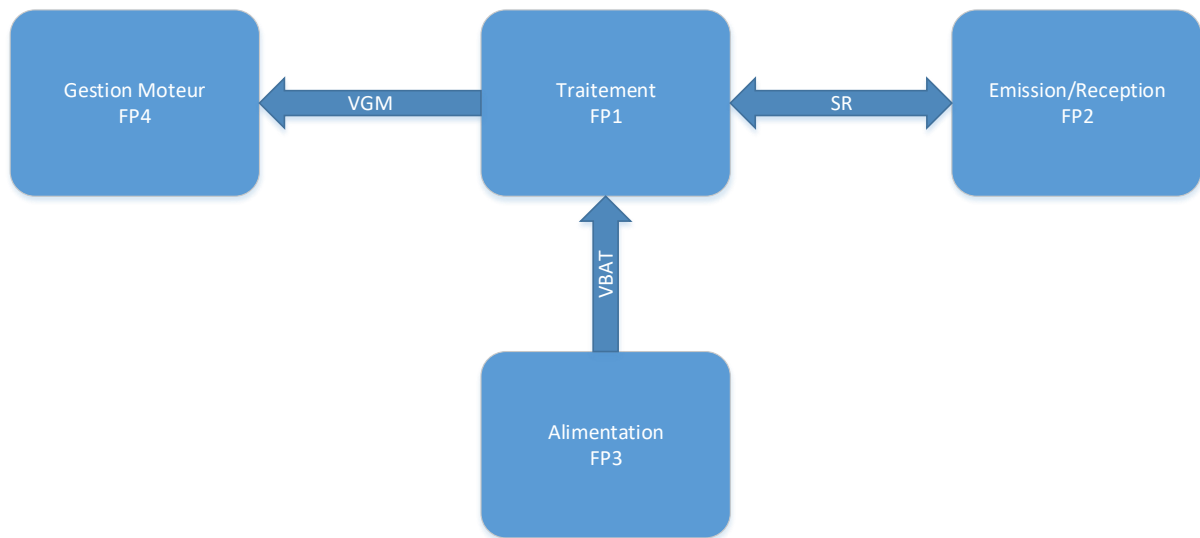


Figure 2 : Schéma fonctionnel de l'aéroglesseur de niveau 1

b. Description des fonctions et signaux

- FP1 : Le traitement des informations est réalisé par une carte microcontrôleur qui permet la gestion des différentes entrées/sorties des cartes présentes sur le drone sous-marin.
- FP2 : La carte de transmission est utilisée comme canal de transmission hertzien entre le drone sous-marin et la télécommande dans le but de transférer toutes les informations de commandes nécessaires au fonctionnement du système.
- FP3 : L'alimentation est réalisée autour d'une batterie qui doit fournir tous les courants et tensions nécessaires pour le fonctionnement du drone sous-marin.
- FP4 : Les cartes de contrôle des moteurs permettent la mise en forme du signal PWM afin de commander les déplacements du drone sous-marin.
- SR : Signaux numériques série asynchrone contenant les informations à échanger avec la télécommande.
- VBAT : Signal analogique image de la charge restante de la batterie.
- VGM : Signaux MLI (PWM), utiles pour la gestion de la vitesse de rotation des hélices du drone sous-marin.

2. Schéma fonctionnel de la télécommande

a. Partie télécommande niveau 1

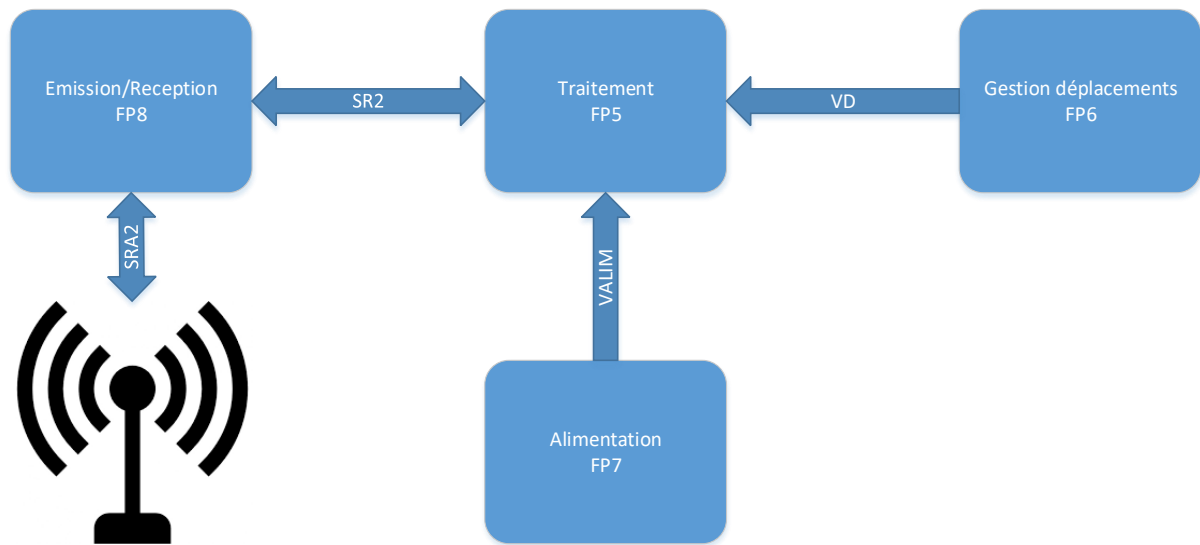


Figure 3 : Schéma fonctionnel de la télécommande de niveau 1

b. Description des fonctions et des signaux

- FP5 : Le traitement des informations est réalisé par une carte microcontrôleur qui permet la gestion des différentes entrées/sorties des cartes présentes sur la télécommande.
- FP6 : La gestion des déplacements se fait à l'aide d'une interface homme-machine.
- FP7 : L'alimentation est réalisée autour d'une batterie qui doit fournir tous les courants et tensions nécessaires pour le fonctionnement de la télécommande.
- FP8 : La carte de transmission est utilisée comme canal de transmission hertzien entre le drone sous-marin et la télécommande de façon à transférer toutes les informations de commandes nécessaires au fonctionnement du système.
- VD : Tensions analogiques (0 à 3V) provenant des joysticks et signaux tout ou rien provenant des boutons poussoirs sur les joysticks.
- VALIM : Signal analogique image de la charge restante de la batterie.
- SR2 : Signaux numériques série asynchrone contenant les informations à échanger avec le drone sous-marin.
- SRA2 : Signaux analogiques full duplex hertzien avec une porteuse à 2,4 GHz

c. Partie télécommande niveau 2

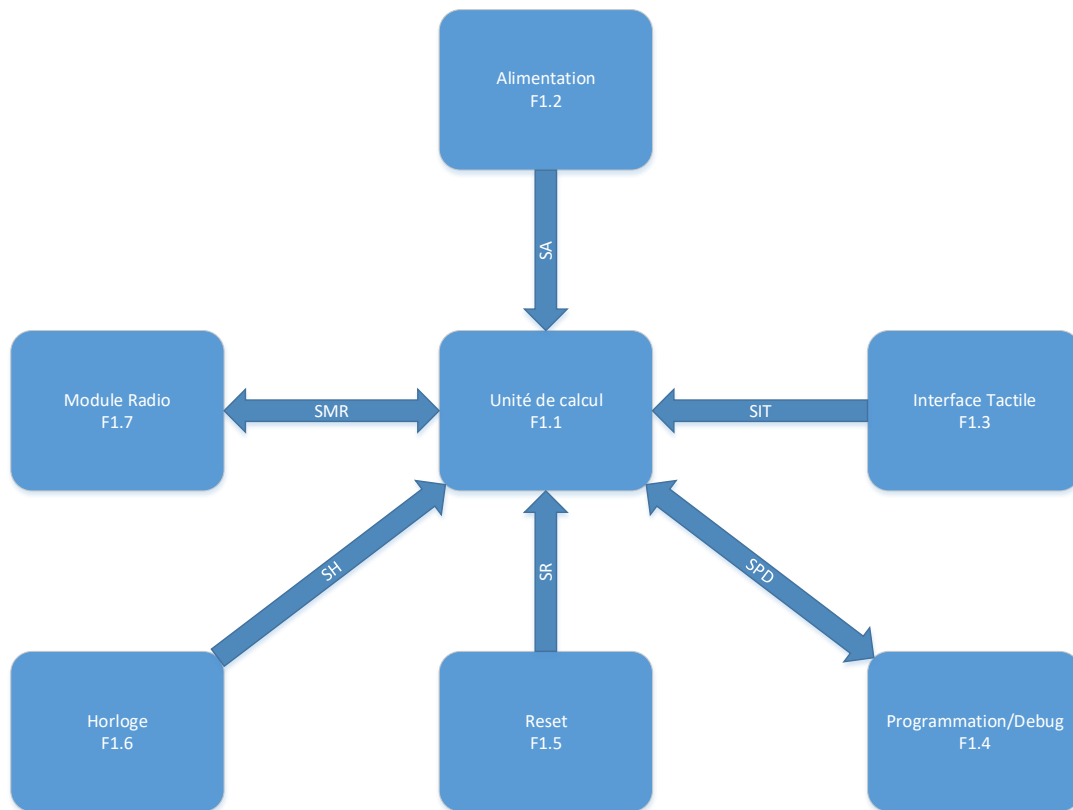


Figure 4 : Schéma fonctionnel de la télécommande de niveau 2

d. Description des fonctions et signaux

- F1.1 : Unité de calcul comprenant le microcontrôleur qui permet la réalisation des fonctionnalités.
- F1.2 : Alimentation de la carte microcontrôleur ainsi que le module radio.
- F1.3 : Interface Tactile, correspond à la manipulation des joysticks.
- F1.4 : Programmation/Debug, qui permet de programmer le microcontrôleur
- F1.5 : Reset, permet de réinitialiser le microcontrôleur à l'état originel.
- F1.6 : Horloge, le montage horloge / oscillateur génère des signaux carrés. Ce dernier s'appuie sur le fonctionnement du quartz.
- F1.7 : Module Radio, permettant la communication entre le drone sous-marin et la télécommande.
- Sit : Signal unidirectionnel numérique généré par la manipulation des joysticks.
- Spd : Signal bidirectionnel numérique permettant de communiquer avec le microcontrôleur
- Sr : Signal unidirectionnel numérique tout ou rien permettant l'action RESET.
- Sh : Signal unidirectionnel numérique générant le signal d'horloge.
- Smr : Signal bidirectionnel hertzien avec une porteuse de 2.4Ghz.
- Sa : Signal unidirectionnel fournissant l'alimentation de la carte microcontrôleur et du module radio.

