



École Polytechnique de l'Université de Tours

64, Avenue Jean Portalis
37200 TOURS, France
Tél. +33 (0)2 47 36 14 14

www.polytech.univ-tours.fr

Département Informatique industrielle

Analyse et modélisation			
Projet :	Système de stabilisation motorisé pour caméra		
Réalisé par :	Sébastien LAPORTE	Encadré par :	Ameur SOUKHAL
Début du projet :	22/09/2015	Fin du projet :	10/02/2015
Historique des modifications			
Version	Date	Note de version	
1.0	04/01/2016	Version de dépôt initial	

Table des matières

1.	Introduction.....	4
2.	Contexte et objectif du projet	4
3.	Architecture générale du système	4
4.	Partie électronique	5
4.1.	Choix des technologies	5
4.1.1.	Capteur	5
4.1.2.	Motorisation	6
4.1.3.	Microcontrôleur	8
4.1.4.	Alimentation	8
4.2.	Schémas fonctionnels	9
4.2.1.	Fonction « Alimentation »	9
4.2.2.	Fonction « Traitement des données »	9
4.2.3.	Fonction « Captage »	10
4.2.4.	Fonction « Commande moteur »	10
4.2.5.	Schéma fonctionnel du système	10
5.	Partie mécanique	12
5.1.	Schéma du système	12
5.2.	Modélisation de la structure	13
5.2.1.	Schémas complets	13
5.2.2.	Schémas des pièces	15
5.3.	Conception des éléments de la structure	16
5.4.	Conception des éléments de protection	17
5.5.	Choix techniques pour la réalisation	18
5.5.1.	Matériaux utilisés pour la structure	18
5.5.2.	Technique de réalisation des éléments de protection	18
4.4.2.	Fixation de la caméra	18
6.	Partie logicielle	19
6.1.	Exploitation des données d'un capteur IMU 6 DoF	19
6.2.	Exploitation des données d'un capteur IMU 9 DoF	20
6.3.	Technique de commande d'un moteur BLDC.....	21
6.4.	Modélisation des fonctions.....	22
6.4.1.	Configuration du capteur	22
6.4.2.	Acquisition des données	23
6.4.3.	Calibration du capteur	24
6.4.4.	Traitement des données	25
6.4.5.	Correction de l'orientation	25
6.4.6.	Asservissement en position	25
6.4.7.	Commande de la motorisation	25
6.5.	Processus / fonctionnement	25
6.5.1.	Initialisation	26

6.5.2.	Fonctionnement normal.....	27
7.	Annexes	28
7.1.	Liste du matériel et références.....	28
7.2.	Références.....	28
7.3.	Ressources : Liens Git Hub	28
7.4.	Conception mécanique : Mises en plan	30
8.	Table des figures	35

1. Introduction

Ce document a pour objectif d'expliquer en détail les choix et solutions mises en place pour la réalisation du projet de fin d'étude. Ce rapport est organisé en plusieurs sous-parties : Électronique, mécanique et logicielle. En effet, comme défini dans le cahier de spécifications, l'objectif de ce projet est de réaliser un prototype abouti d'un système de stabilisation d'une caméra d'action sur trois axes. Le porteur du projet qui agit également en tant que MOE est Sébastien Laporte, étudiant en cinquième année de cycle d'ingénieur en informatique industrielle. L'encadrant du projet représentant la MOA est Monsieur Ameur Soukhal, enseignant chercheur au CNRS.

2. Contexte et objectif du projet

Depuis quelques années, le marché des caméras d'action est en constante évolution. C'est la marque GoPro qui a démocratisé ce type de matériel et qui s'est imposée comme leader dans le domaine. Grâce à cette gamme de caméra, il est possible de filmer ses activités sportives, sur tous les terrains avec un minimum d'encombrement. Cependant, malgré la miniaturisation et la qualité du capteur, il est souvent difficile de filmer de manière stable lorsqu'on conduit une moto ou que l'on descend une piste de ski. L'objectif de ce projet est de concevoir et réaliser un système de stabilisation pour caméra d'action. Fixée sur le système, la caméra sera stabilisée de façon mécanique pour permettre à l'utilisateur de filmer tout en conservant une image stable. De nombreuses solutions existent aujourd'hui. Pour ce projet, nous avons la volonté de développer notre propre solution pour appréhender l'ensemble des problématiques et technologies.

