

# Guide utilisateur

DRONE HYDROGRAPHIQUE DE SURFACE  
MAI 2023 – OCTOBRE 2023

ENCADREMENT | Willian N. Cassols

## Table des matières

|        |                                                         |    |
|--------|---------------------------------------------------------|----|
| 1.     | Abréviations.....                                       | 4  |
| 2.     | Introduction.....                                       | 4  |
| 3.     | Première partie : prérequis .....                       | 5  |
| 3.1.   | Liste du matériel .....                                 | 5  |
| 3.2.   | Préparation de la station de bord .....                 | 8  |
| 3.2.1. | Mission Planner .....                                   | 8  |
| 3.2.2. | Installer les logiciels requis .....                    | 11 |
| 3.2.3. | Télécommande .....                                      | 11 |
| 3.2.4. | Organisation des dossiers pour le post-traitement ..... | 15 |
| 3.3.   | Préparation du Raspberry Pi.....                        | 16 |
| 3.3.1. | Flasher une image.....                                  | 16 |
| 3.3.2. | Connexion ssh.....                                      | 17 |
| 3.3.3. | Installer les librairies python nécessaires .....       | 19 |
| 3.3.4. | Intégration du GPS.....                                 | 20 |
| 4.     | Deuxième partie : premier lancement .....               | 21 |
| 4.1.   | Branchements.....                                       | 21 |
| 4.1.1. | Raspberry Pi.....                                       | 22 |
| 4.1.2. | Pixhawk.....                                            | 23 |
| 4.2.   | Initialisation (ESC) .....                              | 24 |
| 4.3.   | Calibration onshore .....                               | 24 |
| 4.3.1. | Orientation Pixhawk (AHRS_ORIENTATION) .....            | 24 |
| 4.3.2. | Calibration accéléromètre .....                         | 25 |
| 4.3.3. | Calibration GPS .....                                   | 25 |
| 4.3.4. | Calibration Radio .....                                 | 26 |
| 4.3.5. | Flight Modes .....                                      | 26 |
| 4.4.   | Calibration offshore .....                              | 26 |
| 4.4.1. | Calibration vitesse .....                               | 26 |
| 4.4.2. | Calibration des virages.....                            | 27 |
| 5.     | Troisième partie : Mission autopilote .....             | 28 |
| 5.1.   | Créer une mission .....                                 | 28 |
|        | Paramètres supplémentaires.....                         | 28 |
| 5.2.   | Lancer la mission .....                                 | 28 |

|      |                                                    |    |
|------|----------------------------------------------------|----|
| 5.3. | Interrompre une mission en cours .....             | 30 |
|      | Avec la télécommande .....                         | 30 |
|      | Sans télécommande .....                            | 30 |
| 5.4. | Acquisition de la profondeur .....                 | 30 |
| 6.   | Quatrième partie : Après la mission .....          | 31 |
| 6.1. | Télécharger les logs.....                          | 31 |
| 6.2. | Analyse des logs.....                              | 32 |
| 6.3. | Post traitement de la profondeur avec Python ..... | 32 |
| 6.4. | Représentation de la donnée .....                  | 34 |
| 7.   | Cinquième partie : Troubleshooting .....           | 34 |
| 7.1. | Pilote automatique .....                           | 34 |
| 7.2. | Raspberry Pi.....                                  | 35 |

## Table des figures

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| Figure 1 Différentes parties du drone.....                                                  | 6  |
| Figure 2 Choisir le bon port .....                                                          | 8  |
| Figure 3 Onglets de l'interface MP .....                                                    | 8  |
| Figure 4 Actions MP .....                                                                   | 9  |
| Figure 5 Log Path .....                                                                     | 10 |
| Figure 6 Log avec affichage des données "GPS" .....                                         | 11 |
| Figure 7 Télécommande FS-iA6b. Gauche: page d'accueil. Droite: menu principal. ....         | 12 |
| Figure 8 Signal a modulation de largeur d'impulsion .....                                   | 12 |
| Figure 9 Fonctions auxiliaires – erreur : « CH : 7 » et non « 8 » comme sur la photo! ..... | 13 |
| Figure 10 Brider les moteurs .....                                                          | 15 |
| Figure 11 Organisation des dossiers .....                                                   | 15 |
| Figure 12 Balena-etcher .....                                                               | 16 |
| Figure 13 Insérer la carte SD.....                                                          | 17 |
| Figure 14 Ajouter un réseau WIFI.....                                                       | 18 |
| Figure 15 Adresse IP dynamique .....                                                        | 18 |
| Figure 16 Branchement pour le système Raspberry Pi.....                                     | 22 |
| Figure 17 Branchements pour le système autopilote.....                                      | 23 |
| Figure 18 Référentiel (axes) .....                                                          | 24 |
| Figure 19 AHRS_ORIENTATION pour la plateforme finale .....                                  | 25 |
| Figure 20 Setup des modes de pilotage .....                                                 | 26 |
| Figure 21 Réorganiser les WP .....                                                          | 28 |
| Figure 22 Menu pour l'écriture des WPs .....                                                | 29 |
| Figure 23 Démarrage de la mission avec MP .....                                             | 30 |
| Figure 24 Démarrage d'une acquisition par fenêtre de commande .....                         | 31 |
| Figure 25 Téléchargement des logs. ....                                                     | 32 |
| Figure 26 "Review a log" .....                                                              | 32 |
| Figure 27 Diagramme de traitement de l'information. ....                                    | 33 |
| Figure 28 Carte bathymétrique .....                                                         | 34 |

## 1. Abréviations

WP = Waypoint

MP = Mission Planner

ESC = Electronic Speed Controllers (Variateurs Electroniques de Vitesse)

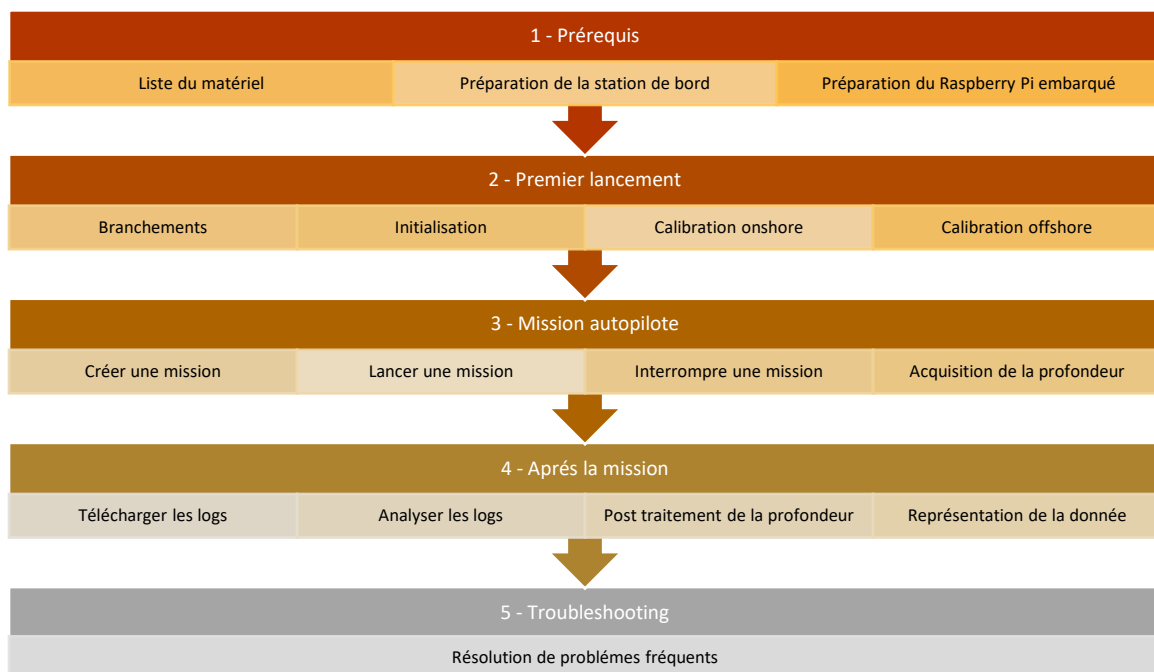
Rpi = Raspberry Pi

## 2. Introduction

Ce guide utilisateur est divisé en cinq parties qui regroupent l'ensemble des connaissances à avoir pour utiliser le drone hydrographique. Ces parties suivent un ordre chronologique : l'utilisateur devrait donc suivre la lecture du guide pour une prise en main optimale du drone.

La première partie est dédiée aux prérequis, c'est-à-dire à l'ensemble des installations et réglages à faire avant même de démarrer le drone sur sa première mission. Dans une deuxième partie, les procédures de calibration qui doivent être faites lors du premier lancement du drone sont expliquées. En troisième, le guide se concentre sur la notion d'autopilotage : comment planifier une mission qui s'effectue ensuite sans intervention de l'utilisateur. La quatrième partie est consacrée à l'après mission et au traitement des données. Finalement, la cinquième partie est un rapport de résolution des problèmes les plus couramment rencontrés.

Le diagramme suivant résume l'organisation général du guide :



### 3. Première partie : prérequis

Afin de faire fonctionner la plateforme, vérifiez que les prérequis présentés dans cette section sont acquis. En particulier, il faut s'assurer que vous disposez de l'ensemble des logiciels et librairies Python, à la fois pour la station de bord et pour le Rpi.

#### 3.1. Liste du matériel

La liste du matériel est organisée par sections. Il y a six sections qui correspondent aux six parties ou accessoires du drone :

1. La « boîte orange » : c'est le cerveau qui contient toute l'électronique. Elle est détachable et complètement étanche (submersion partielle ou complète). Cela étant, elle repose au-dessus du niveau de l'eau et il est attendu qu'elle ne soit mouillée que de manière exceptionnelle.
2. La « structure » : elle est composée d'une structure d'acier qui permet de poser la boîte orange sur les flotteurs. Les moteurs sont fixés à ces derniers. Elle peut être portée en un seul tenant ou démontée en plusieurs parties selon les préférences de l'utilisateur.
3. La « station de bord » : c'est l'ensemble des composantes nécessaires au pilotage du drone qui restent à quai et sont utilisés par l'utilisateur pour diriger les missions.
4. « L'alimentation » : dès lors que le drone n'est plus utilisé, les batteries sont sorties de la boîte orange et stockées indépendamment.
5. La « boîte à outil » : ensemble d'outils qui sont (ou peuvent être) utiles au montage et à l'opération du drone.
6. Les « tréteaux » : ils sont très utiles lors des déplacements de la plateforme afin d'éviter que les moteurs n'entrent en contact avec le sol.





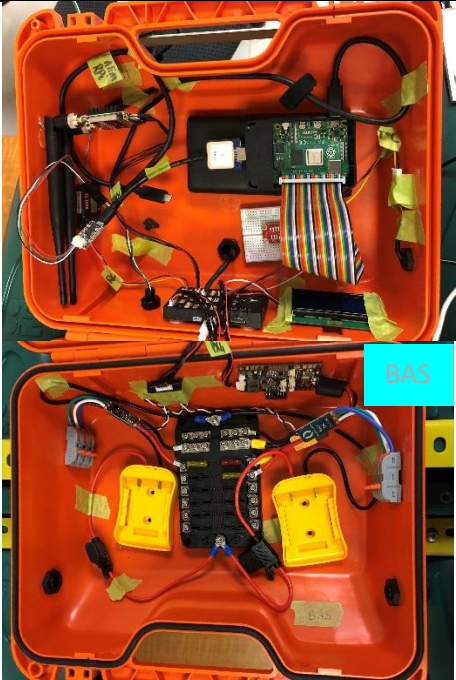
Figure 1 Différentes parties du drone

D'avantage de détails sur le contenu de chaque partie sont donnés dans le Tableau 1.