VI. Cahier des charges

Cahier des charges du drone sous-marin

<u>Caractéristiques</u>	<u>Effets</u>	<u>Exemples</u>
Taille	Prise en main, ergonomie,	Plastique PVC Dimension max 50 cm x 50 cm
Poids	Prise en main	< 10 kg
Forme	Subjectif	Carré, rectangulaire, cylindrique, ovoïdale
Radio	Portée de 20 mètres	Module radio
Batterie	2h	Pile, Batterie rechargeable
Mouvements	Se mouvoir dans l'eau	Avance, recule, gauche, droite, montée, descente
Profondeur/pression	Fiabilité du produit	10 mètres
Température de fonctionnement	Fiabilité du produit	Entre 10 °C et 40 °C
Milieu		Eau douce et sans courant
Vitesse	Contrôle du sous-marin	Max : 4 nœuds
Eclairage	Aide pour la caméra	2400 lumens
Retour vidéo	Visualisation des fonds marins	Caméra
Possibilité de transport	Transport pour salon, voiture	Poignée, sangle

Cahier des charges de la télécommande

<u>Caractéristiques</u>	<u>Effets</u>	<u>Exemples</u>
Taille	Prise en main, ergonomie,	Plastique, bois, caoutchouc, silicone, aluminium
Poids	Prise en main	< 600 g
Forme	Subjectif	Toute forme possible, Carré, rectangulaire, ovale
Actionneur	Piloter le sous-marin	Joystick numérique ou analogique, Bouton ou gâchettes (montée et descente)
Radio	Portée de 100 mètres	Module radio
Batterie	2h	Pile, Batterie rechargeable, barre graphe, switch, led de fonctionnement, led de charge
Ouverture	Réparation possible	Réparation par le développeur
Flottaison	En cas de chute dans l'eau	
Reset	Reset de la télécommande	Permet de rebooter la télécommande
Support écran	Visualisation de la caméra	Ecran
Possibilité de transport	Transport pour salon, voiture	Poignée, sangle

VII. Répartition des différentes réalisations

Réalisation	Nom et Prénom
Conception de la carte microcontrôleur, de l'ordonnanceur du système sur le drone sous-marin et choix du châssis.	Marc-Antoine Woodcock 100€
Etude de toute la partie moteur, ballastes et mouvements = déplacement.	Clément Millereux 500€
Flux vidéo, Eclairage. Etude de la réalisation de l'alimentation de la télécommande rechargeable.	Thomas Coutant 300€
Etude de la réalisation de l'alimentation pour le drone sous-marin. Sonar Mise en place d'un module radio et d'un protocole de communication pour l'ensemble du système.	Éric Rebillon 200€
Conception de la carte microcontrôleur et de l'ordonnanceur du système sur la télécommande. Châssis télécommande 3D Mise au point de commandes manuelles pour contrôler du drone sous-marin.	Damien Nouhant 100€

Estimation du projet au vue des premières recherches de solutions :

- 1200 euros

VIII. Marc-Antoine Woodcock : Carte-mère et Châssis du drone sous-marin

Dans ce projet, je suis en charge de la partie carte-mère du drone sous-marin ainsi que son châssis.

La carte-mère devra respecter les fonctionnalités définies dans le cahier des charges.

La contrainte majeure que l'on va observer lors de la fabrication du châssis du drone sous-marin est : L'ÉTANCHEITE ainsi que l'intégration matérielle du système embarqué.

La zone de test est une notion également à étudier.

1. Carte-mère

Plusieurs solutions s'offrent à moi pour le choix de la carte mère du drone sous-marin. Cependant certaines contraintes ont été établies dans le cahier des charges :

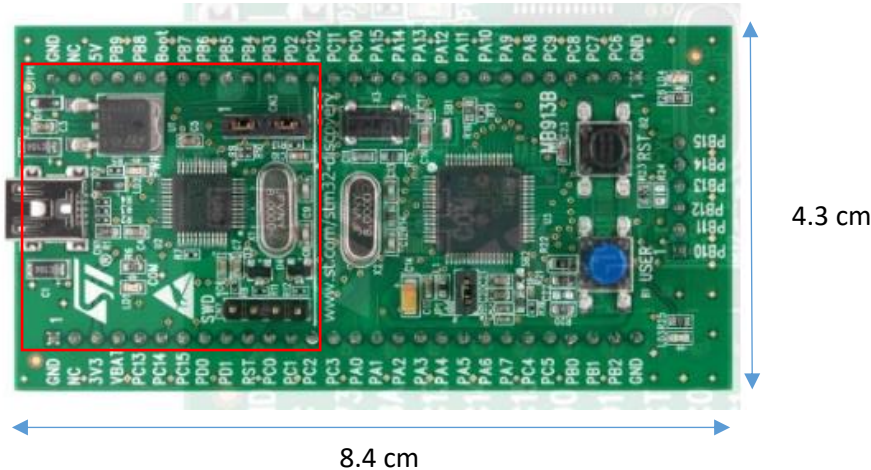
- UART pour la communication entre le drone sous-marin et la télécommande
- TIMER pour la commande des moteurs et servomoteurs en PWM
- Alimentation en 3.3V ou 5V
- Débogage facile depuis un pc
- Taille longueur < 15 cm et largeur < 5 cm

Afin de respecter ces contraintes, mon choix s'est porté sur différentes carte ARM :

- Nucleo-xxxxx
- STM32L152C-DISCO
- STM32VLDISCOVERY
- STM32F0308-DISCO
- STM32F072B-DISCO
- STM32F0DISCOVERY
- STM32F3348-DISCO

J'ai fait le choix d'une carte ARM car cette dernière permet de réaliser les fonctionnalités définies dans le cahier des charges. De plus, elle permet d'avoir un résultat de fonctionnement plus rapide qu'une carte électronique réalisée sur Altium Designer sans avoir une garantie de fonctionnement.

2. Choix réalisé



Les caractéristiques de la STM32VLDISCOVERY :

- STM32F100RBT6B microcontroller, 128 KB Flash, 8 KB RAM in 64-pin LQFP
- On-board ST-Link
- Deux LEDs rouges; LD1 pour l'USB communication, LD2 for 3.3 volts power on
- Alimentation 5 V ou 3.3 V
- 2 LEDs USER, LD3 and LD4 (verte et bleu)
- Deux boutons (User et Reset)

Mon choix s'est porté sur la STM32VLDISCOVERY, cette dernière répond aux fonctionnalités voulues et aux dimensions requises d'après le cahier des charges.

La STM32VLDISCOVERY a une taille de 8.4 cm x 4.3 cm soit inférieure aux dimensions énoncées précédemment.

La STM32VLDISCOVERY possède toutes les fonctionnalités nécessaires pour le drone sous-marin.

La carte possède :

- USART pour la communication entre le drone sous-marin et la télécommande à l'aide des modules RF.
- 7 Timers PWM 16 bits avec 24 channels qui permettra de contrôler les ESC/moteur pour la propulsion ainsi que les servomoteurs pour l'inclinaison pods.
- Alimentation se fera à partir du drone sous-marin à l'aide de batteries.

Par la suite j'ai réalisé plusieurs programmes de tests :

- Lecture d'un bouton
- Communication USART
- PWM
- Convertisseur analogique-numérique

Ordonnancement des fonctionnalités du drone Sous-marin :

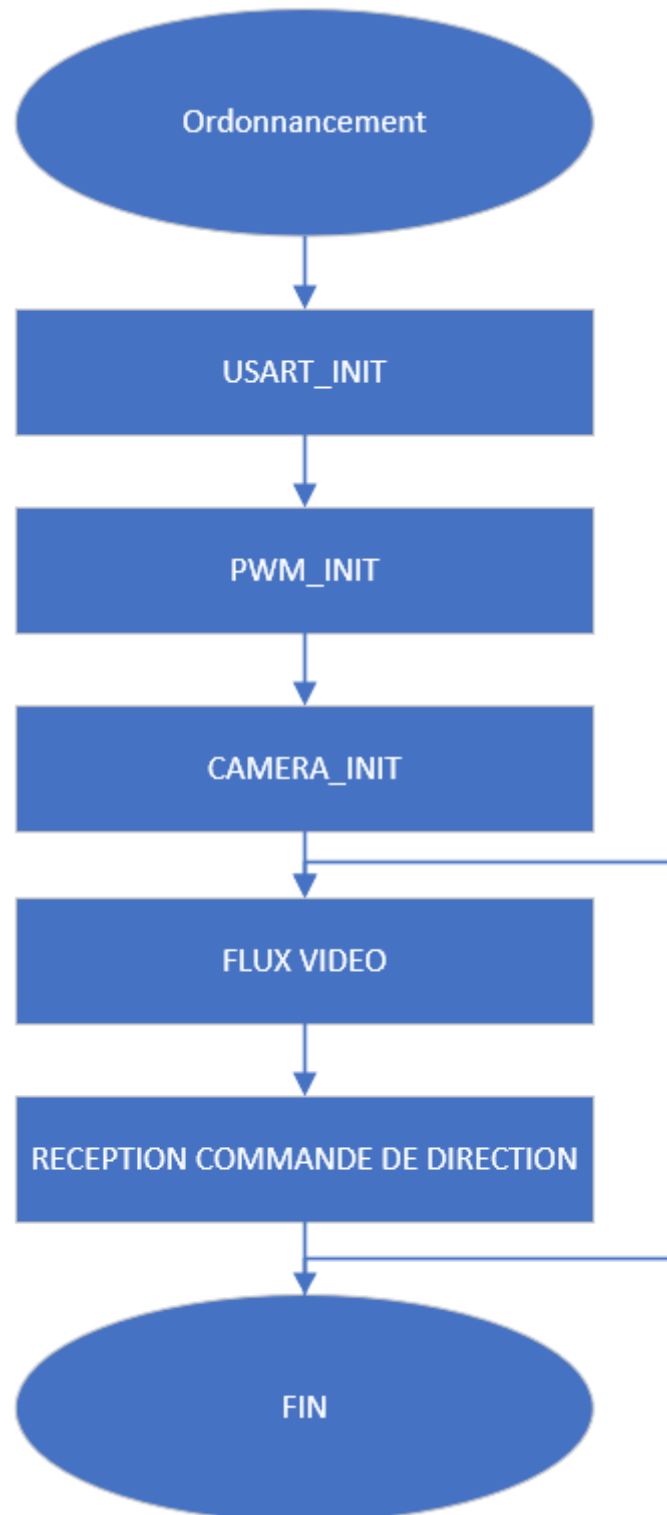


Figure 5: Ordonnancement du drone sous-marin

3. Châssis

Pour l'élaboration du châssis du drone sous-marin, nous avons 2 solutions possibles :

a. Solution n°1

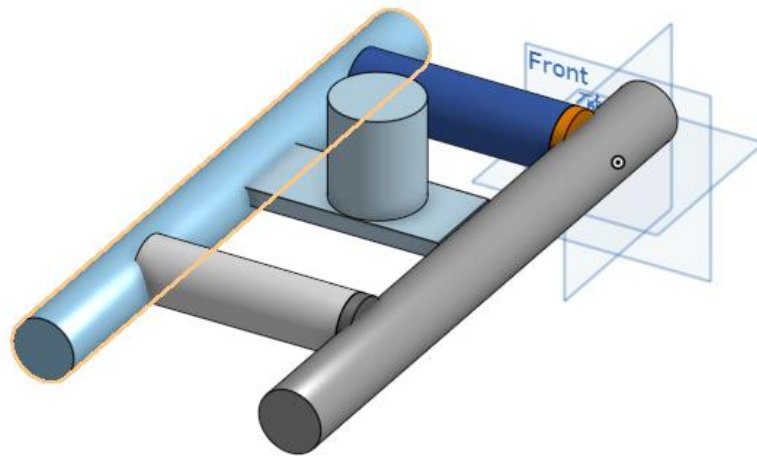


Figure 6 : Solution n°01

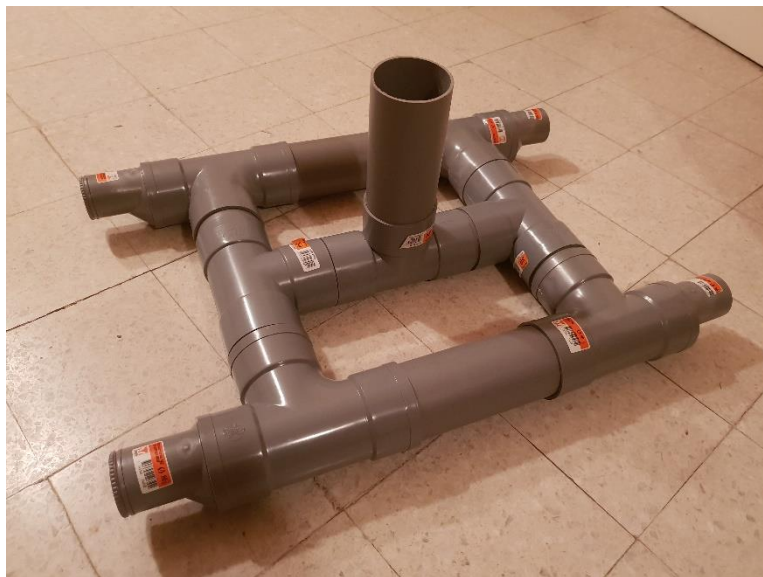


Figure 7 : Solution n°01 réel

Ceci est le premier prototype de drone sous-marin possible, il respecte les contraintes de poids et taille cependant au vu de l'intégration de la partie hardware tels que la caméra et l'éclairage, la structure ne semble pas optimale pour être un drone sous-marin donc on a réfléchi à une deuxième structure.