3. Architecture générale du système

Le schéma ci-dessous présente l'architecture du système, il est issu du cahier de spécifications. Chaque élément est traité dans les différentes parties : électronique, mécanique et logicielle. Pour plus de précisions sur le fonctionnement du système, se référer au cahier de spécification au chapitre « Architecture générale du système ».

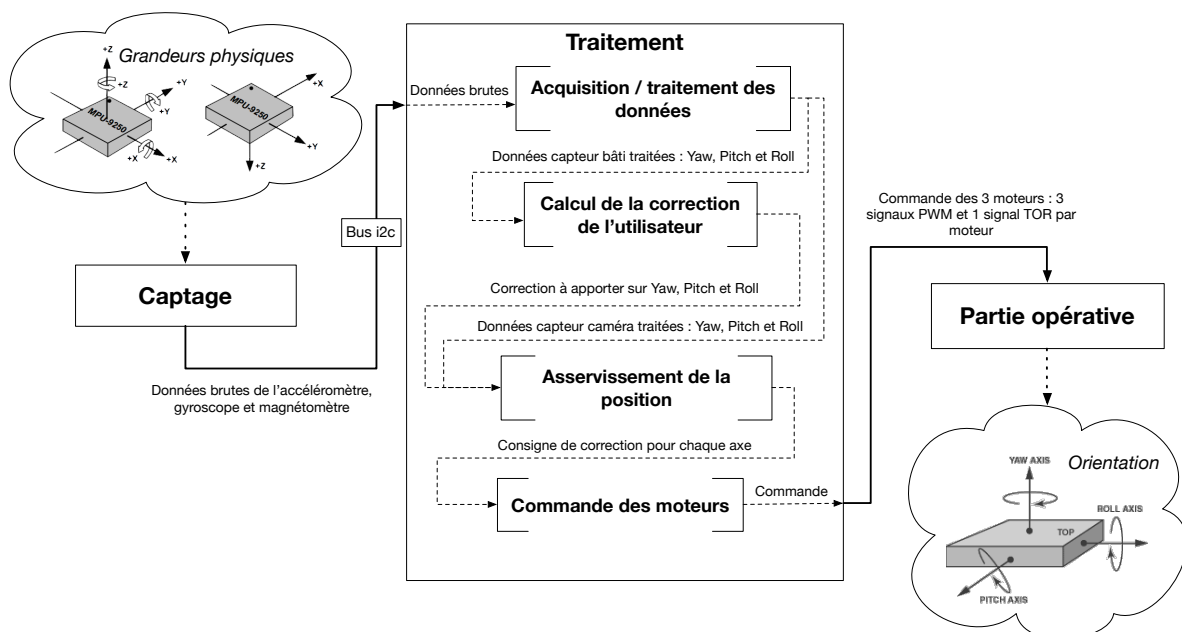


Figure 1 : Architecture générale du système

4. Partie électronique

Dans cette première partie, nous allons présenter l'ensemble des composants qui ont été choisis pour la partie électronique et la manière dont nous avons architecturé les fonctions électroniques.

4.1. Choix des technologies

Le choix des technologies a été réalisé suite à la phase de recherche et de spécifications. L'ensemble des références sélectionnées se trouvent en annexe. Le prix des composants se trouve sur les factures également en annexe.

Remarque : L'ensemble du matériel sélectionné est très spécifique à ce domaine d'application. En effet, lors de la phase de recherche, nous nous sommes intéressés au domaine des drones, notamment aux systèmes de stabilisation de caméra appelés « nacelles ». Ces technologies étant très spécifiques, la majorité des composants n'a donc pas nécessité d'étude comparative tant le nombre de références de produit est restreint.

4.1.1. Capteur

Au cours de l'étude qui a été menée pour le choix du capteur, nous avons sélectionné deux modèles pour la réalisation des prototypes 2 axes et 3 axes. En effet, dans un premier temps, nous avons réalisé un prototype sur 2 axes dans le but d'appréhender les notions inhérentes à cette technologie de capteur. Puis, pour répondre au cahier des charges, nous avons dû reprendre une phase d'étude pour choisir un nouveau capteur pour modéliser les mouvements sur 3 axes d'une façon optimale.

Remarque : Ce choix technologique est étroitement lié avec l'étude qui a été menée sur l'exploitation des données.