Tableau 1 Liste du matériel

| Nom | Comprend... | Image |
|-----|-------------|-------|
|-----|-------------|-------|

HAUT

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Boîte orange    | <p>1 GPS M8N</p> <p>1 sonde de température DS18B20</p> <p>1 sonde monofaisceau Ping2 avec adaptateur USB</p> <p>1 autopilote Pixhawk</p> <p>1 émetteur/récepteur radio RFD900x (≈900 GHz)</p> <p>1 récepteur radio iA6b (≈2.5GHz)</p> <p>1 Raspberry Pi 4B</p> <p>1 GPS USB low-cost</p> <p>1 écran LCD</p> <p>1 Power Bank 5V</p> <p>1 extension board</p> <p>2 supports pour batteries Li-ion 5V, 6Ah</p> <p>1 porte fusible</p> <p>2 Basic ESC</p> <p>1 Power Module PM07</p> <p>1 module d'extension I/O PWM out</p> |    |
| Structure       | <p>2 stabilisateurs de canot</p> <p>1 structure rectangulaire en acier</p> <p>1 support aluminium pour la sonde</p> <p>2 moteurs T200 sur support</p>                                                                                                                                                                                                                                                                                                                                                                    | Voir Figure 1                                                                        |
| Station de bord | <p>1 télécommande Flysky FS-iA6b</p> <p>1 émetteur/récepteur radio RFD900x (≈900 GHz)</p> <p>1 ordinateur portable avec MP</p>                                                                                                                                                                                                                                                                                                                                                                                           |  |
| Alimentation    | 4 batteries de perceuse Li-ion 20V, 5Ah                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                      |
| Boîte à outil   | <p>1 tournevis multi-embouts</p> <p>Serres fixes</p> <p>Élastiques/ Tendeurs</p> <p>1 câble de connexion USB/micro USB (Pixhawk)</p> <p>1 multimètre</p> <p>1 ruban à mesurer</p> <p>Gommette</p> <p>Velcro</p> <p>Scotch magique</p>                                                                                                                                                                                                                                                                                    |                                                                                      |
| Tréteaux        | 2 tréteaux                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Voir Figure 1                                                                        |



## 3.2. Préparation de la station de bord

### 3.2.1. Mission Planner

#### 3.2.1.1. Installation

Se rendre sur : <https://ardupilot.org/planner/docs/mission-planner-installation.html> . Suivre les instructions.

#### 3.2.1.2. Télécharger le firmware

Chaque type de plateforme (e.g. avion, drone, bateau, rover) a un firmware associé différent. Dans le cas d'un bateau, il faut télécharger le firmware Rover sur le Pixhawk. A cette fin, brancher le Pixhawk en USB, et sélectionner (chercher dans le menu déroulant COMx : le numéro peut changer) avec un baud rate de 115200. À cette étape, il est important de ne pas connecter (voir Figure 2).



Figure 2 Choisir le bon port

Aller dans « SETUP » puis « Install Firmware » et installer ArduRover.

Une fois installé, le Pixhawk peut être débranché. Au prochain branchement, aller dans « CONFIG » puis « Full Parameter List ». Chercher « FRAME\_CLASS » et lui attribuer la valeur 2 (Boat). « FRAME\_TYPE » peut être laissé à « undefined » (défaut).

Maintenant que le Pixhawk est configuré en mode bateau, il reste à le configurer en mode skid steering (c'est-à-dire tel que la direction est faite par un différentiel de puissance dans les moteurs). Pour cela, dans « CONFIG » puis « Full Parameter List » rentrer les paramètres suivants :

- SERVO2\_FUNCTION = 74 (Throttle Left)
- SERVO4\_FUNCTION = 73 (Throttle Right)

#### 3.2.1.3. Interface

L'interface MP est instinctive. Les principales fonctionnalités sont présentées rapidement dans cette section.

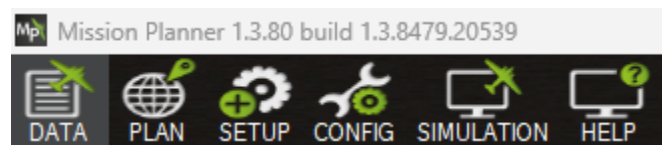


Figure 3 Onglets de l'interface MP

Lorsque l'utilisateur ouvre l'application MP, une fenêtre s'affiche : en haut à gauche de cette fenêtre, il est possible de naviguer dans six onglets différents présentées Figure 3. Parmi les six onglets, seuls les quatre premiers sont utilisés dans ce guide. Chaque onglet intervient dans des phases différentes :

- L'onglet « DATA » intervient dans la phase de pilotage du drone (autopilotage et pilotage manuel), ainsi que dans les phases post-mission (logs) et troubleshooting.
- L'onglet « PLAN » intervient uniquement pour des missions d'autopilotage.
- L'onglet « SETUP » sert à la calibration (onshore, premier lancement)
- L'onglet « CONFIG » intervient essentiellement dans la calibration (offshore, premier lancement).

L'onglet DATA regroupe une quantité d'information conséquentes qui ne sont pas toutes aussi pertinentes dans le cadre de notre projet : les plus utiles sont détaillées ci-dessous.

#### « Quick »

Il est possible de changer les paramètres qui s'affichent en temps réel dans l'onglet « Quick » en double cliquant sur chacun d'eux.

#### « Actions »

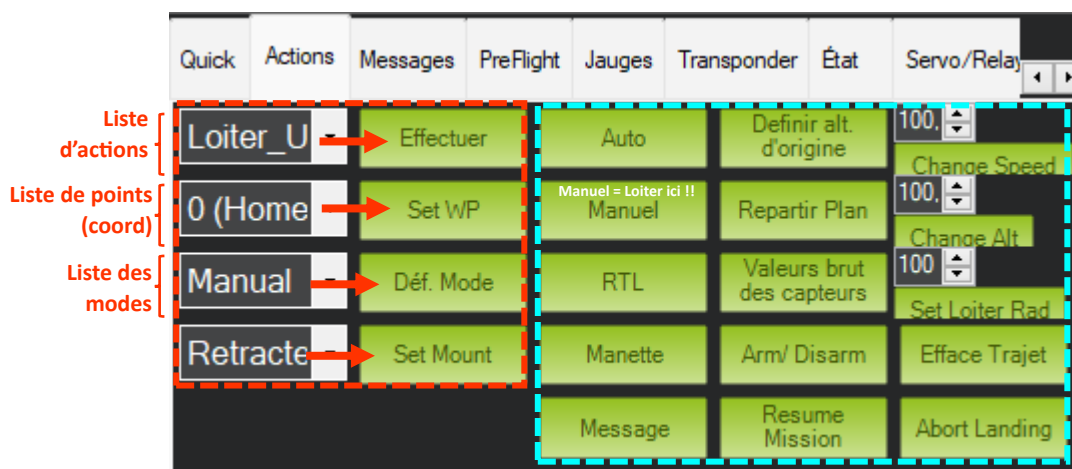


Figure 4 Actions MP

#### « Messages »

Dans cet onglet s'affichent les messages émis par le Pixhawk (initialisation, arm...), en temps réel. Permet de vérifier que tout va bien. Lorsque le Pixhawk refuse de s'armer, c'est ici que l'on peut voir l'erreur que ce dernier renvoie.

#### « État »

Affichage en temps réel de tous les paramètres visibles dans mission planner. Ces paramètres sont également ceux que l'on peut afficher dans l'onglet « M. à point » lorsque l'on double clique sur la fenêtre d'affichage.

#### « Journaux du télémetrie »

Permet de rejouer un vol. En particulier, l'onglet « M. à point » donne accès à l'évolution des paramètres tout au long de la mission.

#### « DataFlash Logs »

Cet onglet est le plus utile quand il s'agit de télécharger et d'analyser les logs enregistrés pendant la mission.

Pour définir l'information stockée dans les logs, changer le paramètre LOG\_BITMASK. Par défaut, le drone n'enregistre les données que lorsque celui-ci est armé. Pour changer cela, passer LOG\_DISARMED à 1. Pour que chaque log s'arrête au moment où le drone est désarmé, passer le paramètre LOG\_FILE\_DSRMROT à 1 : autrement un nouveau log n'est créé que lorsque le Pixhawk reboote.

Le fichier dans lequel sont sauvegardés les logs peut être modifié. Aller dans « CONFIG » puis « Planner ». Sous « Log Path » cliquer « Browse » pour changer la destination.

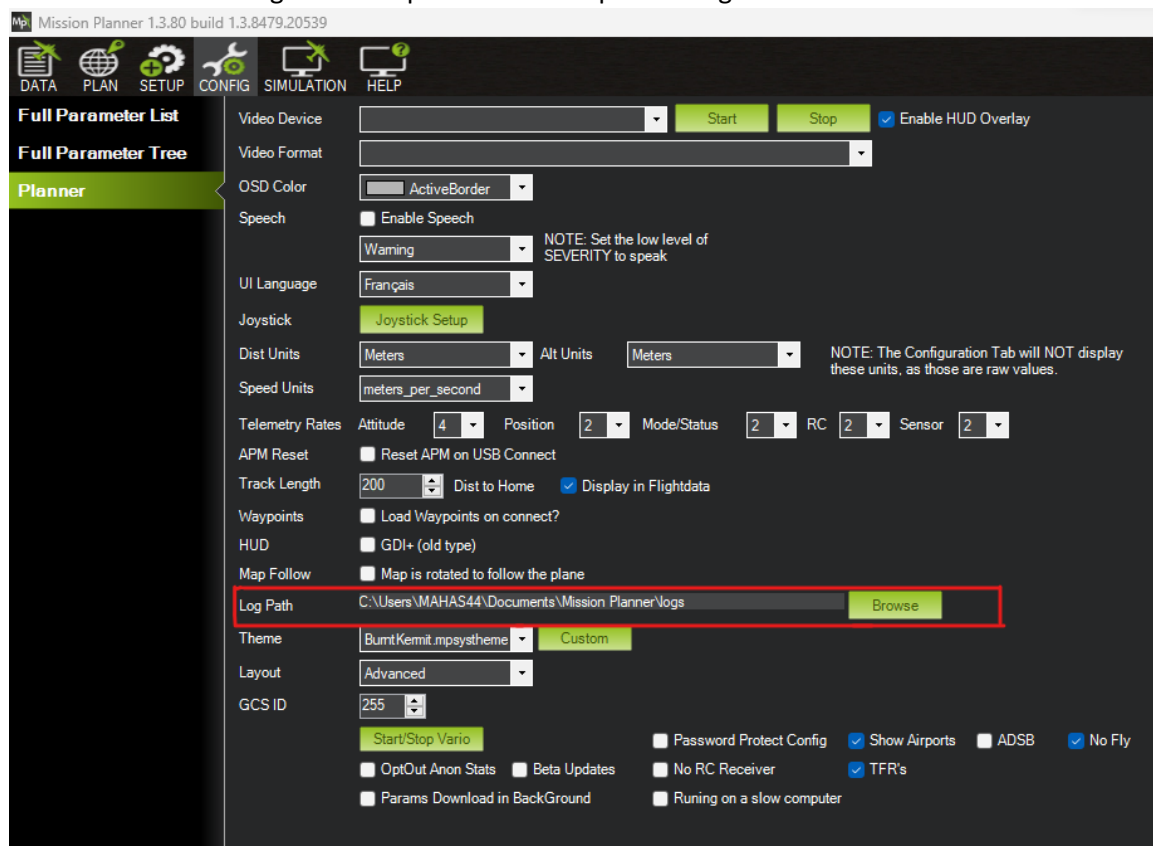


Figure 5 Log Path

Pour le téléchargement des logs, cliquer sur « Download DataFlash Log via Mavlink », puis dans la fenêtre qui apparaît, sélectionner les logs voulus et cliquer sur « Charger ces journaux ». La durée de téléchargement d'un log peut varier de quelques secondes à plus d'une heure selon la taille du fichier.

Une fois les logs téléchargés, il est possible de les analyser. Dans « Review a Log », sélectionner le log d'intérêt.

Il est alors possible de filtrer les informations : cliquer sur une colonne puis filtrer selon le paramètre voulu (ex : GPS). Pour télécharger la table ainsi obtenue, faire clic droit, puis « Export

Visible ». Un fichier sans extension est créé. La figure suivante montre l'exemple d'un log filtré selon le critère « GPS » dans MP.