b. Solution n°2



Figure 8 : Solution n°02

Voici une deuxième possibilité de châssis pour un drone sous-marin. Cette structure permet l'intégration plus optimale pour la caméra, l'éclairage ainsi que les batteries.

c. Choix réalisé

J'ai réalisé mon choix concernant la structure du drone sous-marin : la solution n°2 est la plus appropriée. En effet, elle permet de mettre en place les moteurs/pods ainsi que les ballasts.

La caméra sera située dans le cylindre principal et deux spots LEDs dans les tubes latéraux. La structure mesure 50 cm x 50 cm et donc respecte bien le cahier de charge.

La structure fait un poids inférieur au 10 kg requis.

La 1^{ère} structure n'étant pas choisie pour le châssis du drone, cette structure sera utilisée en tant que bouée d'émission du drone sous-marin.

IX. Éric Rebillon : Alimentation et communication

1. Diagramme fonctionnel Niveau 2 Communication

Ce diagramme nous explique le fonctionnement de la communication mais également sa procédure de réalisation.

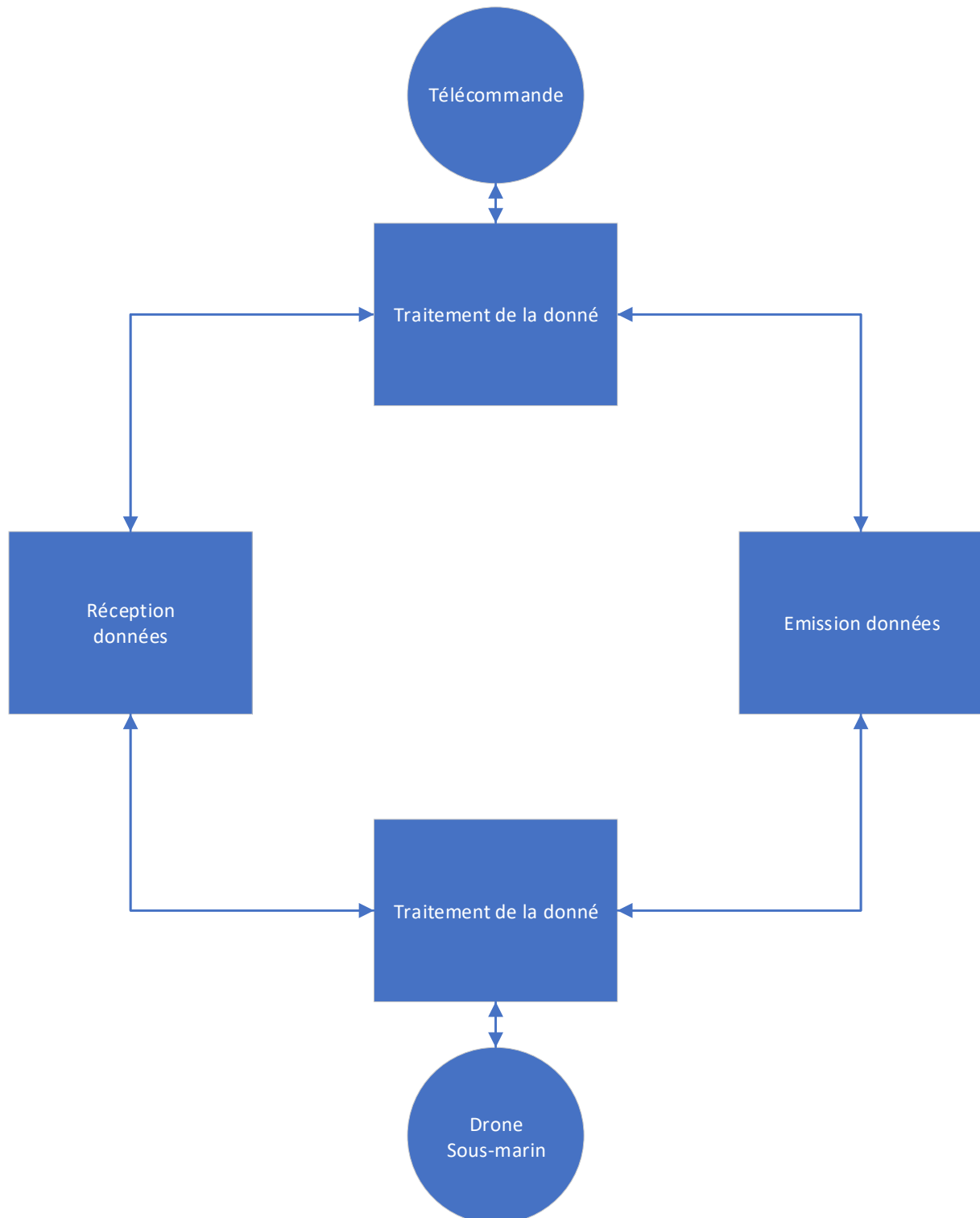


Figure 9 : Schéma Fonctionnel communication niveau 2

2. Solution Communication

Solution choisie pour répondre aux problèmes de communication dans l'eau. Donc pour solutionner ce problème nous avons décidé de prendre une bouée qui fasse intermédiaire entre le drone et la télécommande. La bouée sera reliée en filaire aux drones et la bouée communiquera en RF avec la télécommande.

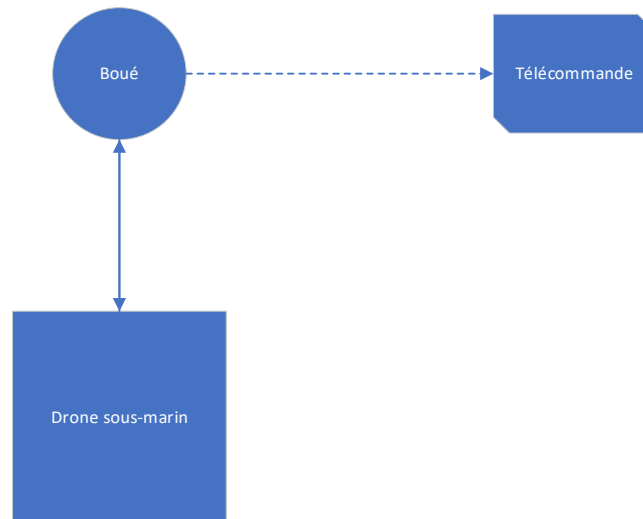


Figure 10: Solution Communication Sous-l'eau

3. Diagramme fonctionnel Niveau 2 Alimentation

Ce diagramme résume la partie alimentation. Le choix de la batterie s'est porté sur une LIPO, qui enverra une certaine tension non-stable et un courant au régulateur de tension qui lui, renverra une tension stable (soit tension augmentée ou diminuée). Ces alimentations seront réparties sur l'ensemble des cartes présentes sur le drone sous-marin.

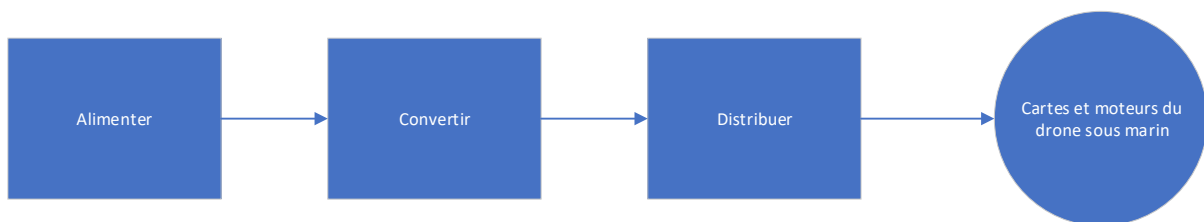


Figure 11: Schéma Fonctionnel Alimentation niveau 2

4. Solution Alimentation

La consommation de chaque partie nous donne les deux caractéristiques pour le bon choix de la batterie LIPO.

Partie à alimenter	Consommation en courant (A)	Tension (V)
Déplacement	40 A + 0.26 A	12V
Carte-mère	0.001A	5V/3.3V
Communication	Non Déterminé	Non Déterminé
Caméra	Non Déterminé	Non Déterminé
Batterie	82Ah	12V

Sur le tableau ci-dessus, nous pouvons remarquer une consommation en courant extrême sur les moteurs donc on adaptera l'alimentation moteur pour éviter les risques sur les autres cartes.

Pour la batterie, il faudrait une capacité de 82Ah pour tenir les deux heures indiquées par le cahier des charges.

5. Solution pour les tensions inférieures à celle de la batterie

Pour la solution d'abaissement de tension, on aura une réception de tension abaissement et à chaque carte. Comme nous sommes sur du courant faible, nous pouvons rassembler plusieurs alimentations de carte.

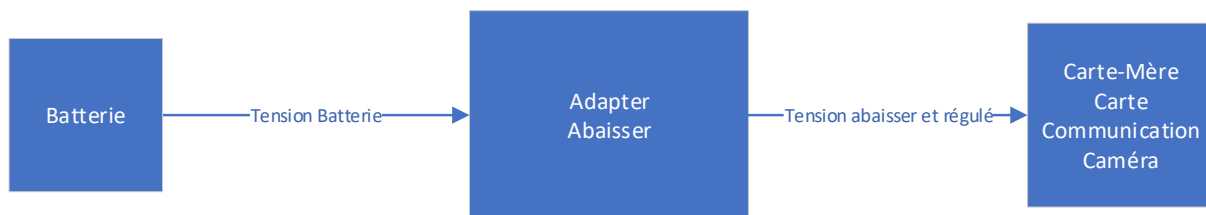


Figure 12: Solution Alimentation

6. Solution pour les tensions supérieures à celle de la batterie

Permet l'éclaircissement de ce qu'il faut faire schématiquement. Cette solution permettra la stabilité en tension pour les moteurs avec une grande résistance en courant.