Pour modéliser l'orientation d'un solide dans l'espace, il existe 4 grandes familles de capteur :

- **Gyroscope** : Capteur permettant de mesurer la vitesse angulaire. L'unité de mesure est le degré / seconde ou encore le radian / seconde.
- **Accéléromètre** : Capteur permettant de mesurer l'accélération linéaire sur les axes X, Y et Z. L'unité de mesure est le m/s^2 ou peut également être exprimé en « g ».
- **Magnétomètre** : Également appelé boussole, ce capteur permet de mesurer le champ magnétique de l'environnement. L'unité de mesure est le Tesla ou le Gauss.
- **Altimètre** : Un altimètre permet de mesurer une distance verticale entre un point et une surface de référence en mesurant un niveau de pression atmosphérique.

4.1.1.1. Capteur pour un prototype sur deux axes

Pour modéliser l'orientation d'un solide sur les deux axes : Pitch et Roll, il est nécessaire d'utiliser un gyroscope et un accéléromètre. En effet, nous verrons dans la partie « Exploitation des données » que l'utilisation de ces deux types de capteurs permet d'obtenir les données souhaitées.

Nous avons sélectionné le capteur MPU-6050 de la société [InvenSens](#) qui est une référence incontournable dans ce domaine. Ce capteur a l'avantage de proposer dans un même composant un accéléromètre et un gyroscope ce qui élimine les problèmes d'alignement. La communication avec ce capteur s'effectue en i2c. La précision et la sensibilité de ce capteur est paramétrable : ± 250 , ± 500 , ± 1000 , and ± 2000 °/sec pour le gyroscope et $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$ pour l'accéléromètre ce qui permet d'avoir un champs d'application extrêmement large. Dans le cadre de ce projet, les sensibilités minimales seront suffisantes.

Ce capteur étant uniquement produit en CMS, nous avons sélectionné un fabricant proposant ce capteur sur un PCB avec les composants nécessaires à son fonctionnement : Le capteur IMU 6 DoF MPU-6050 par Drotek. La carte peut être alimentée en 5 ou 3,3v.

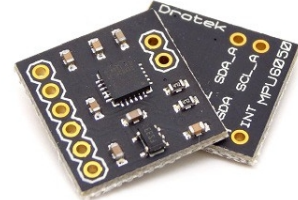


Figure 2 : MPU-6050 par Drotek

4.1.1.2. Capteur pour un prototype sur trois axes

L'ajout de l'axe Yaw pour la modélisation sur trois axes ajoute une complexité technique. Au cours de mes recherches, j'ai pu rencontrer plusieurs solutions permettant d'obtenir l'orientation les 3 axes. Dans certains cas, l'orientation est obtenue uniquement grâce à un accéléromètre et un gyroscope. Cependant cette solution n'est pas optimale. En effet, un gyroscope « drift » au cours du temps, ce qui donne une orientation erronée sur l'axe Yaw. Pour compenser ce glissement, il est possible d'ajouter un autre capteur : un magnétomètre. Cette solution sera plus détaillée dans la partie « Exploitation des données ».

La société InvenSens propose également un capteur doté d'un gyroscope, d'un accéléromètre et d'un magnétomètre dans un même composant : Le MPU-9250. Là encore, l'entreprise Drotek propose ce capteur sur un PCB intégrant les composants électroniques nécessaires à son fonctionnement. La carte peut être alimentée en 5 ou 3,3v.

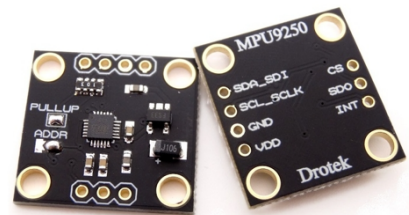


Figure 3 : MPU-9250 par Drotek

4.1.2. Motorisation

4.1.2.1. Les moteurs

La motorisation est un élément essentiel du système car ce sont les moteurs qui vont être garant de la stabilisation de la caméra. Il existe aujourd'hui sur le marché une très large gamme de moteurs. Lors de la phase de recherche, nous avons constaté qu'il existe deux types de moteurs utilisés pour ce type d'application : Les servomoteurs et les moteurs brushless. Les servomoteurs ont l'avantage d'être relativement simple à commander. De plus, de nombreuses bibliothèques permettent de faciliter leur commande. L'inconvénient est qu'ils sont peu précis : de l'ordre 1°. C'est à cause de cette précision insuffisante que nous avons décidé de ne pas utiliser de type de moteur.