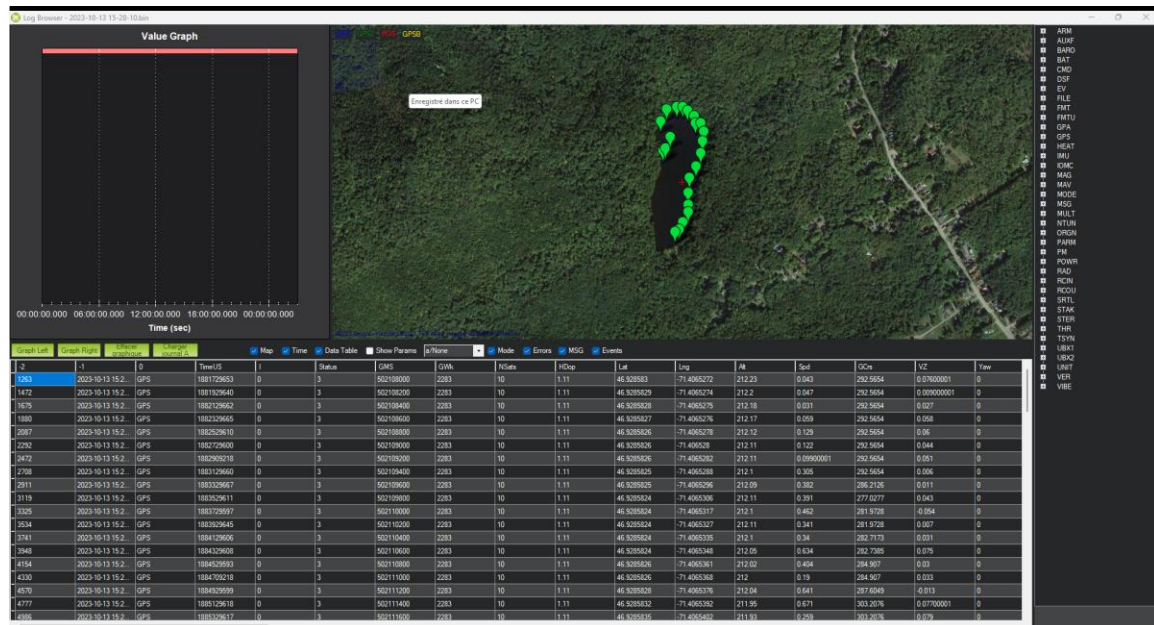


Figure 6 Log avec affichage des données "GPS"

### 3.2.2. Installer les logiciels requis

Après l'installation de MP, il reste deux applications à installer.

Premièrement, il est recommandé d'installer ping-viewer. Ce logiciel est développé par Blue Robotics et permet de vérifier le bon fonctionnement du sonar en le branchant par USB a la station de bord. Il est disponible à cette adresse : <https://docs.bluerobotics.com/ping-viewer/#installing-and-running-the-application>.

Pour le post-traitement des données, un éditeur Python est requis. PyCharm est recommandé. Il peut être téléchargé ici : <https://www.jetbrains.com/pycharm/download/?section=windows>. Seule la version « Community Edition » est disponible gratuitement.

### 3.2.3. Télécommande

Avant la mise à l'eau, il est nécessaire de paramétrer la télécommande. Celle utilisée dans le projet est présenté dans la figure ci-dessous. Les deux boutons essentiels sont ceux situés sur la droite « OK » et « CANCEL ».

Quelques règles de navigation dans les menus:

- Le récepteur ne peut pas être alimenté lors des réglages. Les menus ne sont pas accessibles lorsque ce dernier est allumé.
- Appuyer brièvement sur « OK » pour valider/ passer à l'étape suivante du menu en cours.
- Appuyer brièvement sur « CANCEL » pour sortir du menu en cours sans enregistrer les modifications.
- Appuyer longuement sur « OK » pour se rendre au menu principal.

- Appuyer longuement sur « CANCEL » pour enregistrer les modifications faites dans le menu en cours.
- « UP » et « DOWN » permettent d'ajuster les paramètres (changer la valeur) et de naviguer dans les menus



Figure 7 Télécommande FS-iA6b. Gauche: page d'accueil. Droite: menu principal.

### 3.2.3.1. Protocole de communication

Ensuite, il faut choisir le protocole de communication. La télécommande FS-iA6b supporte quatre protocoles pour la communication :

- PPM
- PWM (Pulse Width Modulation)
- I-BUS
- S-BUS

Le protocole retenu est le protocole PWM, qui signifie modulation de largeur d'impulsion. La télécommande envoie un signal créneau, similaire à celui représenté Figure 8. Il est constitué d'une série d'impulsions de largeur variable : plus une impulsion est large, plus la puissance délivrée aux moteurs sera importante. Le signal PWM est reçu par le Pixhawk et transmis aux variateurs électroniques de vitesse (ESC) qui le convertissent en une puissance. C'est cette puissance qui est délivrée aux moteurs.

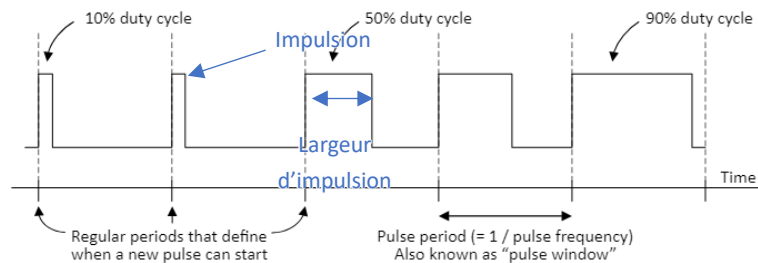


Figure 8 Signal a modulation de largeur d'impulsion

Pour choisir PWM faire :

- OK (long)
- System Setup
- RX setup
- Output mode



- Sous « Output », sélectionner PPM. Dans « Serial », laisser n'importe laquelle des deux options, ce ne sera pas le protocole utilisé.
- « CANCEL » (long) pour enregistrer

### 3.2.3.2. Fonctions auxiliaires

Les fonctions auxiliaires sont des fonctions qui peuvent être assignées aux boutons de la télécommande (hors fonctions basiques de navigation assignées par défaut aux joysticks). Par exemple, une touche de la télécommande peut être assignée à l'armement du véhicule.

Dans ce projet les fonctions auxiliaires utilisées font partie des fonctions auxiliaires définies dans ArduRover (voir [ici](#)). Trois fonctions ont été utilisées à ce jour :

- Fonction 153 ARM/DISARM
- Fonction 50 Learn Cruise (calibration)
- Fonction Modes (changement de mode)

Il faut activer trois canaux supplémentaires (par défaut, les quatre canaux pour Throttle, Roll, Yaw, Pitch sont activées et assignées aux deux joysticks). Les boutons choisis pour les trois canaux sont SWA, SWC et SWD. SWC a trois positions possibles (Haut, milieu et bas), tandis que SWA et SWD n'en ont que deux. Pour cette raison, c'est SWC qui est choisi pour le changement télécommandé des modes (trois modes sont fréquemment utilisés).



Figure 9 Fonctions auxiliaires – erreur : « CH : 7 » et non « 8 » comme sur la photo!

La marche à suivre est :



- « OK » (long)
- System Setup
- Aux switches (dernière ligne du menu)
- Up & Down pour activer ou désactiver chaque bouton. Renseigner sous « Ch : » le nombre de channel actives total (en comptant les quatre channel principaux Throttle, Roll, Pitch et Yaw).
- « CANCEL » (long) pour enregistrer

Maintenant que les boutons de la télécommande voulus sont activés, il reste à assigner chaque bouton a un channel.

- « OK » (long)
- Functions Setup
- Aux channels
- Pour chaque channel, utiliser UP & DOWN pour y associer le bouton voulu.
- « CANCEL » (long) pour enregistrer

Pour le montage fait dans ce projet, le réglage final est :

| Channel number | Source |
|----------------|--------|
| 5              | SwD    |
| 6              | SwC    |
| 7              | None   |
| 8              | SwA    |
| 9              | None   |

### 3.2.3.3. *Brider les moteurs (optionnel)*

Dans la phase de test du bateau, il est possible de brider les moteurs (i.e. limiter la puissance minimale et la puissance maximale qui peuvent être commandées par la télécommande). Les channels visées sont les channels de direction (Channel 1 à 4).

- « OK » (long)
- Functions Setup
- End Points
- Pour chaque channel, bouger le bouton associé de la télécommande pour naviguer entre la limite inférieure (colonne de gauche – reverse) et la limite supérieure (colonne de droite – forward).
- « CANCEL » (long)



Figure 10 Brider les moteurs

### 3.2.4. Organisation des dossiers pour le post-traitement

Le post traitement se fait à partir d'un code python. Il est recommandé d'adopter l'architecture de la Figure 11 pour les dossiers.

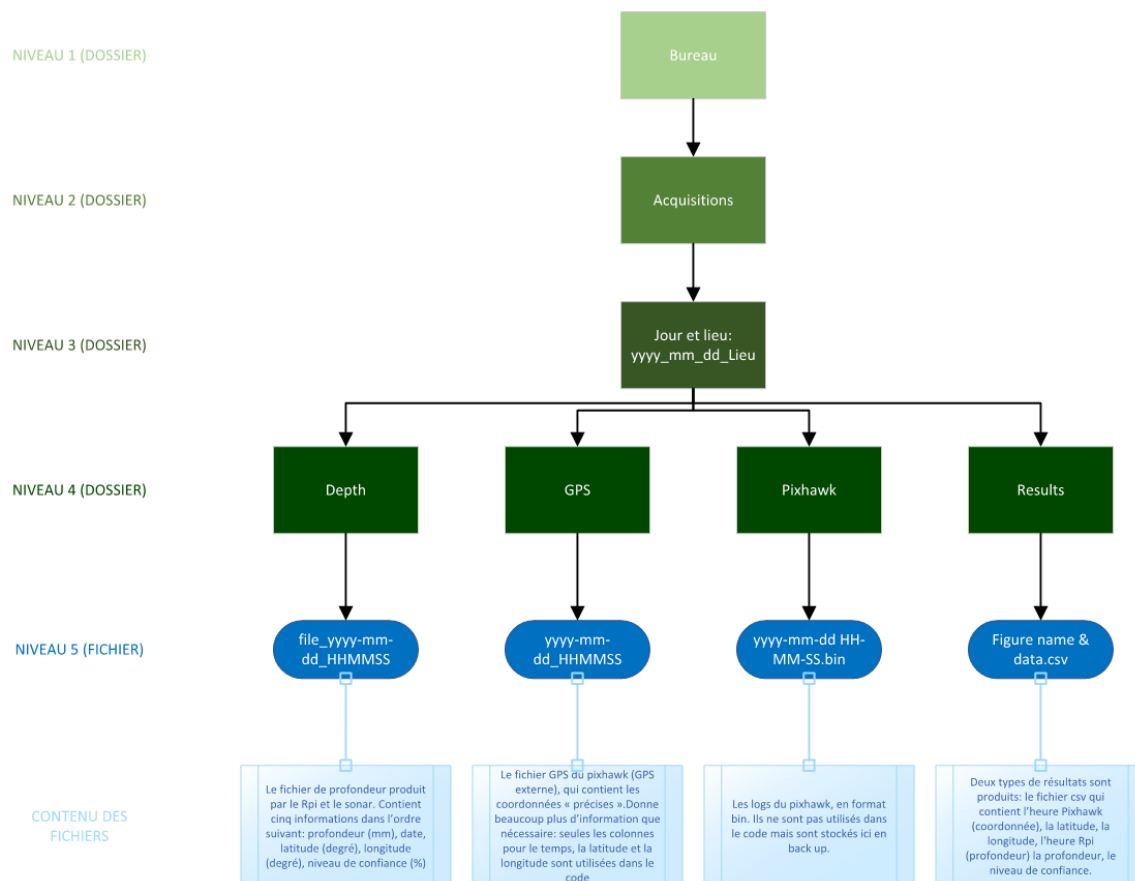


Figure 11 Organisation des dossiers

La Figure 11 montre également la nomenclature des fichiers utilisés dans le code. Pour minimiser le débogage, vérifier que les noms des fichiers respectent toujours le format indiqué.