Figure 13 : Solution Alimentation

7. Validation du fonctionnement du module RF et procédure de test

Module choisi pour la communication RF :
Permet de réaliser une communication
En protocoles UART sur une fréquence de
433 et 868MHz.



8. Fonction Lora

Les fonctions Lora définie dans les drivers effectuent directement les transmission réception par l'intermédiaire des commandes processeur (cf. Rapport individuels).

Pour valider le fonctionnement de l'ensemble des drivers Lora il faut effectuer différentes adaptation (cf. Rapport individuel).

Le Lora n'a pas besoin de mettre en œuvre une trame avec un protocole bien définie car les drivers réaliseront cela et simplifie l'utilisation si les drivers était complet et parfaitement adaptés ais une fois les drivers modifies l'utilisation est plus facile.

Ce qui permet d'accéder directement aux matérielles dans le module le Lora et accéder au différent état et activités en cours de traitement et permettent ainsi d'optimiser le code et donc le fonctionnement.

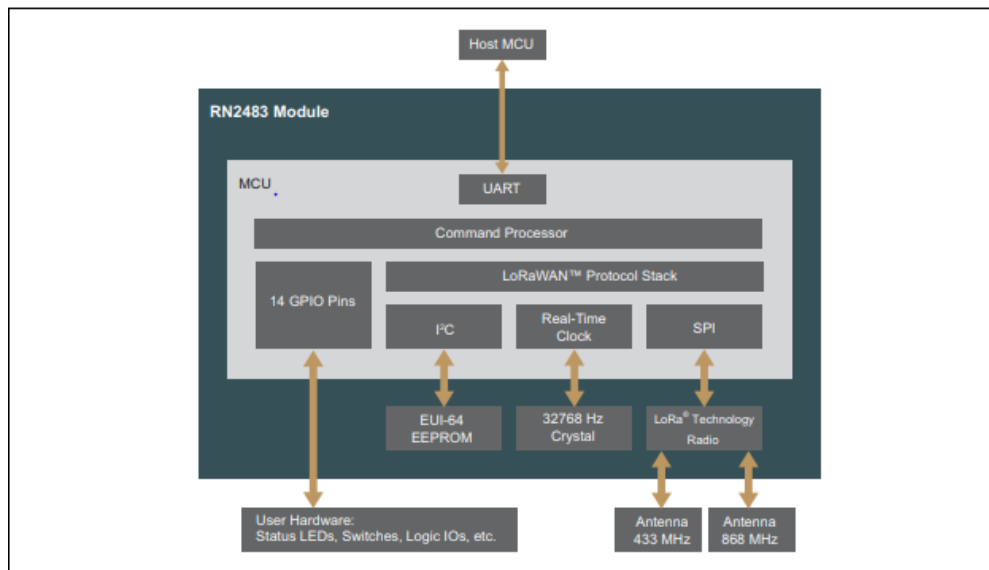


Figure 14 intérieur du module Lora

9. Validation et test de la transmission et réception

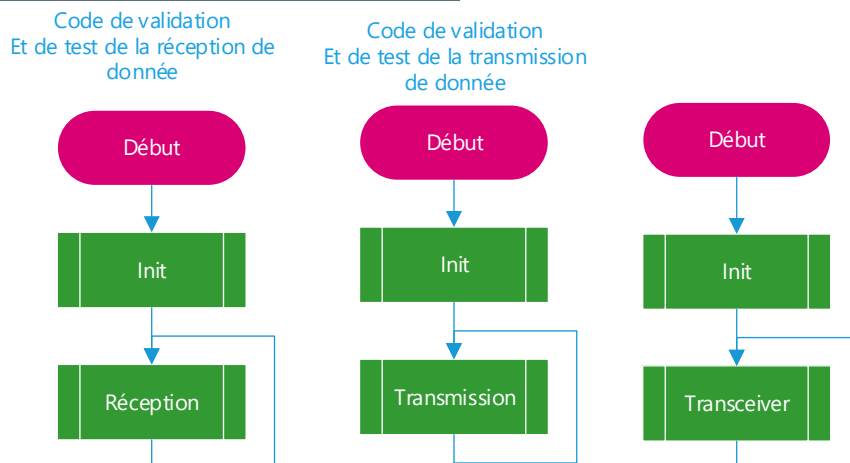


Figure 15 Test transmission et réception

Dans un premier temps je vais réaliser une communication en envoyant les commande de contrôle du module depuis le terminal et puis en second temps j'envverrai les commandes depuis la stm32.

a. Test de communication sans carte de développement

Pour les validations de l'envoi et de la réception, je réalise tout d'abord, grâce au commande radio et mac, une connexion entre deux modules. Tout d'abord, je vais les envoyer directement aux modules sans passer par une carte de développement. Pour valider un fonctionnement et vérifier la réaction du module. Connexion directe aux modules grâce à la commande Linux suivante : « sudo python miniterm.py /dev/ttyUSB1 57600 » cette commande me connecte aux modules puis j'envoie les commandes : « mac pause » pour initialiser le module puis « radio tx <data> » pour réaliser une transmission et « radio rx 0 » pour se mettre en mode de réception et attendre une donnée.

Ce qui nous donne :

<pre>radio_tx_ok radio_tx AA ok radio_tx_ok radio_tx aa ok radio_tx_ok radio_tx 44 ok radio_tx_ok radio_tx 44 ok radio_tx_ok</pre>	<pre>radio rx 0 ok radio_rx AA radio rx 0 ok radio_rx AA radio rx 0 ok radio_rx 44</pre>
Module n°1 envoi une donnée	Module n°2 réception d'une donnée

Figure 16 Terminal Test communication

Ces deux relevés nous montrent un envoi d'une donnée 44 puis l'autre module la reçoit.

b. Communication depuis la stm32

La validation de la transmission me permet de confirmer un envoi de donnée. Cette donnée devra être envoyée par le module RF Lora. Pour le processus transmission, on utilisera une librairie arm incluse dans toutes les cartes arm car elles sont dans les librairies standard, cette librairie est « HAL » permettant de faire appel à des fonctions optimisées et de réaliser un code portable sur plusieurs cartes. L'appel de ces fonctions par l'intermédiaire des librairies permet de gérer tout le buffer et de gérer l'envoi octets par octets. Pour envoyer une donnée, il faut utiliser une commande « radio tx <data> » que j'envoie en chaîne de caractère au module et qui ensuite enverra le module. On répète l'opération avec la réception sauf que l'on envoie « radio rx 0 » au module puis attend la réception de donnée.

10. Code envoie/réception

Pour l'envoi et réception, cela se passe en 4 étapes :

- Réinitialisation du module effectué juste au démarrage,
- Envoie de la commande permettant de s'assurer que l'envoi peut-être réaliser.
- Puis envoie de la donnée
- Et commande à effectuer après chaque commande radio pour valider si tous ces biens passer et vérifier l'état de la pile, pour voir si nous pouvons envoyer plusieurs commande radio si nous ne faisons pas cela, on ne pourra pas faire plusieurs commande radio.

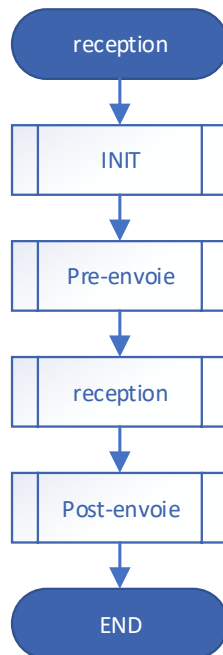
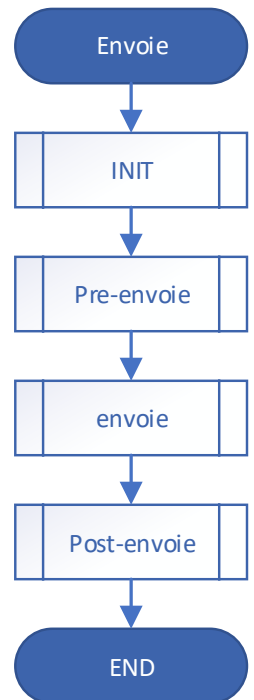


Figure 17 transceiver

Ces algorithmes permettent l'envoi et réception de commande. Cela permet de visualiser comment se passe un envoie et réception de donnée.

Pour voir la description de chaque sous-processus, il faut aller voir le rapport sur la communication.



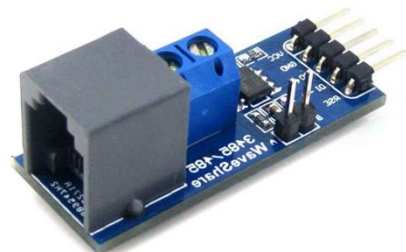
11. Envoie par RJ45 Filaire

Pour la communication sous l'eau nous passerons en filaire en passants les données et commandes par l'intermédiaire d'un câble RJ45

Le module utilisé pour réaliser cela permet de passer de l'UART sur le câble RJ45.

D'après la norme du câble, on aurait une portée de fonctionnement supérieur à celle que l'on sait imposée, c'est-à-dire 10m.

Module utilisé, MAX485 :



12. Validation du fonctionnement des modules d'alimentation

Module choisi pour réaliser l'alimentation de l'ensemble du drone sous-marin

Modules abaisseurs :

Modules éleveurs :

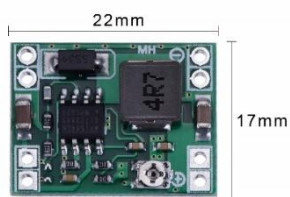


Figure 18 : Module Abaisseur

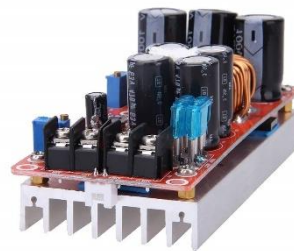


Figure 19 : Module éleveur

13. Test en tension

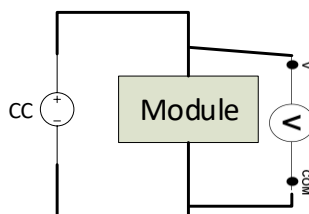


Figure 20 : Schéma de test en tension

Pour valider le fonctionnement de ces deux modules différents, j'ai utilisé une alimentation stabilisée. On envoie une tension d'entrée de 12V et si le module fonctionne (avec les bons réglages) il renvoie une tension de 5V pour le premier module et 12V pour le deuxième. Je ferai varier la tension d'entrée pour vérifier que les tensions de sortie restent stables.

14. Test en courant

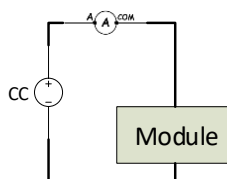


Figure 21 : Schéma test en courant

Pour vérifier sa résistance en tension, je testerai avec l'alimentation stabilisée si je peux tirer du courant et valider le courant limite indiqué par le constructeur.

15. Test en tension mesure

Module régulant pour la STM, caméra autrement dit le module 3A abaisseur.

Après réglages, ayant besoin d'alimenter en 5V et 3.3V il faudra réaliser deux réglages sur deux modules différents. Pour le réglage il suffira de modifier la valeur du potentiomètre en surveillant la tension de sortie jusqu'à tomber sur la tension recherchée.