Les moteurs brushless sont usuellement couplés à un capteur à effet hall pour connaître la position de celui-ci et effectuer l'asservissement. Lors de nos recherches, nous avons découvert un nouveau type de moteur qui est largement utilisé aujourd'hui pour motoriser les nacelles de drones. Ces moteurs sont dits BLDC pour Brushless DC. Ce type de moteur est commandé par 3 signaux PWM comme le montre la figure ci-dessous.

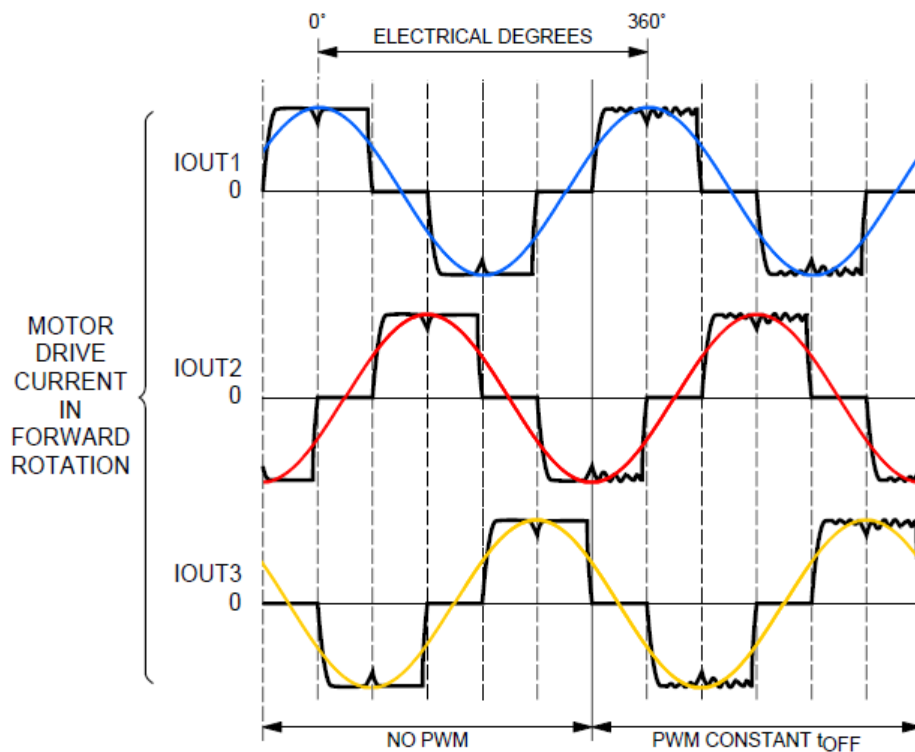


Figure 4 : Signaux de commande d'un moteur BLDC

On peut observer sur la figure ci-dessus qu'il est possible d'obtenir une position en degrés à partir d'un trinôme de signaux PWM. Lors de tests sur un banc d'essai, nous avons pu constater une précision de 0,1° ce qui est tout à fait satisfaisant pour notre système.

Pour trouver une référence de ce type de moteur, nous nous sommes orientés vers une boutique spécialisée dans les pièces détachées pour drones. Le choix s'est porté sur une référence d'entrée de gamme car elle correspond à notre besoin. En effet, le type de caméra que nous souhaitons stabiliser est relativement légère (environ 110g). Nous avons choisi la référence GBM3508-130 qui permet de stabiliser des caméras dans une gamme de poids allant de 200 à 400g, ce qui est largement en dessous de notre spécification. Néanmoins, cette sécurité est justifiée par le poids non négligeable de la structure à réaliser, qui peut être plus important que celui des structures du commerce (habituellement en plastique ou en carbone).



Figure 5 : Moteur BLDC

4.1.2.1. Les drivers de moteur

La technologie de ces moteurs BLDC implique l'utilisation d'un driver spécifique. [STmicroelectronics](#) commercialise un driver pour ce type de moteur : Le L6234 est un conteneur à trois ponts en H et permet de commander en puissance le moteur brushless DC. Ce composant existe au format DIP20 et en CMS. Il permet de commander les moteurs jusqu'à une fréquence de 50 KHz ce qui correspond au besoin. La plage d'alimentation du moteur peut varier entre 7 et 52v avec un courant maximum de 5A. La tension d'alimentation recommandée pour les moteurs sélectionnés est entre 7 et 12v ce qui correspond à la plage de fonctionnement. Le courant maximum de 5A ne sera jamais atteint avec le poids de la caméra utilisé.