### 3.3. Préparation du Raspberry Pi

#### 3.3.1. Flasher une image

Le Raspberry Pi marche avec une carte SD sur lequel a été préalablement flashé une image (un système d'exploitation). Raspberry Pi OS et Ubuntu sont les deux images les plus communément utilisées.

Dans ce projet, l'image choisie est une version modifiée de Raspberry Pi OS disponible [ici](#) (intitulée rpanion-server). Il s'agit de l'image la plus communément utilisée dans les projets cherchant à établir une communication entre un autopilote et un Raspberry Pi.

Pour une installation optimale de rpanion-server, il faut :

- Un clavier branchable en USB
- Une souris branchable en USB
- Un écran avec une entrée HDMI
- Un câble HDMI vers micro HDMI
- Un câble ethernet (facultatif)
- Un chargeur USB-C
- Une carte micro SD à flasher

La première étape consiste à flasher la carte SD. Pour cela, il faut :

- Une carte micro SD
- Un adaptateur micro SD vers USB
- Un ordinateur

Tout d'abord, installer une application dédiée pour flasher des cartes SD. Typiquement, il est possible de télécharger balena-etcher [ici](#). Ensuite, sélectionner l'emplacement de l'image à flasher sur le RPi (à télécharger au préalable), puis l'emplacement de la carte SD, et cliquer « Flash! ».

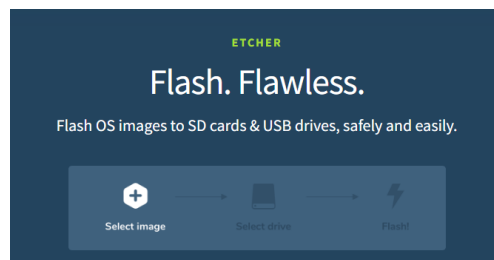


Figure 12 Balena-etcher

Une fois l'image flashée, retirer la carte SD et la replacer dans le RPi.

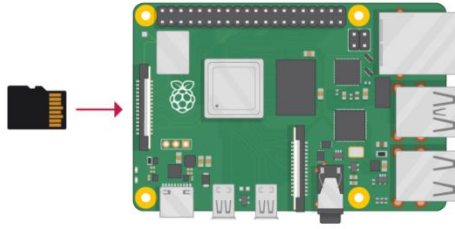


Figure 13 Insérer la carte SD

Rpanion-server dispose d'une interface graphique dédiée au pilotage d'un autopilote. En revanche, pour le reste, il est obligatoire de passer par la fenêtre de commande. Notamment, pour coder en python.

Si la communication entre le Raspberry Pi et le Pixhawk ne peut être établie en suivant les instructions préconisées sur le github de rpanion-server, il serait sans doute pertinent de flasher plutôt l'image « Raspberry Pi OS (with desktop) ». En effet, cette image dispose d'une interface utilisateur type Windows, plus agréable pour la programmation.

### 3.3.2. Connexion ssh

Pour plus de simplicité, il a été décidé de passer par la connexion à distance pour contrôler le Raspberry Pi.

La contrainte principale est d'avoir un réseau WIFI commun auquel soient connectés à la fois le Raspberry Pi et la station de bord. Typiquement, il est pratique d'utiliser le partage de connexion d'un cellulaire.

#### Étape 1 : connexion au WIFI commun

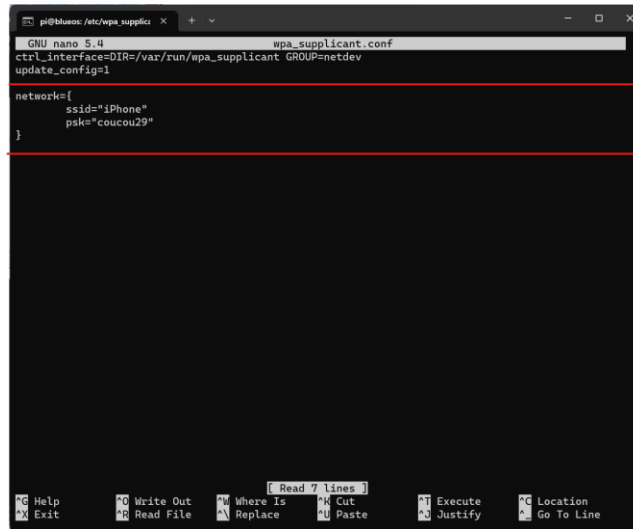
Pour la station de bord, suivre le processus habituel sous Windows pour se connecter au WIFI.

Pour le Raspberry Pi, il faut ajouter le réseau aux réseaux connus afin que la connexion se fasse par la suite de manière automatique. Dans la fenêtre de commande, exécuter les lignes suivantes :

```
« cd /etc/wpa_supplicant » #se rendre dans le bon dossier
```

```
« sudo nano wpa_supplicant.conf » #ouvrir le fichier pour modification
```

Dans le fichier qui s'ouvre, ajouter le réseau WIFI utilisé. Dans l'exemple ci-dessous, il s'agit d'un partage de connexion avec un téléphone cellulaire (iPhone) dont le mot de passe est « coucou29 ».



```
pi@blues: /etc/wpa_supplicant $ nano wpa_supplicant.conf
GNU nano 5.4 wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="iPhone"
    psk="coucou29"
}

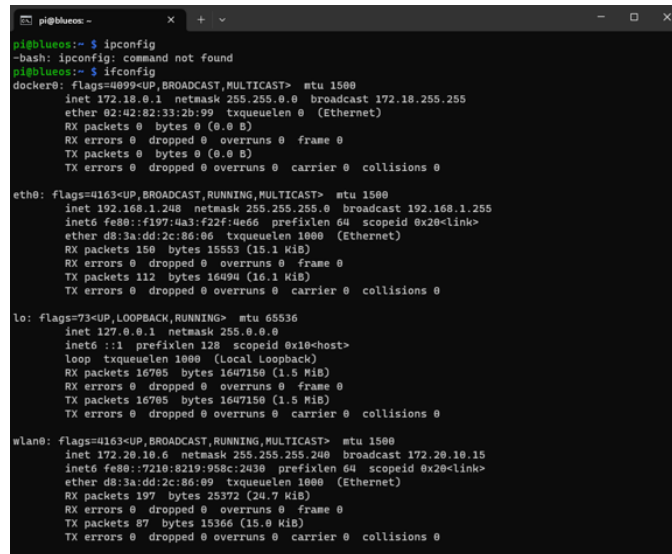
[Read 7 lines]
Help  Write Out  Where Is  Read File  Replace  Paste  Execute  Location
Exit  Read File  Replace  Paste  Justify  Go To Line
```

Figure 14 Ajouter un réseau WIFI

### Étape 2 : identifier l'adresse IP

L'adresse IP du Raspberry Pi est la plus importante à identifier. Dans la fenêtre de commande du Raspberry, taper la commande « ifconfig ». Cette commande donne l'ensemble des connexions internet du Pi.