Réglages pour une tension de 5V, 3A :

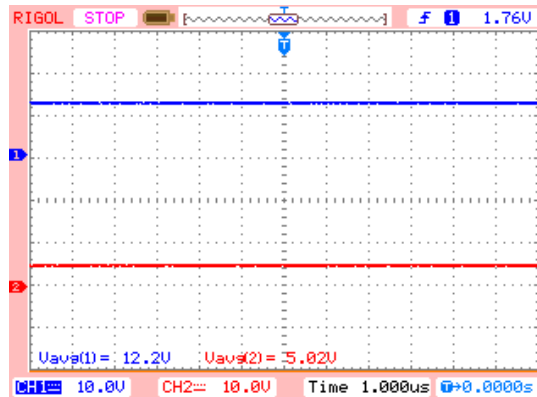


Figure 22 : Oscilloscope tension Module abaisseur réglages 5V.

Réglages pour une tension de 3.3V, 3A :

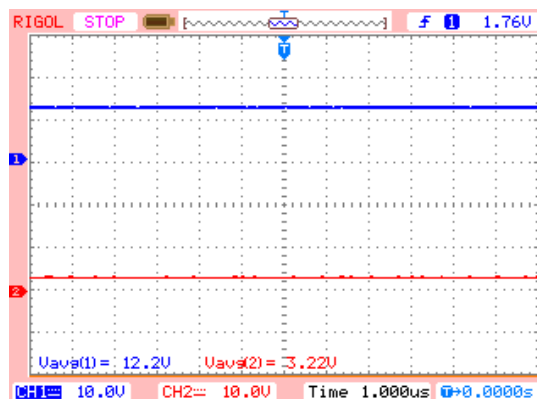


Figure 23: Oscilloscope tension Module abaisseur réglages 3.3V.

Validation de tous les fonctionnements des cartes alimentations. Test effectué avec les moteurs pour l'alimentation élévateur. Alimentation branchée directement sur batterie ainsi je peux valider le fonctionnement du module et le moteur valide un fonctionnement et une plage de fonctionnement.

X. Damien NOUHAN : Télécommande

Je m'occupe de la partie télécommande du projet. L'organisation du boîtier se séparerait entre les boutons de commande, l'écran, avec nos deux systèmes de contrôle.

Schéma de la télécommande :

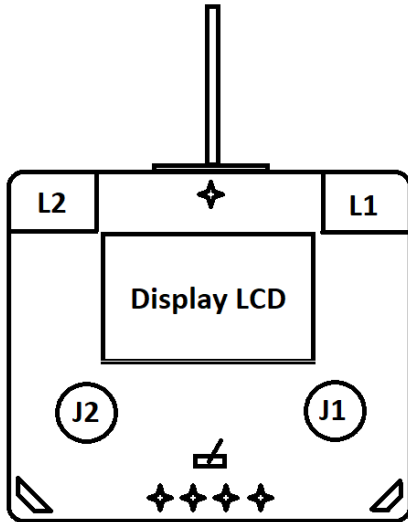


Figure 25: Télécommande Face 2

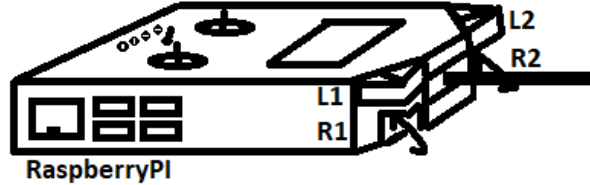


Figure 24: Télécommande Face° 1

- L2** : Mode plongée **R2** : Variation de la vitesse du moteur J2
- L1** : Mode surface **R1** : Variation de la vitesse du moteur J1
- J1** : Contrôle de la direction du moteur droit
- J2** : Contrôle de la direction du moteur gauche

Le choix de l'organisation de notre télécommande s'inspire à la fois de celle d'aéromodélisme mais aussi de la célèbre console de jeu « Playstation 3 ». Comme son nom l'indique, elle permettra de commander indépendamment le drone pour chacun de ces moteurs comme un Tank, et nous afficherons sur l'écran LCD le retour de la caméra embarquée.

La télécommande est conçue pour gérer indépendamment les ordres de commandes et l'affichage de la caméra embarquée sur le sous-marin.

1. Choix possible :

Dans la suite de ma réflexion sur l'interface de commande et la gestion de celles-ci, cela m'a ouvert deux possibilités :

- **La première solution** consistait à mettre en place le système embarqué sur la carte pilote RaspberryPI contrôlant à la fois les commandes de boutons et la gestion de l'affichage camera. Mais cette solution pose une seconde problématique possible, à savoir : de quelle manière allons-nous transmettre nos commandes de direction et de puissance ? La RaspberryPI est effectivement capable de transmettre plusieurs informations en même temps mais sachant que nous disposons d'un système STM32 pour la gestion des commandes embarquées et d'une PI-Zero pour la gestion vidéo. Allons-nous avoir des problèmes de compatibilité de communication ?
- **La seconde solution** s'inscrit dans la prudence afin d'être sûr de réaliser les communications entre la télécommande et le drone. Celle-ci consiste à mélanger deux systèmes ARM à savoir une carte STM32 pour faciliter l'échange de données de commande en temps réel et la RaspberryPI nous permettant de réaliser l'échange et l'affichage de la caméra embarquée ainsi que de la gestion complète de la télécommande pour laquelle les données seront émises via 2 antennes distinctes.

2. Les caractéristiques de la télécommande :

- Le boîtier sera en composite de plastique imprimable en 3D
- Carte RaspberryPI 3 model B+
- Carte STM32F1 Value Line Discovery
- Carte de prototypage sur laquelle seront branchés les boutons
- 4 LED affichant le niveau de batterie
- 1 LED signalant la connexion établie avec le drone (transmission de données)
- 2 Joysticks potentiométrique miniature
- 2 gâchettes boutons
- 2 gâchettes boutons variables du type « Playstation3 »
- 1 écran LCD compatible RaspberryPI3
- 2 modules Tx/Rx équipé d'antenne (2)
- 1 batterie lithium-ion assurant l'autonomie

3. Le développement

Afin de répondre à la problématique d'émission et de réception des données de commande ainsi que du retour du flux de la vidéo embarquée, nous avons implémenté la solution suivante :

La solution choisie à mettre en œuvre deux systèmes indépendants permettant de gérer les commandes et le flux vidéo.

Cette solution s'inscrit dans la prudence mais aussi dans la rapidité de production alliant plusieurs protocoles de communication (UART, Wifi, RF « 868MHz ») mais aussi deux systèmes d'exploitation prévue, un système YOCTO nous permettant la gestion vidéo ainsi que de gérer le retour caméra embarquée avec ses boutons de capture « Screen shot » et « Recording cam » sur la télécommande permettant à l'utilisateur de développer les possibilités qu'offre la télécommande. Le deuxième système mis en œuvre est une trame de communication permettant la gestion des commandes et des batteries embarquées (drone & télécommande).

Suite à l'évolution de la télécommande, la STM32F103 a été changée par la STM32F1 Value Line DISCOVERY nous permettant de maintenir la carte opérationnelle grâce à la partie debug permettant d'envisager la possibilité d'installer un bouton test d'implémentation via les LED qui sont positionnées sur la carte, ainsi qu'un bouton reset (maintenance embarquée).

L'organisation de celle-ci a évolué dans la pratique afin d'optimiser une meilleure maniabilité (Mode TANK) :

L1 : Descente **R1** : Montée

J1 : Contrôle de la puissance du moteur droit

J2 : Contrôle de la puissance du moteur gauche

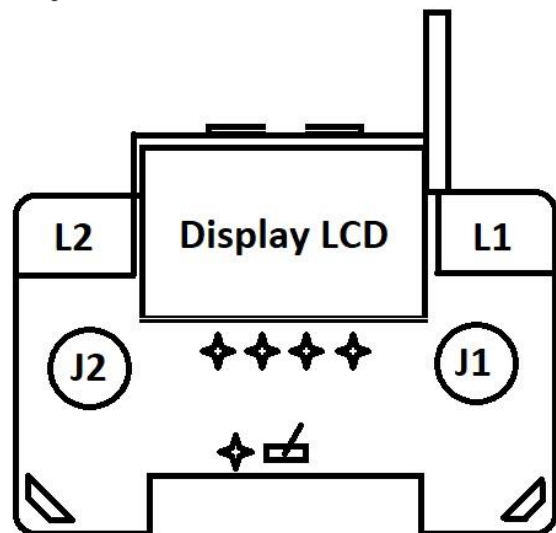


Figure 26 : Dessin de la manette

L'évolution du boîtier 3D SOLIDWORKS est pratiquement finaliser comme vous pouvez le voir ci-dessous. Une fois finalisé, celui-ci comportera l'ensemble des cartes et du circuit PCB implémentant les boutons de commande des directions.

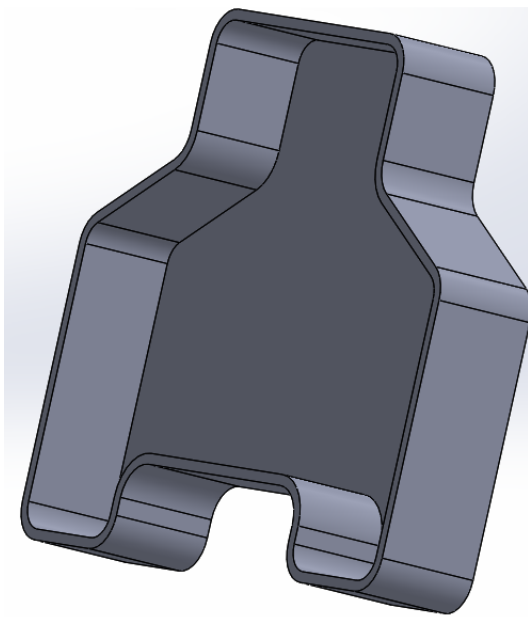


Figure 27 : Pièce arrière

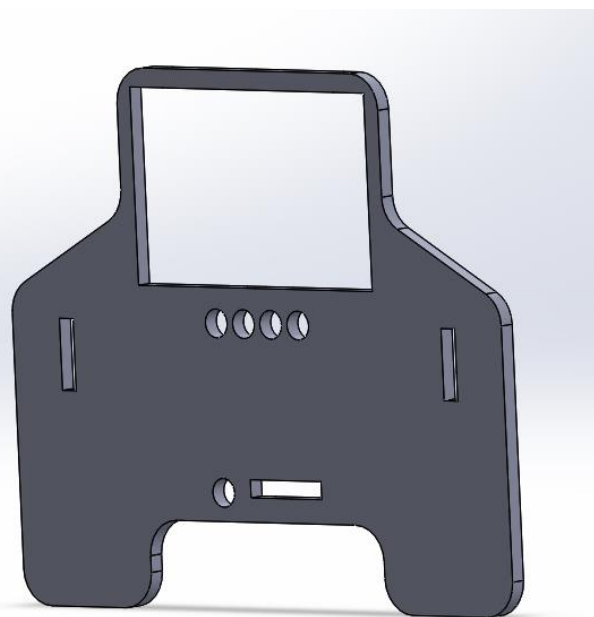


Figure 28 : Pièce avant

Afin de visualiser notre PCB de la carte implémentant les commandes de direction du drone, le schéma en cours de réalisation sur ALTIUM sera orienté sous cette forme :