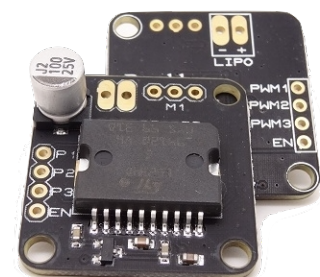


Figure 6 : L6234 par Drotek

Afin de réaliser des tests sur un banc d'essai, nous avons fait le choix dans un premier temps d'utiliser une platine de développement intégrant un driver L6234 avec l'électronique nécessaire à son fonctionnement. Cette Breakout Board est conçue par la société Drotek. La conception électronique qu'ils ont réalisé s'appuie sur une note d'application fournie par ST.

4.1.3. Microcontrôleur

L'élément central du système doit permettre d'effectuer l'acquisition des données par le biais d'un bus I2C et de pouvoir commander les trois moteurs. Nous verrons dans la partie « Exploitation des données » que le traitement requiert une grande capacité de calcul. La commande des trois moteurs implique 9 sorties permettant de produire un signal PWM ainsi que 3 sorties TOR. De plus, il est nécessaire de faire fonctionner ces PWM à une fréquence de 32 KHz.

Dans un premier temps, nous avons étudié l'ensemble des solutions qui étaient à notre disposition. L'utilisation d'un PIC a rapidement été éliminée. En effet, ce type de microcontrôleur ne dispose du nombre de sorties PWM requis pour le système. De plus, la fréquence de fonctionnement n'est pas suffisante. Nous avons envisagé la solution du co-design avec l'utilisation d'un FPGA. Cependant, cette solution n'aurait pas permis d'aboutir à un prototype suffisamment compact et transportable à cause de l'utilisation d'une plateforme de développement. Pour finir, nous avons recherché un microcontrôleur sur base ARM. Ce type de matériel dispose d'un grand nombre d'entrées / sorties et peut fonctionner à des fréquences intéressantes pour notre système. Nos recherches nous ont mené à la carte Teensy 3.1.

La Teensy est une plateforme de développement extrêmement compacte, animée par un microcontrôleur ARM Cortex M4 qui peut fonctionner à 96Mhz et qui dispose de 12 sorties capable de produire un signal PWM. Cette carte peut être alimentée en 5v ou en 3,3v. Ses dimensions sont : 35 mm x 18 mm.

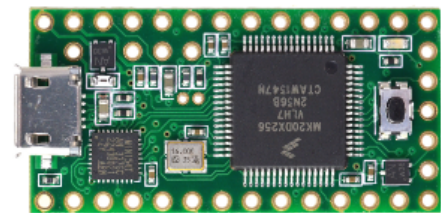


Figure 7 : Teensy 3.1

4.1.4. Alimentation

Le système possède deux besoins en alimentation : Une alimentation très basse tension pour l'alimentation des éléments de commande (Capteur, microcontrôleur) et une alimentation dite de puissance pour les moteurs (par l'intermédiaire des drivers). Comme indiqué dans le cahier des charges, le système doit être autonome en énergie.

4.1.4.1. Choix de la batterie

Les moteurs sélectionnés peuvent consommer au maximum jusqu'à 5A chacun. Avec 3 moteurs, il est donc nécessaire d'avoir une batterie qui puisse fournir un courant maximum de 15A environ. Pour répondre à ce besoin, la technologie LIPO est tout à fait appropriée. Ce genre de batterie est utilisée pour motoriser des drones. Ayant besoin d'une tension comprise entre 7 et 12v, nous avons choisi d'utiliser une batterie LIPO 3S (qui signifie que la batterie possède 3 éléments). La tension de cette batterie est de 11,7v. Nous avons choisi dans un premier temps d'utiliser une batterie de 2200 mAh. Cette valeur pourrait être ajustée lors de tests afin de correspondre à une autonomie d'1 heure qui est spécifiée dans le cahier des charges. La batterie sélectionnée possède un courant maximum de 70A, ce qui est largement suffisant pour notre besoin.