Les connexions WIFI sont sous l'onglet « wlan0 ». L'adresse IP du Raspberry pour cette connexion est visible sous « inet ». Dans l'exemple suivant, cette adresse est 172.20.10.6.



```
pi@blues: ~ $ ifconfig
-bash: ifconfig: command not found
pi@blues: ~ $ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255
    ether 02:42:82:33:2b:99 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.248 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::f197:a3:f22f:4ed6 prefixlen 64 scopeid 0x20<link>
    ether d8:3a:dd:2c:86:06 txqueuelen 1000 (Ethernet)
    RX packets 159 bytes 15553 (15.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 112 bytes 16494 (16.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 16705 bytes 1647150 (1.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16705 bytes 1647150 (1.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.10.6 netmask 255.255.255.240 broadcast 172.20.10.15
    inet6 fe80::7210:8219:958c:2430 prefixlen 64 scopeid 0x20<link>
    ether d8:3a:dd:2c:86:09 txqueuelen 1000 (Ethernet)
    RX packets 197 bytes 25372 (24.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 87 bytes 15366 (15.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 15 Adresse IP dynamique

### Étape 3 : connexion à distance

Dès lors que les deux systèmes (Raspberry Pi et station de bord) sont connectés au même réseau, il est possible de se connecter avec la station de bord au Raspberry Pi. Pour ce faire, ouvrir une fenêtre de commande sur la station de bord, puis taper « ssh identifiant@ipadress », ou



« identifiant » désigne l'identifiant du Raspberry Pi (par défaut « pi ») et ipadress désigne son adresse ip (dans l'exemple 172.20.10.6).

En reprenant les valeurs de l'exemple, cette commande devient « ssh [pi@172.20.20.6](#) ».

Lorsqu'elle est exécutée, le mot de passe du Raspberry devrait être demandé. Il se peut que la commande prenne du temps à s'exécuter, auquel cas il est possible de :

- Interrompre par Ctrl+C et relancer la même commande
- Vérifier que la station de bord est connectée au bon réseau WIFI.
- Déconnecter et reconnecter le WIFI

### Transfert de fichiers

Il est possible de transférer des fichiers depuis le Raspberry vers la station de bord et vice versa lorsque ces derniers sont proches. En effet, le transfert passe par le réseau WIFI, qui est perdu à partir d'une dizaine de mètre de distance (dans le meilleur des cas).

- Raspberry -> Station de bord
  - Trouver l'emplacement du fichier sur le RPi : path (eg `:/home/pi/Rpanion-server/test.txt`). Ne pas oublier le nom du fichier dans l'adresse.
  - Noter la destination sur le PC (eg `C:\Users\MAHAS44\Desktop\`) : attention, ici il s'agit d'un dossier de destination, ne pas spécifier de nom de fichier. -> folder
  - Scp `pi@ipadress:path folder` (eg `"scp pi@172.20.10.6:/home/pi/ping-python/routine_sonar.py C:\Users\MAHAS44\Desktop\"`)
- Station de bord -> Raspberry
  - Trouver l'emplacement du fichier sur le PC : path (eg `C:\Users\MAHAS44\Desktop\test.txt`). Ne pas oublier le nom du fichier dans l'adresse.
  - Noter la destination sur le RPi (eg `/home/pi/`) : attention, ici il s'agit d'un dossier de destination, ne pas spécifier de nom de fichier. -> folder
  - scp `path pi@ipadress:folder` (eg `"scp C:\Users\MAHAS44\Desktop\2023-09-29_routine_sonar.py pi@172.20.10.6:/home/pi/ping-python/"`)

### 3.3.3. Installer les librairies python nécessaires

Le Rpi va fonctionner avec un programme codé en Python. Pour qu'il puisse fonctionner, un certain nombre de librairies doivent être installées. Elles sont déjà installées sur le Rpi du drone actuel.

Pour les installer sur un nouveau Rpi, exécuter les commandes suivantes dans l'ordre :

- `sudo apt-get install python3-pip python3-matplotlib python3-serial`  
Installation des librairies python avec les droits administrateur (sudo)
- `cd /home/pi/`  
Se rend dans le dossier /home/pi/
- `git clone https://github.com/sunfounder/SunFounder\_SensorKit\_for\_RPi2`

- Installation de la librairie python pour l'usage des capteurs SunFounder
- `pip install --user bluerobotics-ping --upgrade`
- Installation de la librairie python pour l'utilisation du sonar
- `git clone --single-branch --branch deployment https://github.com/bluerobotics/ping-python.git`
- Également pour l'utilisation du sonar
- `cd ping-python`
- Repositionnement de dossier
- `python setup.py install --user`

Faire un essai de connexion du sonar avec les commandes :

- `cd ping-python`
- `python`  
Ouvre l'application Python dans la fenêtre de commande. Des lors, il est possible et nécessaire d'utiliser le langage Python
- `from brping import Ping1D`  
Importer la librairie du sonar
- `myPing = Ping1D`  
Créer un objet Ping
- `myPing.connect_serial("/dev/ttyUSB0", 115200)`  
Ouvrir le port série et connecter le Ping.  
*Note: cette commande risque de retourner "[Errno 16] could not open port /dev/ttyUSB0: [Errno 16] Device or resource busy: '/dev/ttyUSB0'". Dans ce cas, débrancher et rebrancher le sonar. Cela devrait résoudre le problème. Si ce n'est pas le cas, essayer de brancher le sonar à l'ordinateur portable de la station de bord (sur un port USB), puis le rebrancher sur le Rpi.*

#### 3.3.4. Intégration du GPS

Le GPS est branché en USB. Il s'agit d'une communication série, il faut donc activer ce type de communication sur le Rpi si ce n'est pas déjà fait, avec la commande « `sudo raspi-config` ».

Pour vérifier que le GPS est bien branché est reconnu :

- « `lsusb` » : liste les connexions usb, chercher « U-Blox AG »
- « `dmesg | grep tty` » : liste les connexions série, chercher « USB ACM Device ». Le nom du port associé au GPS est donné ici. Dans la suite, il est noté x.
- « `cat /dev/ttyx` » : cette commande ouvre le port série sur lequel se trouve le GPS et renvoie les données qu'il acquiert. Il utilise le format NMEA et capte une diversité de messages parmi GP GGA, GP RMC, GP VTG, GP GSV... Un exemple de message est : « `$GPGGA,181908.00,3404.7041778,N,07044.3966270,W,4,13,1.00,495.144,M,29.200,M,0.10,0000*40` », il se divise en différents champs présentés Tableau 2.

Tableau 2 Champs d'un message GP GGA

| Identifiant message |                                          | Champ                  |                              |                          |                               |                                  |                                   |                    |              |                       |                      |                           |                        |                          |                           |
|---------------------|------------------------------------------|------------------------|------------------------------|--------------------------|-------------------------------|----------------------------------|-----------------------------------|--------------------|--------------|-----------------------|----------------------|---------------------------|------------------------|--------------------------|---------------------------|
| GP – GPSONly        | GGA - Global Positioning System Fix Data | UTC (hhmmss.ss)<br>Eg: | Latitude (III.IIIIII)<br>Eg: | Hemisp here (N/S)<br>Eg: | Longitude (III.IIIIII)<br>Eg: | Hemisp here of Long (E/W)<br>Eg: | GPS quality indicator (0-5)<br>Eg | Nb satellites (XX) | HD OP (XX.X) | Altitude (geoid X.XX) | Unit of altitude (M) | Geoidal separation (X.XX) | Unit of geoid. Sep (M) | Age of differential data | Difference ref station ID |