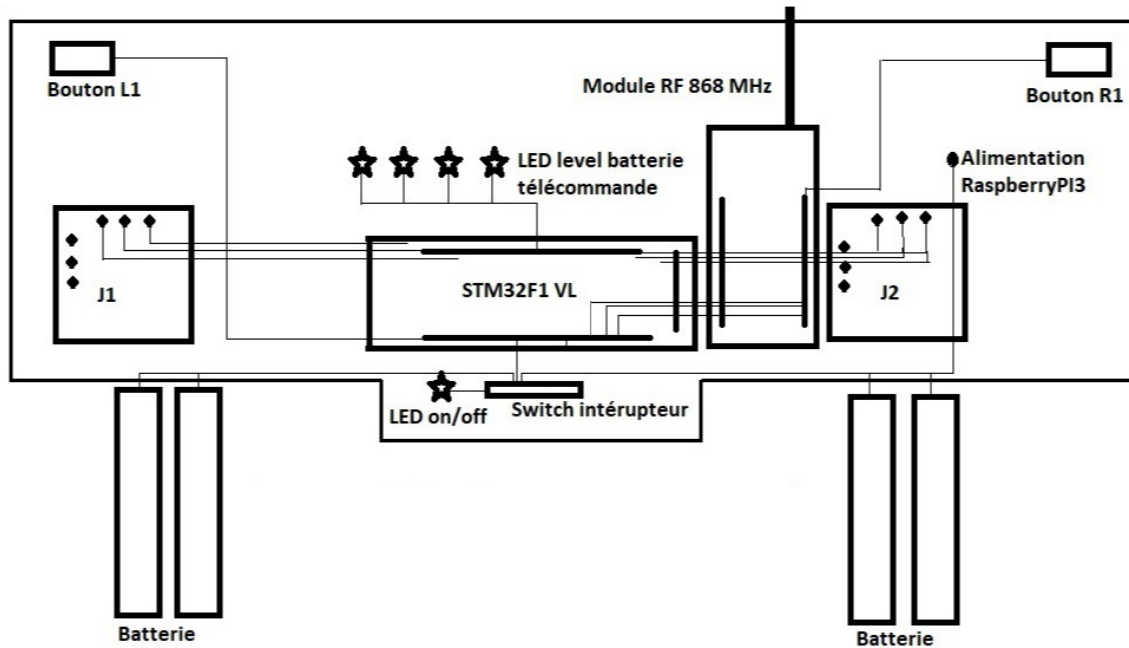
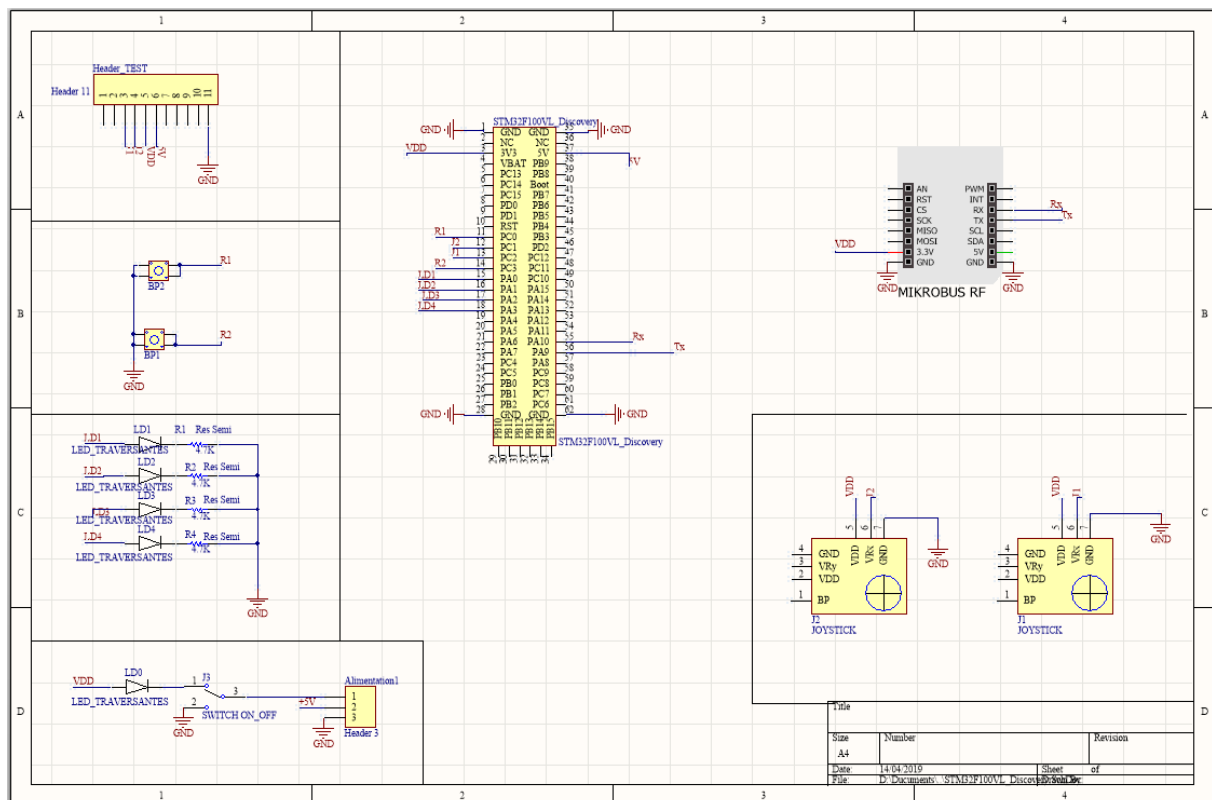


Figure 29: Schématique du PCB télécommande



III – carte de commande STM32

Pour la partie d'implémentation des commandes afin de commander le drone, cette partie est en cours de réalisation, via l'interface CubeMX liés à Keil μ vision.

Via cubeMX, l'interface nous permet de mettre en place l'initialisation des broches pour ainsi faciliter une bonne base d'implémentation du code sur Keil. Par la suite après avoir instancié les diverses commandes analogiquement, il faudra convertir les données via un tableau que nous avons pensé pour nous permettre une meilleure vitesse de calcul et de réalisation des communications.

XI. Clément MILLEREUX : Déplacements

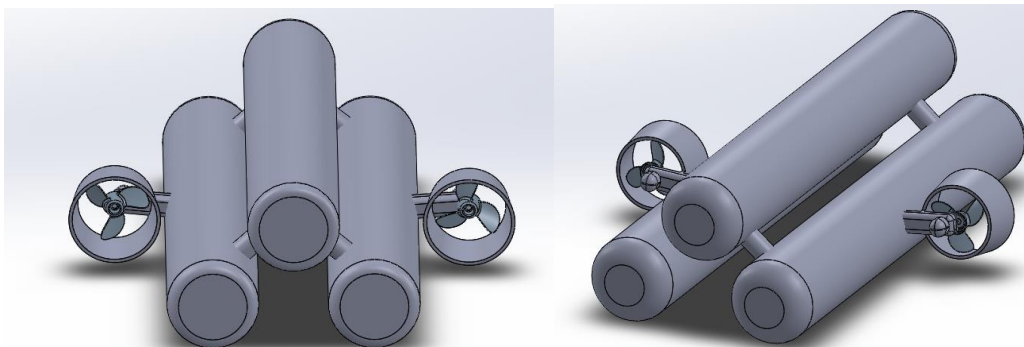
1. Choix à effectuer :

Dans ce projet, je suis en charge de la partie déplacement du projet. Le drone doit pouvoir se déplacer sur tous les axes d'où mon choix pour les pods Graupner 2335 pouvant faire tourner la tuyère sur 360° grâce à un servomoteur rapide. Un moteur coupleux de 1130kv permettra de propulser facilement le sous-marin même s'il y a un peu de courant grâce au rapport de transmission 3:1. Ci-dessous le groupe propulseur :



Ayant participé à l'élaboration du châssis, j'ai décidé de placer les pods à l'horizontal ce qui permettra de diriger le sous-marin à la manière d'un char, par exemple, pour faire une rotation statique il faudra mettre un moteur en marche avant et l'autre en marche arrière. Les pods dirigés vers le haut feront plonger le sous-marin en statique comme en déplacement. Même topo pour émerger, il faudra diriger le flux sortant des tuyères vers le bas.

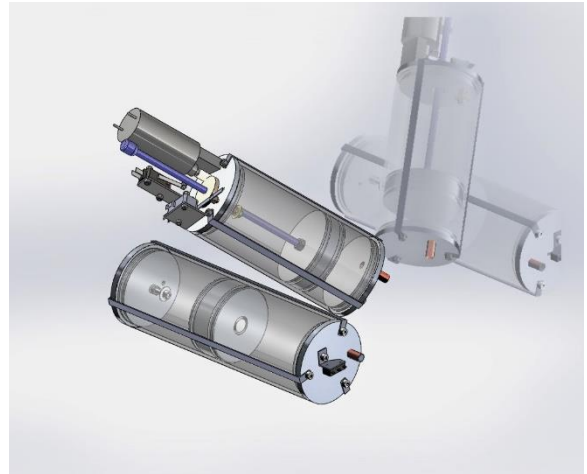
Voici une image plus explicite du placement des pods sur le châssis :



Ces pods doivent contrer la poussée d'Archimède qui est une force, couplée à la gravité, pousse vers le haut tout corps rentrant dans un liquide ou un gaz. Dans notre cas on doit gérer la poussée d'Archimède de la manière suivante :
 F_p correspond à la masse de l'objet (vers le bas)
 F_a est la poussée d'Archimède (vers le haut)

- Si la densité du solide est supérieure à celle du fluide, alors $F_p > F_a$ et le solide coule.
- Si la densité du solide est égale à celle du fluide, alors $F_p = F_a$ et le solide demeure immobile ; il est en équilibre neutre ou indifférent.
- Si la densité du solide est inférieure à celle du fluide, alors $F_p < F_a$ et le solide remonte vers la surface.

J'avais recherché quel type de ballast serait le mieux pour notre projet, et étant donné que notre châssis est tubulaire, j'ai opté pour ce type de ballast à piston motorisé qui fiabilise au mieux ce système :



2. Configuration retenue :

Les pods seront utilisés pour se déplacer de manière optimale sous l'eau. Couplés aux ballasts, ils assureront une agilité certaine en plongée totale.

En effet, les ballasts permettront d'alourdir l'appareil et donc d'avoir une flottabilité neutre ou négative, ce qui le fera plonger, puis l'on pourra se déplacer à notre guise grâce aux tuyères rotatives.

Mon choix s'est porté sur un ballast à piston pour sa fiabilité. Lors d'un échange d'e-mail avec la personne vendant les ballasts à piston, celle-ci m'a conseillé d'utiliser, pour des raisons budgétaires, des ballasts Baxter constitués d'une poche de perfusion à trouver en pharmacie.

Le souci avec ce système est le débit de l'eau qui est assez lent, rendant le déplacement sous l'eau et la plongée/remontée statique ou en mouvement trop lente !



Les modèles vendus par des marques RC affichent des prix en moyenne de 400€.