4.1.4.2. Régulation de l'alimentation en 3,3v

L'alimentation des capteurs et du microcontrôleur peut être faite en 5 ou 3,3v. Concevant un système qui doit être autonome, nous souhaitons réduire au maximum la consommation. Pour cela, nous choisissons d'alimenter ces éléments en 3,3v. Pour réaliser cette fonction, nous avons décidé de réaliser une alimentation à découpage.

En effet, cette solution est la plus appropriée pour ce type d'application car il est moins consommateur d'énergie qu'une simple alimentation basée sur un régulateur de tension. De plus, cette solution peut être simplement mise en place par l'utilisation d'un composant tel que le LM2574. Celui-ci permet de réguler une tension d'entrée avec une tension maximale de 40v ce qui est bien au dessus de la tension de la batterie sélectionnée. Ce composant est disponible au format DIP-8.

4.2. Schémas fonctionnels

Les schémas fonctionnels ont pour objectif de définir l'ensemble des fonctions électroniques du système. Pour chaque sous-ensemble, nous avons identifié les signaux en entrée et en sortie.

Remarque : L'alimentation n'est pas mentionnée dans l'ensemble des blocs.

4.2.1. Fonction « Alimentation »

La fonction d'alimentation permet de fournir une tension régulée de 3,3v à partir de la batterie. Cette fonction permettra d'alimenter le microcontrôleur ainsi que le capteur.

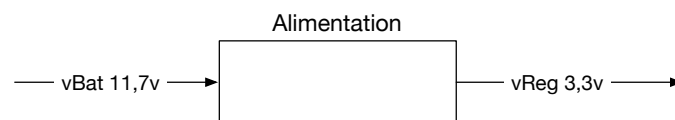


Figure 8 : Partie électronique - Schémas fonctionnels - Fonction alimentation

4.2.2. Fonction « Traitement des données »

Le traitement des données est assuré par le microcontrôleur. Ce bloc fonction permet d'identifier l'ensemble des signaux. En entrée nous avons SDA et SCL qui correspondent au bus i2c, pour la communication avec le capteur. En sortie, ce sont les signaux qui permettent de commander les moteurs par l'intermédiaire des driver de moteur.

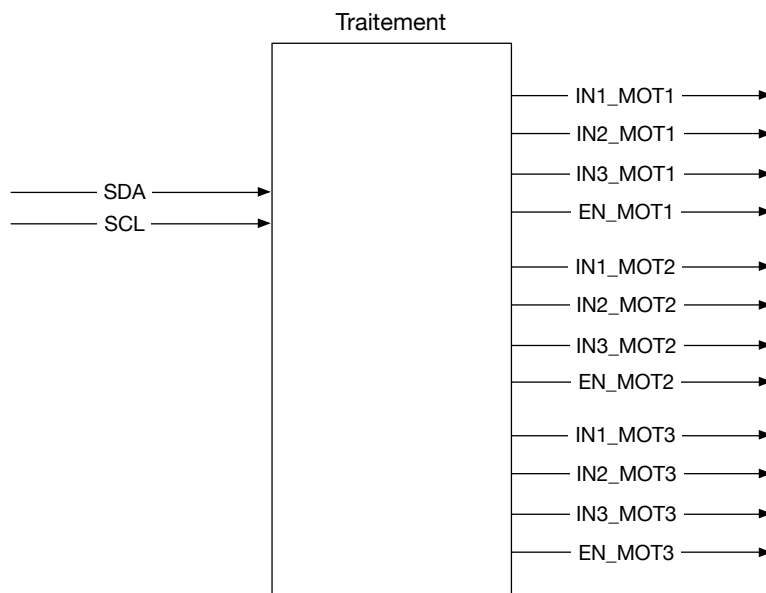


Figure 9 : Partie électronique - Schémas fonctionnels - Fonction traitement des données

4.2.3. Fonction « Captage »

La fonction captage permet de récupérer les informations d'orientation. Cette fonction est assurée par le capteur i2c.

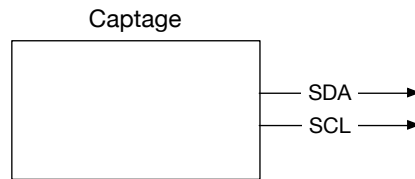


Figure 10 : Partie électronique - Schémas fonctionnels - Fonction captage

4.2.4. Fonction « Commande moteur »

Pour commander en puissance les moteurs, nous utilisons le driver de moteur L6234. C'est ce composant qui remplit la fonction « Commande moteur » en fonction des signaux issus du microcontrôleur. En sortie de ce bloc fonction, ce sont les trois phases du moteur.