| GP                  | GPGGA                                    | 181908.00              | 3404.7041778                 | N                        | 07044.3966270                 | W                                | 4                                 | 13                 | 1.00         | 495.144               | M                    | 29.200                    | M                      | 0.10                     | 0000*40                   |

## 4. Deuxième partie : premier lancement

### 4.1. Branchements

L'intégration du sonar a été un enjeu important du projet. Plusieurs options de connexion avec le Pixhawk ont été considérées. Finalement, le système retenu est composé de deux modules complètement indépendants : il n'y a aucune connexion, ni filaire ni à distance, entre le Pixhawk et toutes les composantes qui lui sont attachées, et le module constitué de la sonde, du Raspberry Pi et de toutes les composantes qui leur sont reliées.

Cette section est donc composée de deux sous sections : la première présente les branchements au sein du module « Raspberry Pi » (sonar, sonde de température, GPS USB, écran LCD et Power Bank), et la seconde présente ceux au sein du module « Pixhawk » (Pixhawk et toutes ses connexions).

#### 4.1.1.1. Raspberry Pi

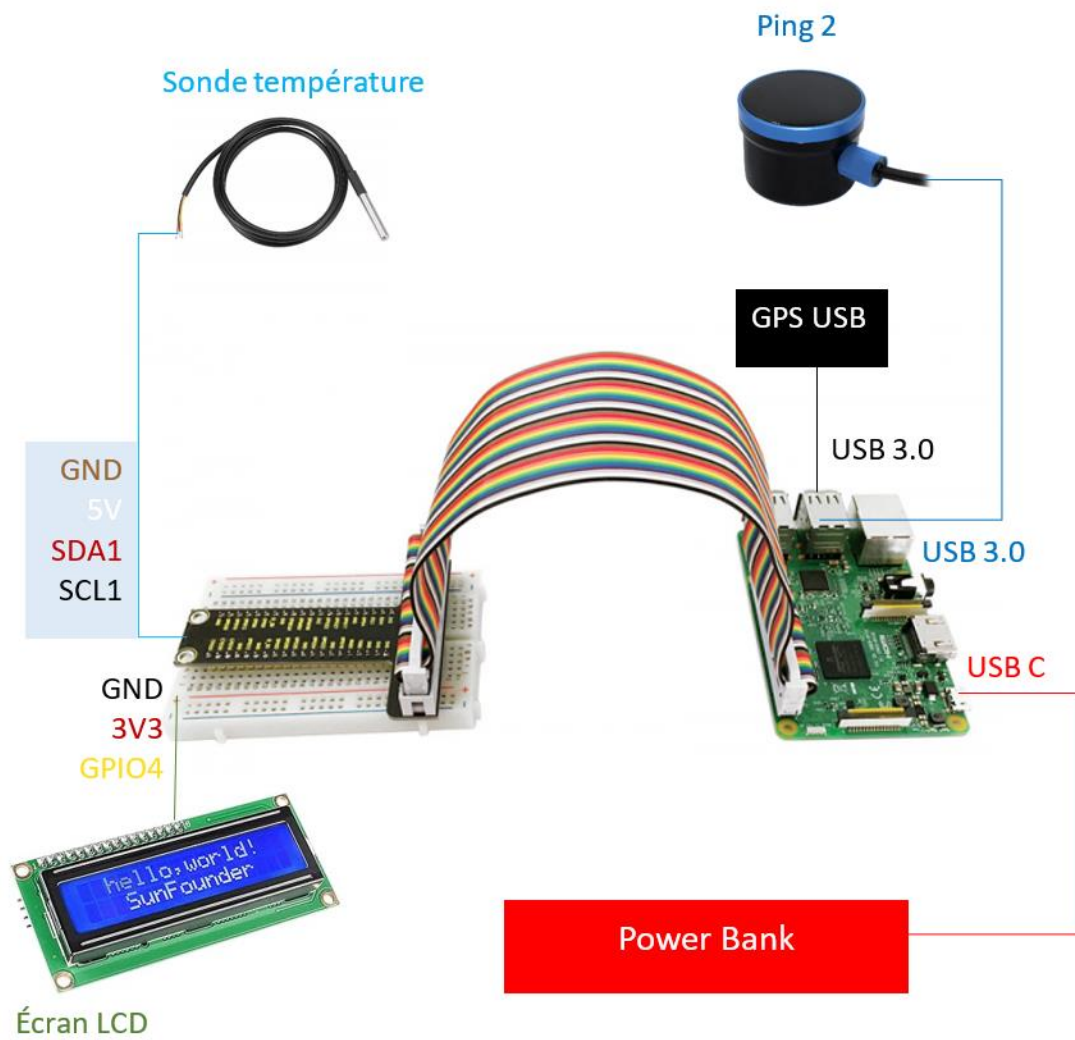


Figure 16 Branchement pour le système Raspberry Pi

#### 4.1.2. Pixhawk

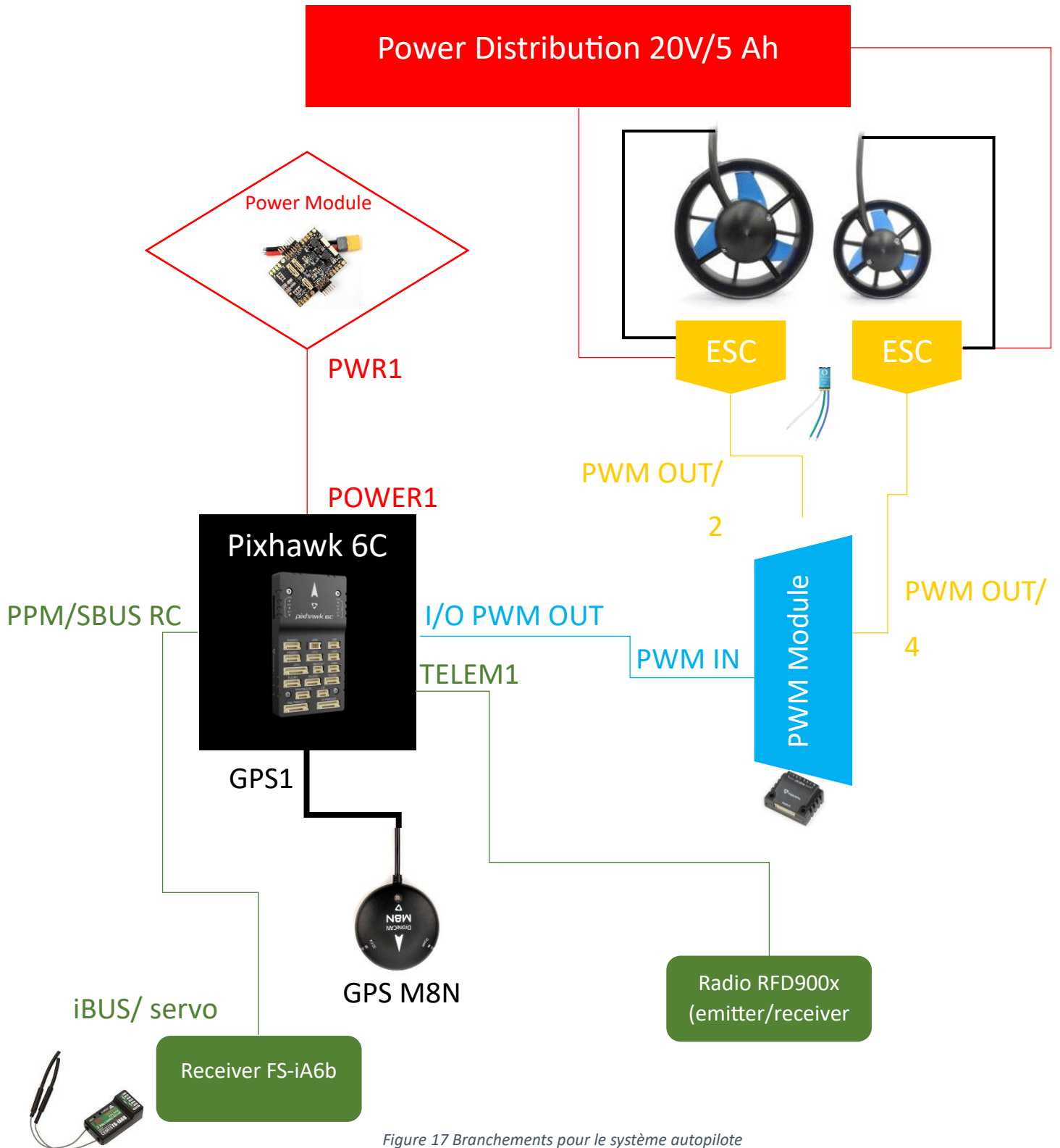


Figure 17 Branchements pour le système autopilote

## 4.2. Initialisation (ESC)

Lorsque l'alimentation est connectée, et si le Pixhawk est branché au GPS, ce dernier émet une série de « bip ». Un « bip » des ESC vient ensuite confirmer que ces derniers reçoivent du courant. Un dernier « bip » marque le fait qu'ils sont initialisés. Si le second « bip » est plus grave, cela indique l'échec de l'initialisation.

## 4.3. Calibration onshore

Une partie de la calibration peut être faite depuis un espace extérieur dégagé (pour le signal GPS) quelconque. Pour cette première partie de calibration, l'accès à un point d'eau de type piscine ou lac n'est pas nécessaire.

### 4.3.1. Orientation Pixhawk (AHRS\_ORIENTATION)

Un premier paramètre doit être défini qui indique la configuration globale (position du Pixhawk). Si ce paramètre est mal ajusté, la calibration risque d'échouer.

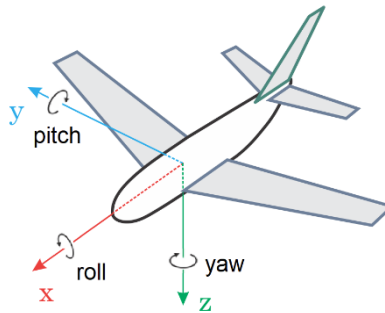


Figure 18 Référentiel (axes)

D'après le dessin ci-dessus :

- Roulis positif = inclinaison vers la droite
- Tangage positif = nez vers le haut
- Yaw positif = nez tourne dans le sens horaire

Plus d'informations peuvent être trouvées sur :

[https://docs.px4.io/main/en/config/flight\\_controller\\_orientation.html](https://docs.px4.io/main/en/config/flight_controller_orientation.html)

Comme l'illustre la Figure 19, le choix d'orientation du Pixhawk qui a été fait n'est pas le choix classique (ie à plat et pointant vers l'avant du drone). Il a donc été nécessaire d'ajuster le paramètre AHRS\_ORIENTATION mentionné précédemment. Le Pixhawk est placé avec le nez pointant vers la gauche, et pivoté de 90° de sorte que sa face ne soit plus orientée vers le haut mais vers l'avant.



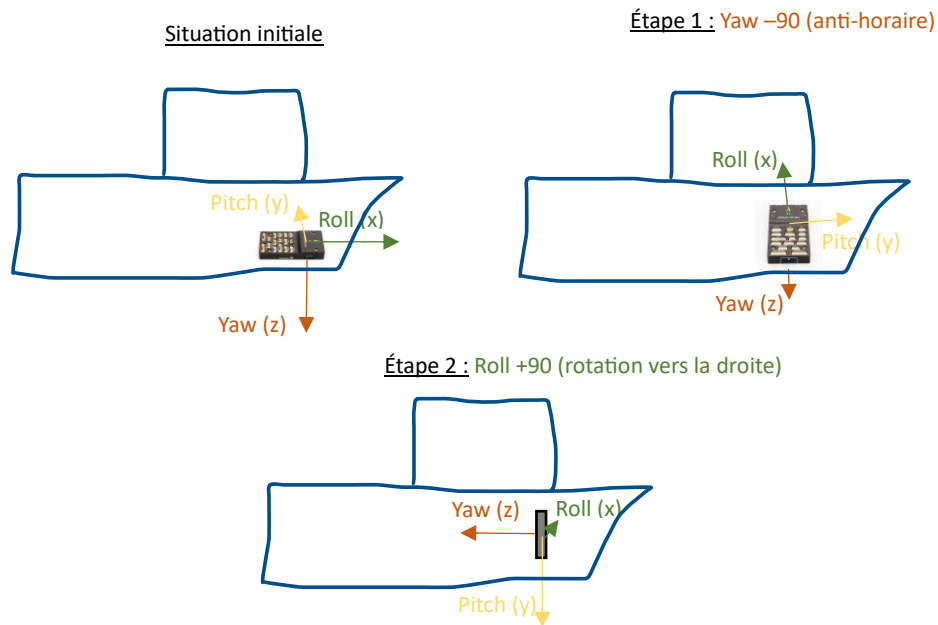


Figure 19 AHRS\_ORIENTATION pour la plateforme finale

#### 4.3.2. Calibration accéléromètre

Pour calibrer l'accéléromètre, il faut placer le bateau sur tous ses côtés. Pour plus de simplicité, la calibration peut être faite en séparant la boîte orange de la structure qui porte les moteurs. La calibration est alors faite avec la boîte seule, ce qui donne des résultats satisfaisants.

Suivre les instructions de MP. Ne pas oublier de « Calibrate Level » (Faire le niveau) après avoir calibré l'accéléromètre. Ce niveau pourra être refait après la calibration GPS et/ou lorsque la plateforme est complètement montée.

Il est important de maintenir le drone le plus fixement possible dans chacune des six positions demandées. En revanche, une imprécision sur l'angle d'inclinaison n'est pas rédhibitoire (jusqu'à 20 deg).

Simple accel calibration : si on veut absolument s'éviter les rotations, ou qu'elles ne donnent pas un résultat satisfaisant, simplement poser le véhicule sur un sol plat et cliquer « simple accel calibration ». La calibration est rapide et ne demande pas de mouvement de la plateforme.

#### 4.3.3. Calibration GPS

Comme dans la section précédente, la calibration peut être faite en séparant la boîte orange de la structure des moteurs.

Suivre les instructions de MP. L'idée est de faire une rotation complète de la boîte selon chaque axe, et ce dans chacun des deux sens de rotation possibles. Essayer de garder le compte des rotations effectuées permet de rendre la calibration plus efficace. Sinon, il y a risque de perdre le fil et de répéter certaines rotations plusieurs fois avant de compléter le processus.

Large vehicle Mag Calibration : comme pour les accéléromètres, il est possible de calibrer le GPS de manière simplifiée. Simplement poser le navire sur un sol plat, et l'orienter vers le nord. Cliquer « Large vehicle Mag. Calibration » et indiquer 0 (angle entre Nord et avant du véhicule). Attention aux interférences magnétiques lors de l'orientation au Nord de la plateforme.

#### 4.3.4. Calibration Radio

Dans MP, sous « SETUP », « Mandatory Hardware » puis « Radio » commencer la calibration. Bouger les gimbal de la télécommande dans leurs positions extrêmes.

Pour l'ajout des fonctions auxiliaires, chercher SERVOxx\_FUNCTION. [Détail](#).

#### 4.3.5. Flight Modes

Trois modes de pilotage sont fréquemment utilisés (« manual », « acro » pour la calibration et « auto ») et assignés à un switch de la télécommande ayant trois positions. La configuration suivante est adoptée pour que chaque position du switch corresponde à un mode différent.

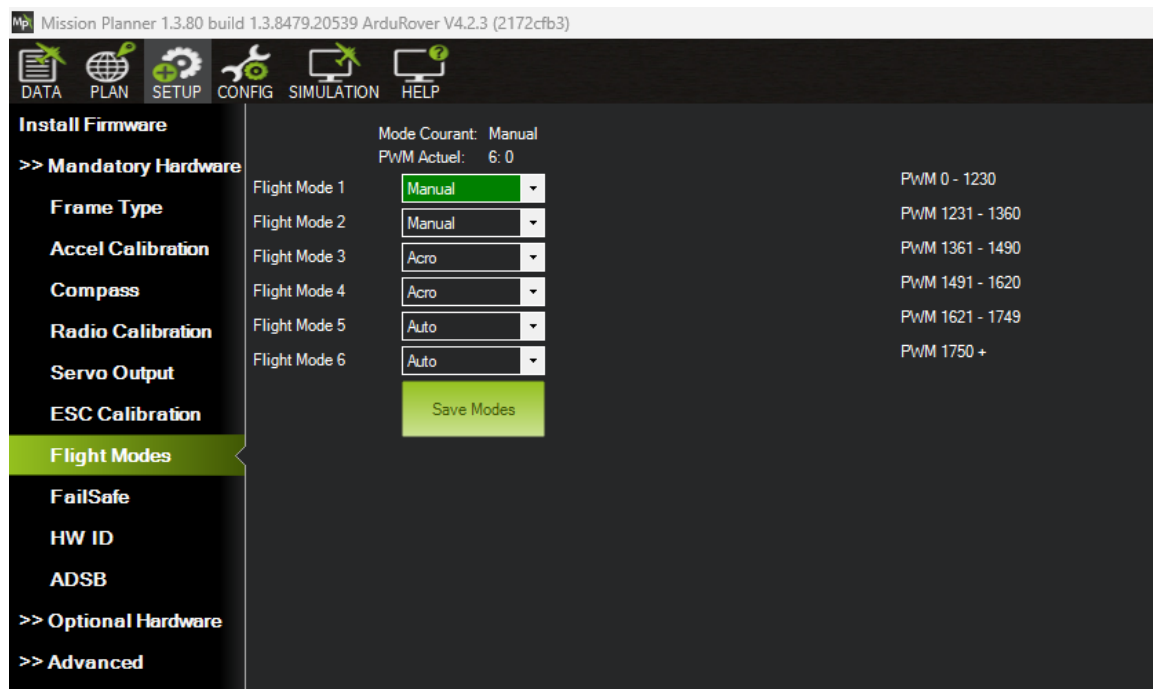


Figure 20 Setup des modes de pilotage

### 4.4. Calibration offshore

La deuxième partie de la calibration a obligatoirement lieu sur une étendue d'eau. Cela étant, une piscine de taille raisonnable peut suffire.

#### 4.4.1. Calibration vitesse

Pour être précis dans ses missions, le Pixhawk a besoin de savoir quelle puissance des moteurs est associé à quelle vitesse. C'est l'objet de la fonction « Learn Cruise Throttle » programmé sur l'interrupteur gauche de la télécommande (SWA).

Pour l'apprentissage, conduire le bateau en ligne droite et activer l'interrupteur SWA. Le but est d'être entre 50% et 80% de la puissance maximum.

Lorsque l'apprentissage réussi, un message s'affiche « Learned Cruise Throttle ». Dans MP, dans l'onglet « Messages », il est également possible de voir les valeurs sauvegardées pour les paramètres CRUISE\_THROTTLE et CRUISE\_SPEED. Ces deux paramètres sont utilisés en mode auto pour le réglage de la vitesse.

La suite de la calibration consiste à régler le contrôleur PID. Pour cela, il faut comparer la valeur réelle du « throttle » (paramètre pid\_achieved) avec la commande (pid\_desired). Si les deux valeurs se suivent lors des accélérations du drone, le contrôleur est bien réglé.

De manière générale, les étapes à suivre pour un réglage optimal sont :