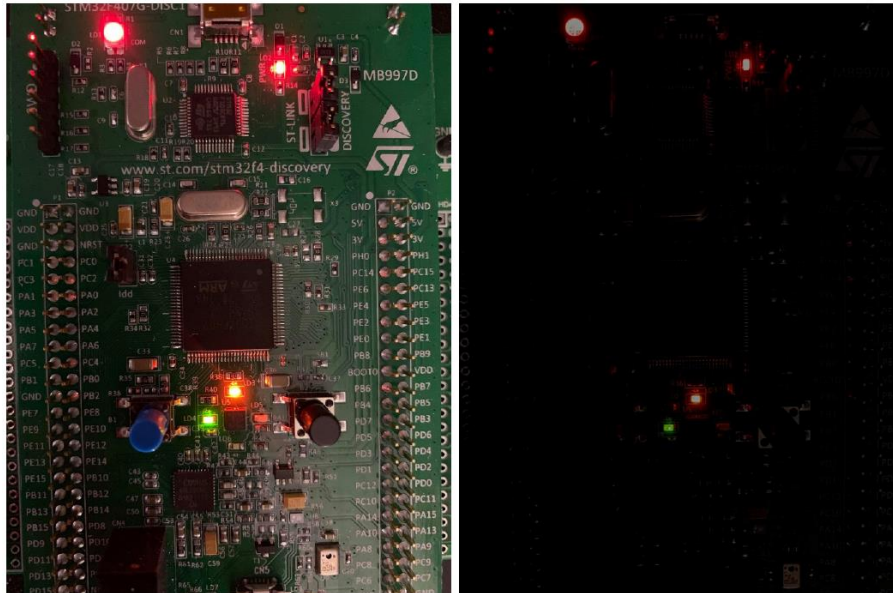
Je suis donc rentré en contact avec une personne qui fabrique elle-même ses propres ballasts à piston comme sur l'image ci-après :

Ses modèles sont en vente à 150€ pièce, et ce, peu importe la taille du réservoir jusqu'à 800ml.



Cette dernière semaine, je me suis concentré sur l'élaboration d'une PWM sur une STM32F407VG pour gérer la luminosité des différentes LEDs à chaque fois que l'on appuie sur le "user Button".

Ceci me permettra de me rapprocher du code à créer pour gérer les moteurs.



Ci-dessus une image des deux LEDs avec différentes PWM. On peut apercevoir deux taux de luminosité différents. La LED verte est plus faible que la LED rouge.

3. Tests

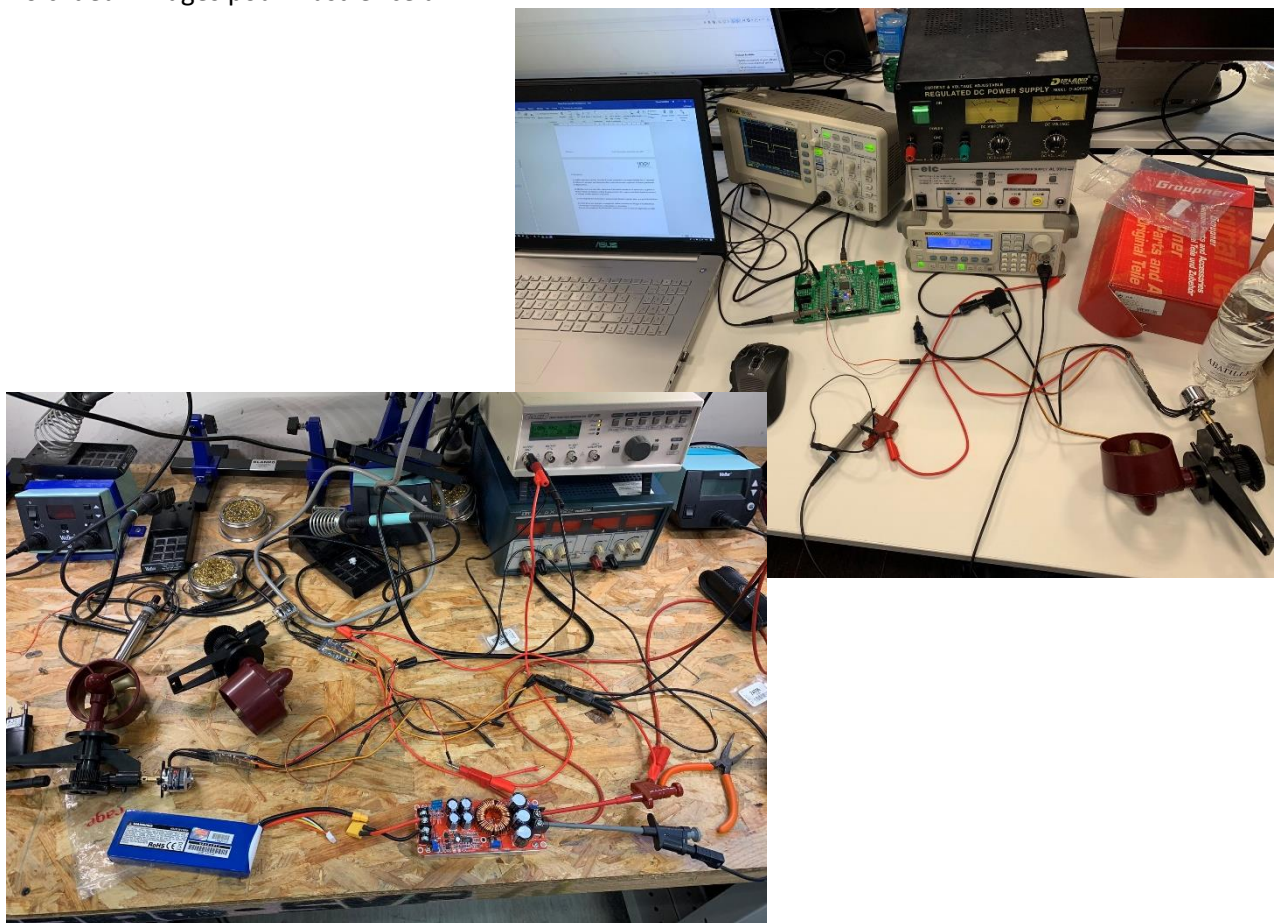
Moteurs, ESC et pods ont été montés afin de procéder aux premiers tests concernant le montage, la PWM et la consommation réelle.



La partie POD, étant trop juste pour rentrer dans le tube inférieur du sous-marin d'un diamètre de 80mm, il faudra changer les deux tubes inférieurs pour des tubes de 100mm de diamètre.

Pour ce qui est du montage de test, j'ai premièrement testé un moteur sur une batterie avec mon collègue Éric, mais aussi j'ai pu utiliser un code PWM sur STM32F407 afin de piloter l'ESC.

Voici deux images pour illustrer cela :



J'ai essayé de porter mon programme sur la STM32F100RBT6 discovery mais lorsque je connecte la carte à mon PC, celle-ci n'est pas correctement reconnue à travers les logiciels System Workbench et STM32CubeMx. J'ai donc dû créer une carte custom sur System Workbench mais à la moindre déconnexion, ce problème réapparaît de façon récurrente. Même souci sur Keil μ Vision 5. Mon collègue Marc-Antoine a semble-t-il une solution à me fournir.

J'ai donc choisi de régler ce problème plus tard en améliorant mon code (sur STM32F407VG) en faisant en sorte que la PWM change dynamiquement lorsque l'on appuie sur le bouton "User".

Ci-dessous quelques captures afin d'illustrer mon travail.

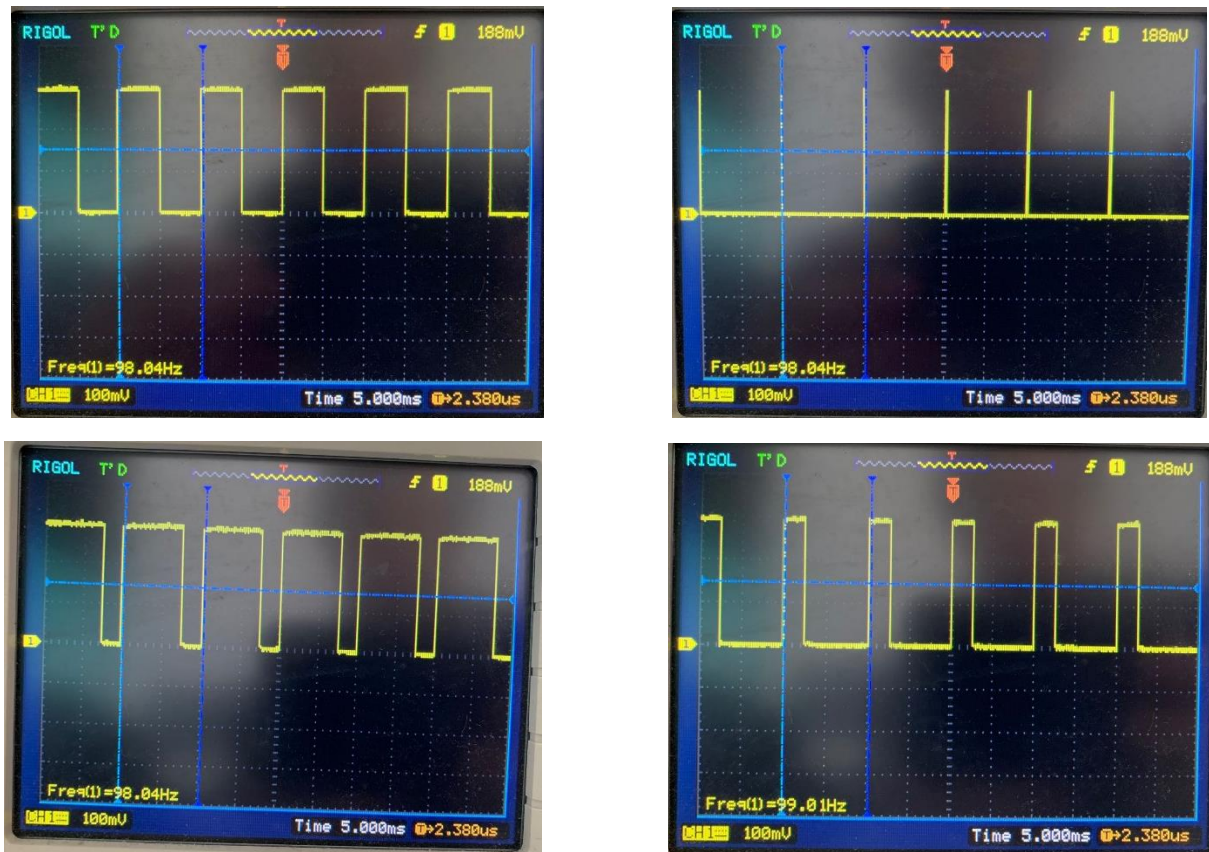


Figure 30 Capture PWM dynamique

En modifiant le châssis pour y intégrer des plaques de plexiglass afin que la caméra puisse filmer les fonds marins, j'ai cassé un des bouchons permettant l'étanchéité de l'appareil. Je dois donc aller racheter un bouchon d'une valeur de 3€ et le repercer.

XII. Thomas COUTANT : Mise en place du flux vidéo et de l'alimentation de la télécommande

Lors de ces dernières semaines, je me suis concentré en priorité sur la partie du flux vidéo. Puisque c'est la partie qui m'est la plus inconnue, et donc j'ai le plus d'appréhension. Pour cette partie des solutions ont été trouvées et ne devraient plus bouger.