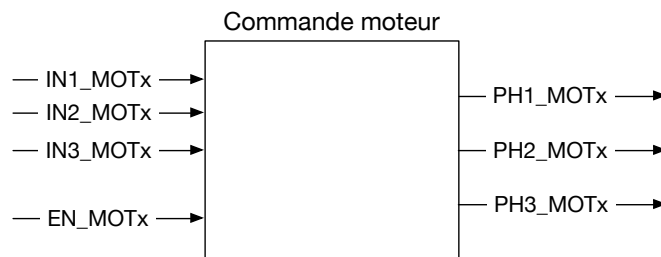


Figure 11 : Partie électronique - Schémas fonctionnels - Fonction commande moteur

4.2.5. Schéma fonctionnel du système

Sur le schéma fonctionnel ci-dessous, on retrouve l'ensemble des fonctions précédemment présentées.

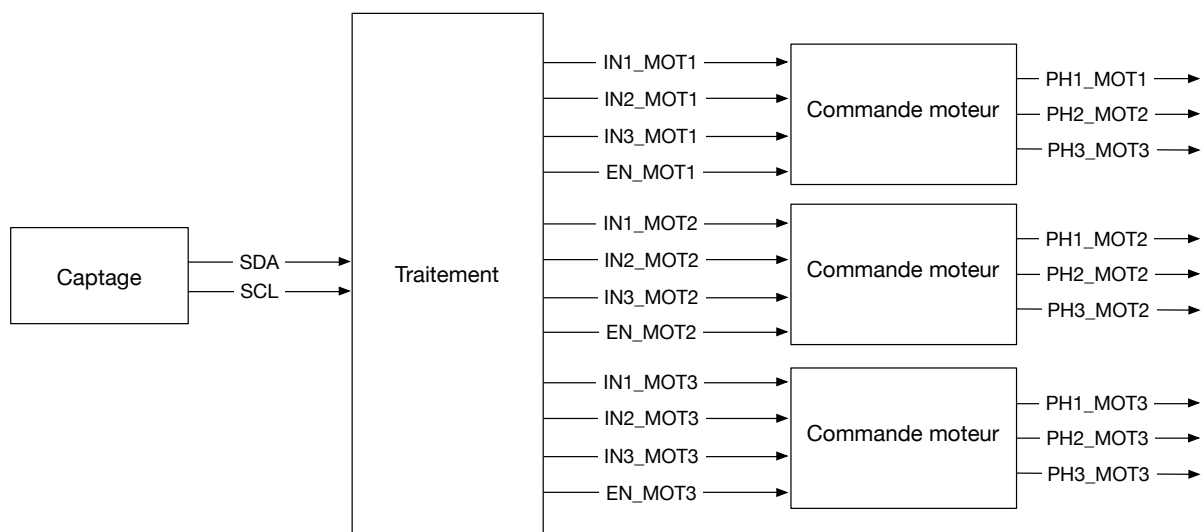


Figure 12 : Partie électronique - Schéma fonctionnel du système

En terme d'alimentation, les blocs fonction « Captage » et « Traitement » sont alimentés en 3,3v par le bloc fonction « Alimentation ». Les trois blocs fonction de « Commande moteur » sont eux directement alimentés par la batterie.

5. Partie mécanique

Ce nouveau chapitre est dédié à la modélisation et à la conception mécanique du système. L'objectif de la partie mécanique du système est d'assurer le maintien de la caméra sur trois articulations effectuées par les moteurs. Cette partie comprend également la modélisation des éléments de protection des cartes électronique.

5.1. Schéma du système

Le schéma ci-dessous définit le repère du système et les noms associés aux moteurs, il est issu du cahier de spécifications. Lors de la phase de spécifications, nous avons spécifié l'utilisation de deux capteurs : un sur le bâti et un sur la caméra. Lors de la phase d'analyse, nous nous sommes aperçu que le capteur du bâti n'est pas nécessaire pour assurer le fonctionnement du système tel qu'il est spécifié. Cette explication sera détaillée dans la partie logicielle avec la modélisation de la fonction « Correction de l'orientation ».