- Dans « CONFIG », « Full Parameter List » choisir "GCS\_PID\_MASK" et cocher « Throttle ». « pid\_achieved » et « pid\_desired » afficheront alors une valeur en lien avec la vitesse.
- Se mettre en mode acro.
- S'il y a un long temps de latence avant que la valeur réelle du « throttle » n'atteigne la valeur commandée, augmenter P.
- Si le véhicule a un déplacement saccadé, réduire P.
- Si la vitesse à long terme oscille, diminuer I.
- Si la vitesse à long terme réelle s'écarte de la vitesse voulue, augmenter I.
- I doit rester inférieur à P.
- D peut être laissé à 0.

Pour un réglage plus précis, plus d'informations sont disponibles au lien suivant : <https://ardupilot.org/rover/docs/rover-tuning-throttle-and-speed.html>.

#### 4.4.2. Calibration des virages

La calibration des virages est très similaire à la calibration de la vitesse. En effet, elle passe elle aussi par le réglage d'un contrôleur PID en comparant la valeur réelle obtenue pour le « steering » avec la valeur commandée.

La première étape (et la plus importante) est cependant différente puisqu'il faut définir manuellement la valeur du paramètre ACRO\_TURN\_RATE (« CONFIG » -> « Full Parameter List ») : rentrer une valeur approximative en degrés par secondes.

Vient ensuite le réglage du contrôleur PID. Suivre les étapes suivantes :

- GCS\_PID\_MASK = 1
- Régler le paramètre FF (ATC\_SPEED\_STR\_FF) : augmenter pour plus de réactivité.
- Régler P et I de la même manière que pour la vitesse.
- I et P doivent être très inférieurs à FF (environ 20% de FF)
- D peut être laissé à 0.

Pour un réglage plus précis, plus d'informations sont disponibles au lien suivant : <https://ardupilot.org/rover/docs/rover-tuning-steering-rate.html>.

## 5. Troisième partie : Mission autopilote

### 5.1. Créer une mission

Ouvrir MP. Se rendre dans L'onglet « PLAN ». Un clic gauche n'importe où sur la carte ajoute un WP à cet endroit. Il est possible de définir une mission ainsi en créant les WPs les uns après les autres. Lorsque l'objet de la mission est une zone donnée de la carte et que l'on veut automatiser la création des WPs, suivre la démarche suivante :

- Clic droit puis « Dessiner Polygone » -> « Draw a Polygon »
- Dessiner le polygone par des clics gauches successifs sur la carte
- Clic droit puis « Auto WP » -> « Simple Grid »
- Dans la fenêtre qui apparaît, changer selon vos préférences :
  - o Angle : l'inclinaison des lignes
  - o Line spacing : distance entre deux lignes consécutives
  - o Spacing inline : distance entre deux WPs consécutifs sur une même ligne
  - o Speed (est) : dans notre cas, la vitesse sera de 0.4m/s en général; indiquer 1 m/s.
- Lorsque la mission correspond aux attentes, cliquer « Accept »
- Il est encore possible de rajouter des WPs (par exemple pour effectuer des lignes croisées) :
  - o Clic droit « Dessiner Polygone » -> « Clear Polygon »
  - o Clics gauches sur la carte aux endroits souhaités
  - o Par défaut, les WP dessinés seront les derniers de la liste. Cela étant, l'ordre peut être changer en utilisant les commandes Haut (↑) et Bas (↓)

|   | Commande        | Delay |   |   |   | Lat        | Long        | Alt | Frame    | Suprim | Haut | Bas | Grad  | Angle | Dist  | AZ  |
|---|-----------------|-------|---|---|---|------------|-------------|-----|----------|--------|------|-----|-------|-------|-------|-----|
| 1 | DO_CHANGE_SPEED | 1     | 5 | 0 | 0 | 0          | 0           | 0   | Relative | X      | ↑    | ↓   | 0     | 0     | 0     | 0   |
| 2 | WAYPOINT        | 0     | 0 | 0 | 0 | 46.9271338 | -71.4057706 | 100 | Relative | X      | ↑    | ↓   | 145.0 | 55.4  | 121.5 | 14  |
| 3 | WAYPOINT        | 0     | 0 | 0 | 0 | 46.9271247 | -71.4059245 | 100 | Relative | X      | ↑    | ↓   | 0.0   | 0.0   | 11.7  | 265 |
| 4 | WAYPOINT        | 0     | 0 | 0 | 0 | 46.9270861 | -71.4065786 | 100 | Relative | X      | ↑    | ↓   | 0.0   | 0.0   | 49.9  | 265 |

Figure 21 Réorganiser les WP

### Paramètres supplémentaires

En plus de la démarche explicitée précédemment, quelques paramètres supplémentaires doivent être réglés. Ils définissent la vitesse de croisière du drone et sa précision :

- WP\_SPEED : Vitesse par défaut en mode autopilote. Pour la plateforme finale, la vitesse est 0.4m/s. cela correspond à 70% de la puissance max.
- WP\_RADIUS : définition de cercles autour des WP. Lorsque le bateau rentre dans ce cercle, il a atteint son WP, et part en direction du suivant. Plus le rayon est faible, plus la mission sera précise.

### 5.2. Lancer la mission

Dans MP, sous « PLAN », faire « Load File » (à droite de la fenêtre) et sélectionner le fichier .Waypoint correspondant à la mission.

Puis, « Écrire PN ». Une barre de chargement d'affiche avec les numéros successifs des WP en cours d'écriture.



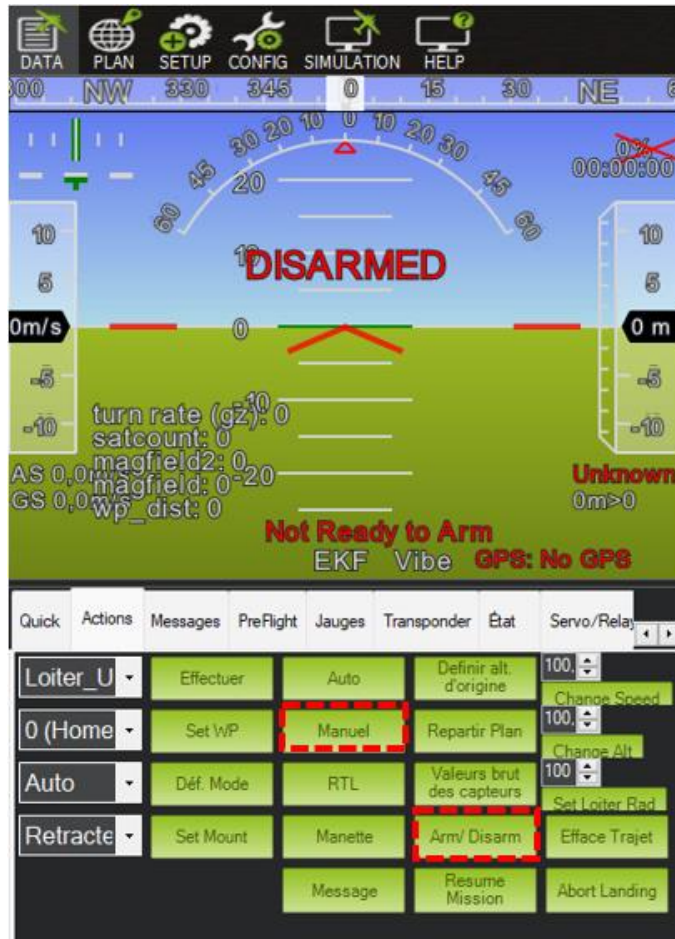


Figure 23 Démarrage de la mission avec MP

### 5.3. Interrompre une mission en cours

Si, pour une raison quelconque, une mission ne se passe pas bien, il est possible de l'interrompre et de ramener le bateau au point de départ.

*Avec la télécommande*

Repasser en mode « Manuel », puis piloter le bateau afin qu'il revienne vers l'endroit souhaité.

*Sans télécommande*

Dans MP, sous « DATA », faire Clic-droit à l'endroit où le drone doit être ramené puis « Naviguez ici ».

### 5.4. Acquisition de la profondeur

L'acquisition du sonar est gérée par un programme Python. Il y a deux manières de le lancer.

Dès lors que le branchement de la Figure 16 est établi, l'acquisition se déclenche : la première manière de lancer une acquisition est d'alimenter le Rpi. En effet, le programme python écrit pour l'acquisition des données se lance automatiquement au démarrage. Cette méthode est théoriquement très intéressante mais présente un inconvénient important par rapport à la seconde méthode, qui est qu'il y a moins de suivi de l'acquisition (la seule manière de suivre



l'écriture des données de profondeur est de télécharger le log dès que le drone est à portée). En particulier, au démarrage, on n'a pas accès à l'affichage des profondeurs sur la station de bord. Et si le Rpi renvoie une erreur au démarrage notamment, ce qui est fréquent, il est dur d'en trouver la cause et de la résoudre.

La seconde manière de lancer une acquisition est par la fenêtre de commande, en connexion ssh, comme illustré dans la Figure 24. Les ronds indiquent les étapes de la démarche, et les flèches donnent les commandes à entrer dans la fenêtre de commande.

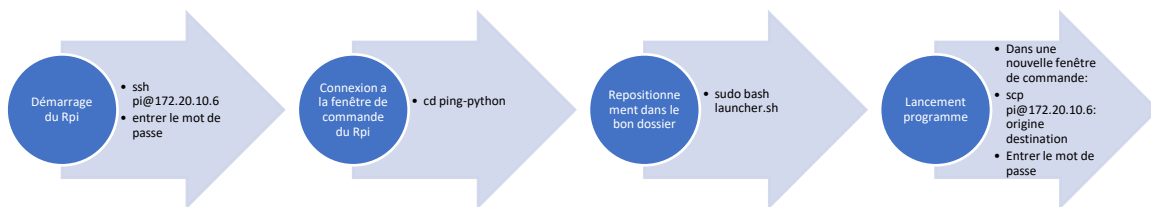


Figure 24 Démarrage d'une acquisition par fenêtre de commande

Au début, et tant que le drone est à portée de WIFI (10m environ), la valeur de la profondeur s'affiche en temps réel dans l'écran connecté. Ensuite, la connexion entre la station de bord et le Rpi est perdue. La connexion peut être rétablie dès que le sonar se rapproche, mais les messages indiquant la profondeur ne seront pas générés sur l'écran connecté.

Pour le suivi de la mission, la méthode retenue est de télécharger le log dès que le drone est à portée, de l'ouvrir, et de vérifier que la dernière date enregistrée porte l'heure actuelle. Pour télécharger un fichier log (dans l'exemple, vers le dossier où sont enregistrés les logs de profondeur – selon la Figure 11):