- L'utilisation d'un câble : Il a été décidé d'utiliser un câble qui relierait le drone à la surface. Pour deux raisons principales.
- La première est d'ordre préventive, le câble permettra d'assurer la remontée du drone à la surface en cas de problème sous l'eau.
- La deuxième est d'ordre technique, le câble nous permettra de faire abstraction du milieu dans lequel se trouve le drone puisque grâce à ce câble nous pourrions transmettre toutes les informations, qui sont préalablement reçues au sol suite à un envoi de la télécommande.

1. L'utilisation d'une Raspberry :

De mon côté, j'ai décidé d'utiliser une Raspberry pour gérer l'envoi de la vidéo sur la télécommande et le traitement de la vidéo. En effet, en utilisant une Raspberry pi zéro w, je pourrais répondre à ces problèmes, pour un bas prix, ce qui laissera plus de liberté pour le reste.

Voici quelques photos pour en avoir un aperçu :



Figure 31: Bureau Raspbian sur l'écran Raspberry



Figure 32: Photo prise par la caméra Raspberry

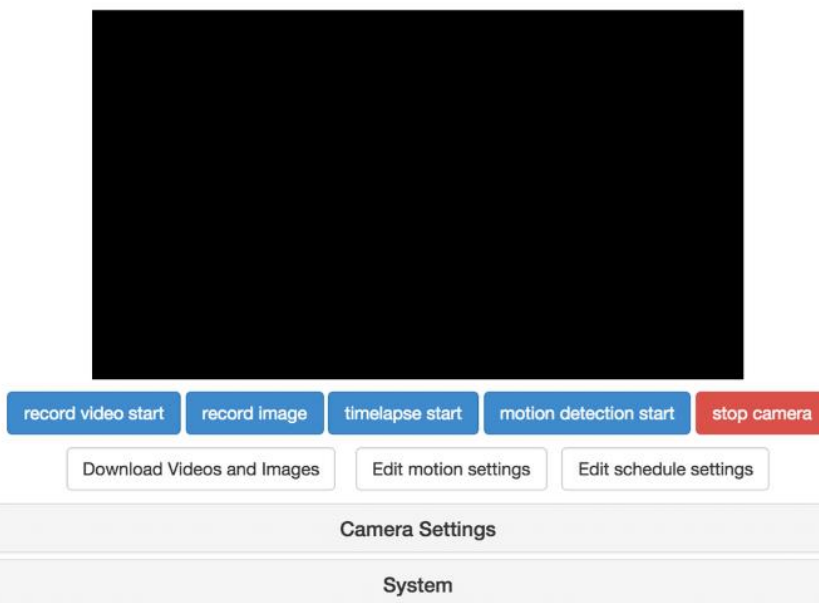


Figure 33: Retour vidéo caméra Raspberry

Pour la Raspberry, je m'oriente vers l'utilisation de YOCTO. Mais je ne suis qu'au début de mes recherches, j'attends donc dans les prochains jours d'en apprendre plus. Pour avoir un avis plus critique sur le sujet, qui répondra aux attentes du projet.



Pour ce qui est de l'alimentation je ne m'y suis pas encore véritablement plongé. Cependant avec mes camarades nous avons pu trouver une carte qui répondrait à nos attentes. Puisque Damien qui s'occupe de la télécommande partirait sur une Raspberry pi 3B+, sur laquelle nous pourrions intégrer directement une alimentation externe comme celle-ci :



Figure 34: Alimentation Raspberry

2. Choix réalisé et test :

J'ai d'abord continué mes tests sur Yocto en essayant d'établir la connexion wifi, sur ma Raspberry (wpa_supplicant). Malheureusement après plusieurs tests, je n'y suis pas arrivé puisqu'il semblait me manquer une partie importante lors de la réalisation de l'image, le network interface. Ensuite, je me suis davantage porté sur l'initialisation de la caméra au démarrage de la carte. Sur ce point j'ai réussi à trouver où cela se passait mais toutes les modifications nécessaires ne se font pas.

Par la suite, après discussion avec mon groupe et des élèves de ma classe, je me suis dit qu'il valait mieux que je laisse tomber Yocto pour le moment, pour avancer plus facilement sur le projet. Essayer de réaliser ce qui m'est demandé et une fois finie je pourrais envisager l'utilisation de yocto, qui n'est pas impératif pour mon projet. De ce fait, je suis parti sur l'utilisation d'image Debian classique utilisé sur Raspberry.

Une fois retourné sur Debian, j'ai rapidement réussi à installer la caméra puisque c'est quelque chose que j'avais déjà fait au début du projet. Je me suis donc mis à faire du streaming depuis la Raspberry vers mon ordinateur. Pour cela j'ai trouvé plusieurs façons de faire :

- l'utilisation de VLC : Cette méthode est facilement abordable mais présente un fort inconvénient puisqu'il y a un délai de cinq secondes.
- l'utilisation de netcat et mplayer : Cette méthode m'a donné plus de fil à retordre, mais donne malheureusement les mêmes problèmes qu'avec VLC.

A ce stade, il me fallait réduire ce délai mais je ne savais pas comment m'y prendre. J'ai donc poursuivi mes recherches sur le stream vidéo, j'ai fini par trouver une solution intéressante en utilisant python. En effet, il y a des bibliothèques qui me permettraient de réaliser du stream vidéo en wifi. Je me suis donc lancé dans cette idée et j'ai fini par réussir à avoir un stream vidéo sans délai.

Une fois cela réalisé, je l'ai montré à mes camarades qui m'ont orienté vers un problème que je n'avais pas vu puisque pour réaliser mon stream vidéo je connecte la Raspberry et mon ordinateur à un même réseau wifi, celui de mon téléphone.

Or le but que nous cherchons à atteindre est de connecter deux Raspberry entre elles sans aides extérieures. Il me faut donc créer un hotspot avec une des Raspberry.

La création de ce hotspot me fournira un réseau auxquelles me connecter depuis n'importe quel appareil possédant le wifi et donc ma Raspberry. Pour cela Raspberry fournit rasp-ap qui nous configure directement la carte en hotspot. Par la suite, il ne reste plus qu'à faire quelques modifications pour avoir un hotspot personnel.

J'ai donc fini par avoir une carte Raspberry (hotspot) branchée à la caméra qui renvoie la vidéo sur son réseau et une carte Raspberry qui se connecte directement au hotspot de la première et réalise la lecture de la vidéo.

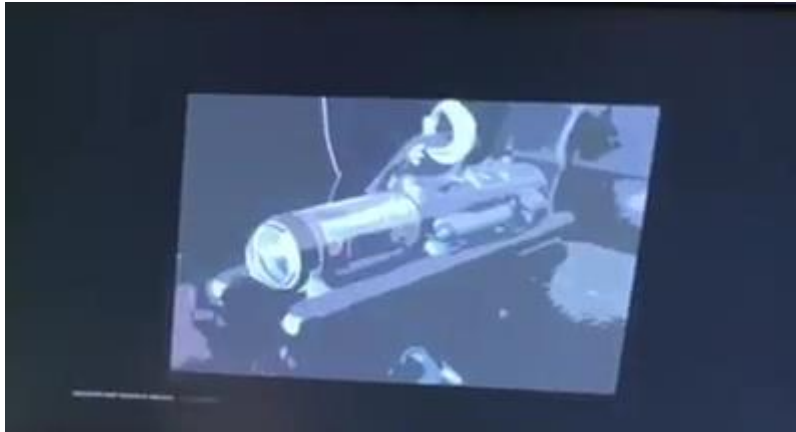


Figure 35 : Splash screen de démarrage

Raspberry Pi - Retour video drone



Figure 36 : Retour vidéo

Après cela j'ai essayé de mettre en place un bouton qui permettrait de réaliser un enregistrement. Cependant, après plusieurs jours je n'y suis pas parvenue, mais je pense avoir trouvé la cause de ce problème. Malgré tout, je ne peux y remédier puisque cela me prendrait beaucoup trop de temps. Car je devrais reprendre en grande partie l'envoi de la vidéo. J'ai donc décidé de ne pas mettre de bouton pour l'enregistrement puisque ce n'est qu'un plus.

Ensuite, j'ai vérifié le protocole de communication de la caméra. Et j'ai réalisé que le câble reliant la caméra à la Raspberry (câble CSI) est un protocole privatisé, nous ne pouvons donc pas en tirer toutes les informations importantes que nous désirons.

Malgré tout, j'ai réussi à trouver le protocole de communication qui se trouve être de l'I2C (voir de l'I3C). Or la distance de communication de ce protocole n'est pas du tout suffisante puisque l'on est de l'ordre de 2m à 4m. Et nous voulons une distance de 10m (voir cahier des charges).

De ce fait je dois changer la mise en place de mon dispositif. Le câble CSI ne conviendra pas pour faire les 10m. Je me suis donc penché sur d'autre moyen de transmission par câble sur la Raspberry. Il y a donc deux possibilités qui s'offre à nous : câble Ethernet et câble usb. Le câble Ethernet semble beaucoup plus approprié puisque l'on peut transmettre sur de très long distance (km) or le câble usb lui ne peut transmettre que sur 5 mètres même si l'on peut raccorder jusqu'à 5 câbles et donc avoir 25m de câble. Cependant le câble Ethernet à l'aire d'une plus grande rigueur pouvant avoir un débit plus important et ne pas être dérangé/perturbé sur le trajet.

Suite à cela, j'ai d'abord commencé par faire un test, avec seulement le câble Ethernet.

Le résultat était convenable avec un retour vidéo fluide et limpide. Le hotspot, lui de son côté fonctionne déjà, je n'ai rien touché de ce côté-là. Il ne me restait donc plus qu'à associer les deux ensembles. Malheureusement après plusieurs essais et tentatives je n'ai pas réussi à faire fonctionner les deux en symbioses. Cela résulte de la configuration du hotspot, puisque celui-ci inhibe la connexion Ethernet en se configurant. J'ai donc passé le reste de mon temps à essayer de trouver un hotspot qui puisse accepter une communication Ethernet en parallèle de la sienne.

Je suis tombé sur une autre solution qui fait cohabiter la connexion Ethernet et le hotspot (ConnMan). ConnMan est un utilitaire de gestion des connexions réseaux, je pourrais donc paramétrer mes connexions comme je le souhaite en gardant la main et un œil sur tout ce qui se fait.

Malgré tout je me heurte à un problème de driver internet. Après plusieurs discussions il semblerait que ce soit un problème de kernel. En effet ma version ne serait pas stable avec le driver qui gère les connexions internet.

XIII. Conclusion

En conclusion, l'avancement du travail global correspond au planning prévu. Chacune des tâches que l'on doit réaliser possède plusieurs solutions possibles et réalisables.

Par la suite du projet, nous allons devoir choisir les solutions adéquates en fonction des contraintes du cahier des charges.

Pour la revue de projet n°3, nous avons pris en considération les solutions que nous avons trouvées en revue de projet n°2. Ainsi nous avons pu faire des choix, notamment concernant les cartes mères pour le système de contrôle du drone sous-marin ainsi que la motorisation du sous-marin qui sont, on le rappelle, deux pods directionnels et deux ballasts. Sans oublier le flux vidéo, qui se fera à l'aide de deux Raspberry Pi et d'un écran.

Toutes ces fonctions respectent le cahier des charges établi lors de la première revue de projet.

L'équipe est soudée. Chacun a le souci de sa tâche et aussi une écoute attentive aux interrogations de l'autre, ce qui élargit notre champ de recherches et de collaboration, encourageant et appréciable. Tous ces aspects nous confortent dans l'avancement et la réussite du projet.