```
« scp pi@172.20.10.6:/home/pi/ping-python/nomdufichier.txt  
C:\Users\MAHAS44\Desktop\Acquisitions\date_lieu\Depth\ »
```

Pour télécharger l'ensemble des fichiers logs (dans l'exemple, vers le dossier où sont enregistrés les logs de profondeur – selon la Figure 11):

```
« scp pi@172.20.10.6:/home/pi/ping-python/*.txt  
C:\Users\MAHAS44\Desktop\Acquisitions\date_lieu\Depth\ »
```

Dans l'idéal, image de toutes ces commandes

## 6. Quatrième partie : Après la mission

### 6.1. Télécharger les logs

Dans MP, aller dans « DATA » puis « Dataflash logs », puis « Download DataFlash Log Via Mavlink ». Sélectionner les logs que vous souhaitez télécharger et cliquer sur « Charger ces journaux ». La fenêtre suivante montre ce qui s'affiche.



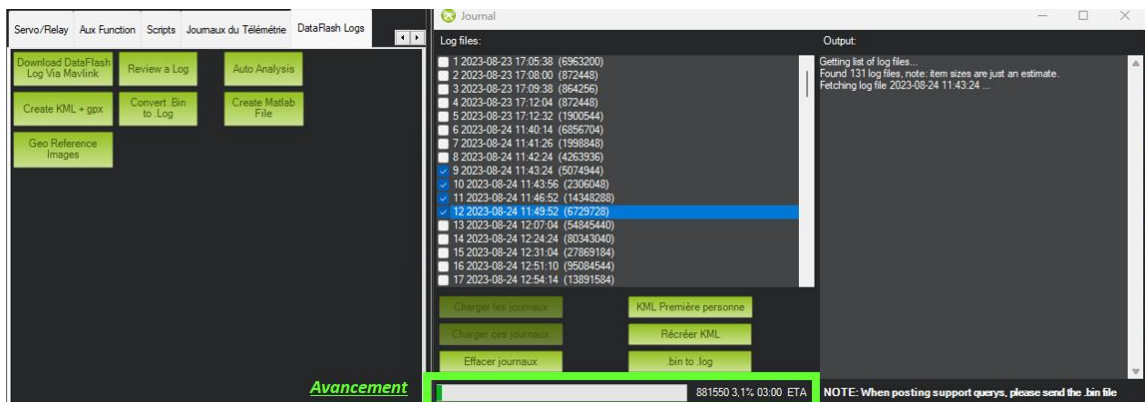


Figure 25 Téléchargement des logs.

## 6.2. Analyse des logs

Dans MP, aller dans « DATA », puis « Dataflash logs », puis « Review a log ».

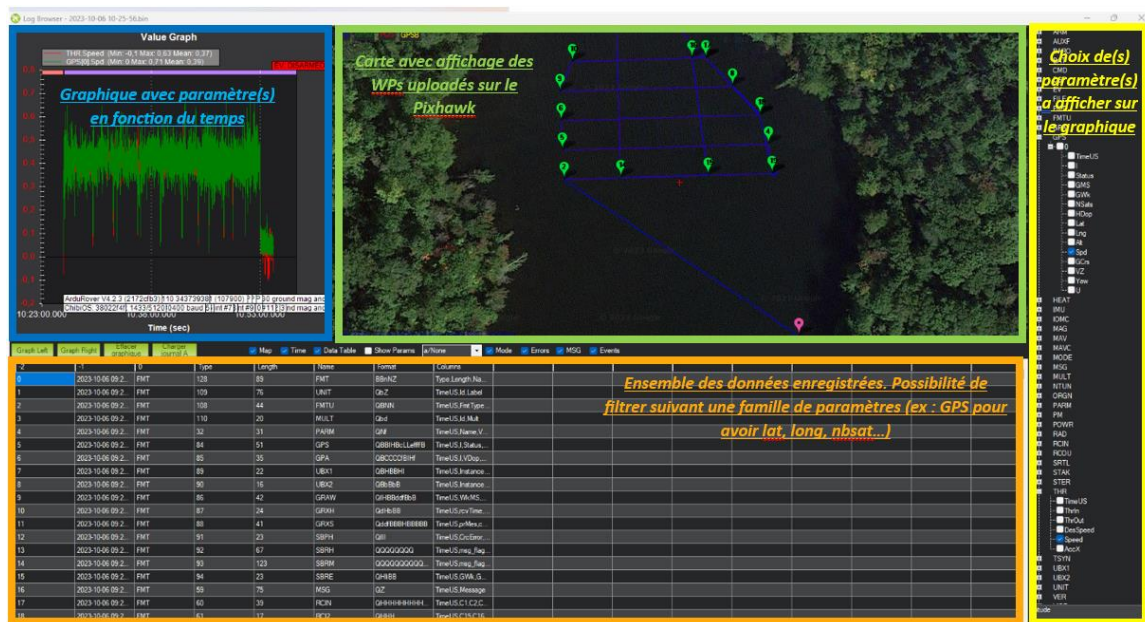


Figure 26 "Review a log"

Pour filtrer, cliquer sur l'en-tête du tableau et trouver le nom de la famille de paramètres dans la liste (eg : GPS, IMU...). Il est possible de télécharger l'ensemble des données visibles dans le tableau (ie les paramètres appartenant à la famille sélectionnée) en faisant un clic droit sur le tableau, puis « export visible ».

## 6.3. Post traitement de la profondeur avec Python

Pour résumer le type de données traitées et leur origine, un diagramme représentant le traitement l'information à partir de son acquisition est donné Figure 27. Il montre quelles sont les informations utilisées pour obtenir le résultat final de l'acquisition (ie table de données avec coordonnées et profondeur et carte).

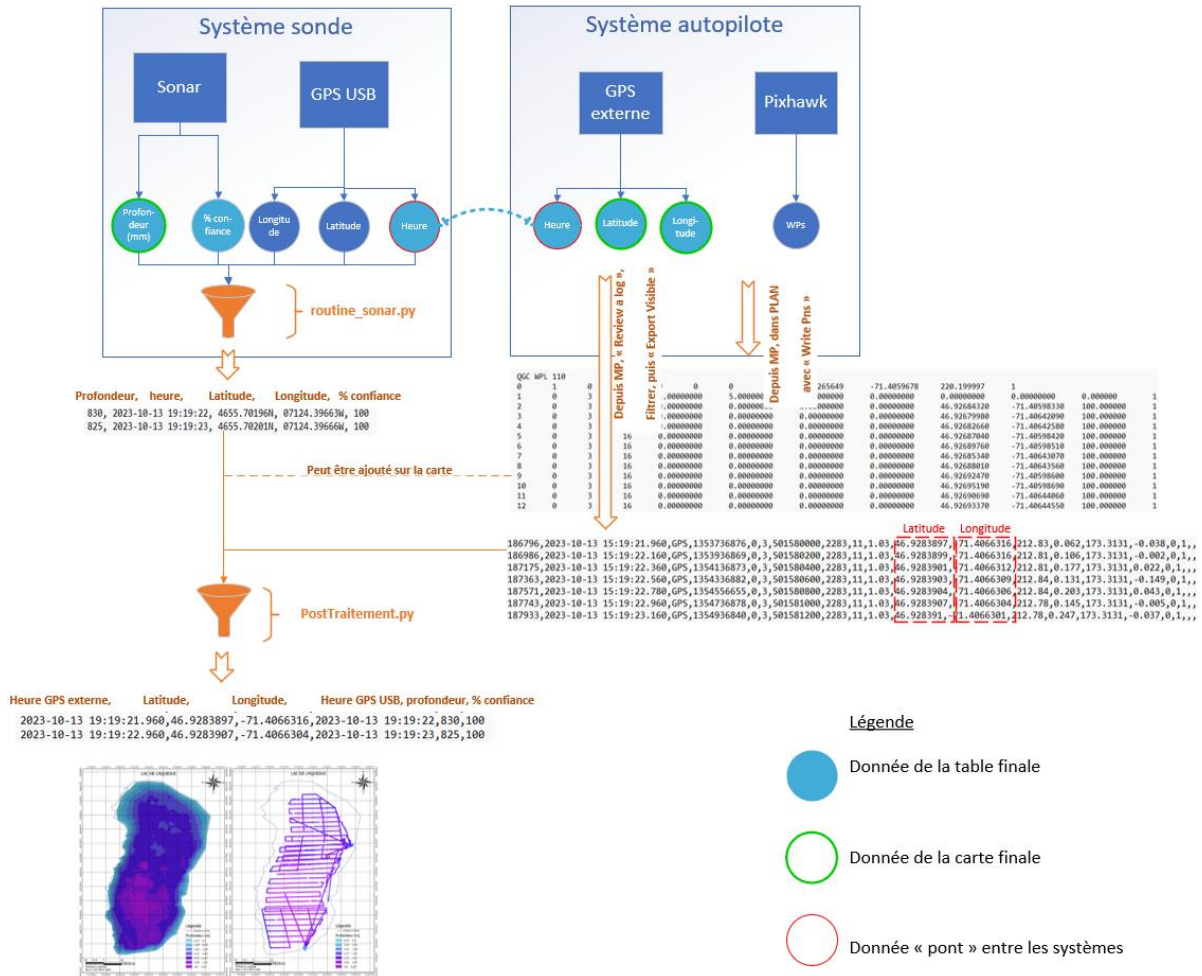


Figure 27 Diagramme de traitement de l'information.

Au niveau de la sonde, c'est seulement la profondeur qui est utilisée dans la carte finale. L'heure et le pourcentage de confiance sont gardés dans le fichier csv final. Dans un premier temps, la profondeur est notée dans un fichier texte, accompagnée de l'heure, des coordonnées GPS, et du degré de confiance. Cette opération est réalisée à l'aide d'un programme Python « routine\_sonar.py ».

Ensuite, les données du GPS externe (système autopilote) doivent être téléchargées, en respectant la procédure décrite dans Analyse des logs.

L'heure GPS permet alors de faire le pont entre les deux systèmes. Les coordonnées du système autopilote sont récupérées et associées aux données de profondeur de la sonde. Le tout grâce au programme Python appelé « PostTraitement.py ». Ce programme produit un tableau au format csv avec six colonnes qui sont l'heure du GPS externe, les coordonnées du GPS externe (latitude puis longitude), l'heure GPS USB, la profondeur en mm et le pourcentage de confiance dans la donnée de profondeur.

Pour le tracé de la carte, la procédure est expliquée dans Représentation de la donnée.

## 6.4. Représentation de la donnée

Les données sont représentées dans un environnement SIG. Dans un premier temps, les coordonnées du GPS doivent être converties en distances, afin d'obtenir un nuage de point métrique. Il est alors finalement possible de visualiser les points de mesure tels qu'ils ont été calculés, mais également de les extrapoler afin d'obtenir une carte bathymétrique (cf Figure 28).

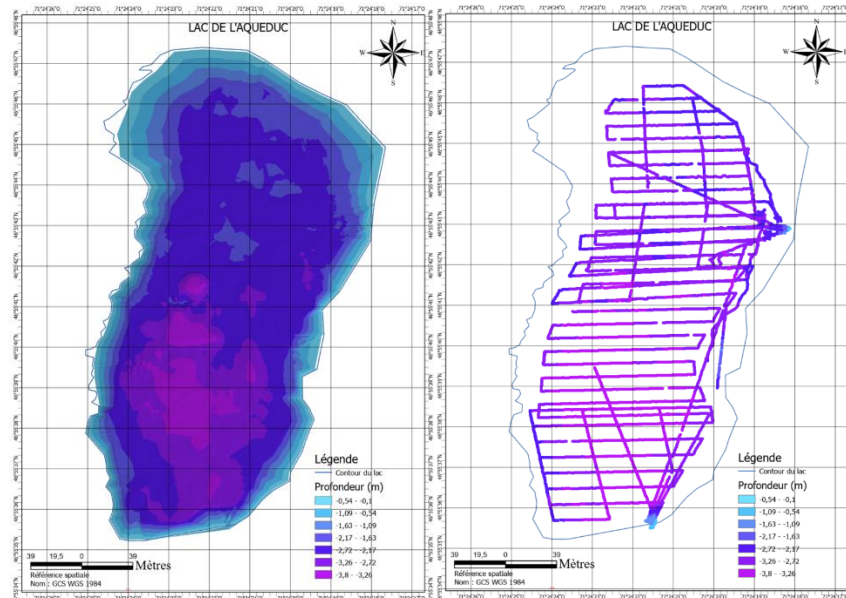


Figure 28 Carte bathymétrique

## 7. Cinquième partie : Troubleshooting

### 7.1. Pilote automatique

« En cours de route, l'axe de mon Pixhawk pivote de 180° (il se met alors à s'éloigner du prochain WP au lieu de s'en rapprocher) » : il s'agit sans doute d'un problème d'orientation de GPS et/ou du Pixhawk. Vérifier si le paramètre « COMPASS\_AUTO\_ROT » est activé. Si c'est le cas, alors l'erreur est peut-être due à une incohérence dans l'orientation des compass : la direction du compass externe est alors automatiquement changée et alignée sur le compass interne.

« J'ai un message d'erreur « Compass inconsistent » » : signifie que le compass interne au pixhawk ne pointe pas vers la même direction que le compass externe (GPS M8N). Vérifier que les paramètres AHRS\_ORIENTATION et COMPASS\_ORIENTATION correspondent à la réalité.

« La position affichée dans MP pour le navire varie lorsque la station de bord se déplace » : cela arrive typiquement lorsque les antennes d'une des radios sont légèrement dévissées. Les revisser devrait résoudre le problème.

« J'ai un message d'erreur « AHRS\_UNHEALTHY » » : Il arrive que ce message apparaisse et disparaisse par lui-même. Il est possible de l'ignorer sans pour autant porter préjudice à la mission.

« En mode manuel, le bateau tourne dans le sens opposé. Si j'essaye d'aller à gauche avec la télécommande, il tourne vers la droite. » : il est probable qu'il y ait une erreur dans le

branchement des moteurs. Vérifier qu'un fil n'ait pas été inversé! (Branchement ESC vers les moteurs doivent respecter l'alignement des couleurs).

« Lors du tuning des virages, le `pid_achieved` est très loin du `pid_desired`, quelques soient les valeurs de P, I et D. » : quelle valeur a été retenue pour `ACRO_TURN_RATE`? Essayer de modifier cette valeur pour qu'elle colle davantage avec la réalité. Cela pourrait résoudre le problème. A titre d'exemple, la première plateforme développée (boogie board) avait un `ACRO_TURN_RATE` de 180 deg/s tandis que ce paramètre est à 35 deg/s pour la plateforme finale (moins réactive).

## 7.2. Raspberry Pi

« Le code retourne une erreur avec un message contenant « /dev/ttyUSB0 resource busy » » : le dossier « /dev/ » du Rpi contient les appareils qui lui sont connectés. Dans le setup actuel, « `ttyUSB0` » est le port série associée à la sonde. Le problème retourné est donc un problème avec la sonde. Pour le résoudre deux pistes sont possibles : manuellement débrancher et rebrancher la sonde, ou la brancher momentanément au laptop de la station de bord pour l'initialiser grâce à ping-viewer.

« J'ai des trous dans mes données » : la manière dont le code python d'acquisition (`routine_sonar.py`) fonctionne est qu'il acquiert la donnée du GPS en premier. Quand il capte un message GPS GPGGA, il va chercher la profondeur du sonar, et il écrit une ligne dans le fichier log. S'il y a des trous dans les données, ça veut sans doute dire qu'aucun message GPGGA n'est arrivé OU que le Rpi a du mal à gérer toutes les données arrivant.