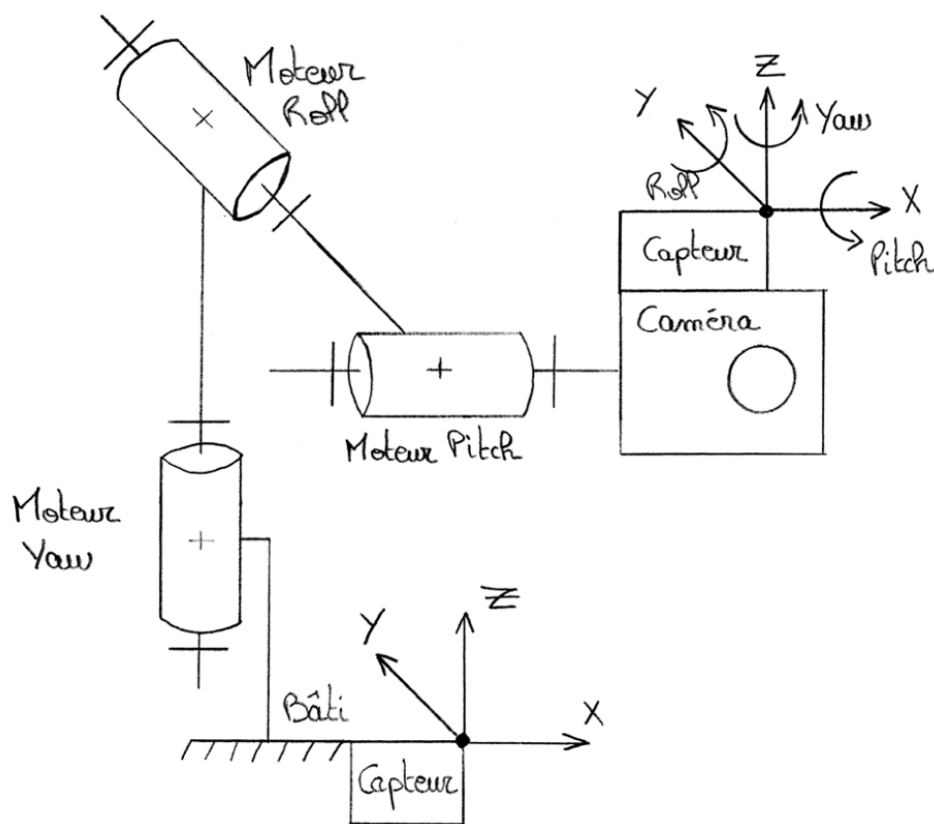


Figure 13 : Schéma général du système

La schématisation de la structure mécanique est nécessaire avant d'effectuer la modélisation du système par un logiciel de DAO. Dans un premier temps, nous avons entrepris des recherches pour observer les différentes structures réalisées sur des systèmes similaires. Cette conception doit également prendre en compte le type de matériaux à utiliser. En effet, dans le cadre de ce projet, le prototype doit être réalisable par nous même afin de limiter le coût de fabrication. Cependant, la réalisation doit pouvoir être sous traitée si l'on souhaite effectuer la production d'une petite série du système.

5.2. Modélisation de la structure

Une première modélisation de la partie mécanique sous formes de schémas permet de définir la structure de base du système. Ces schémas seront le point de départ de la CAO. Ils définissent également le nom de pièces maitresses de la structure.

5.2.1. Schémas complets

Le schéma ci-dessous représente le système en vue de côté. La structure est constituée de 3 articulations : Une pour chaque axe. Les bielles sont identifiées par le nom de l'axe d'articulation.

La bielle Pitch supporte le support de caméra. Celui-ci sera fixé par vis et écrous à main pour pouvoir être démonté rapidement par l'utilisateur. Le capteur est également lié à cette bielle pour estimer l'orientation de la caméra.

La carte électronique sera fixée sur la bielle Yaw car cet emplacement permet de limiter les torsions sur les câbles des moteurs. En effet, les deux moteurs liés à cette bielle (Moteur Roll et moteur Yaw) sont fixés par leur stator ce qui permet de les relier électriquement à la carte électronique sans avoir d'articulation entre les câbles.

Le bâti permet de fixer la batterie. C'est aussi cette partie qui permet à l'utilisateur de porter le système. En effet, grâce à l'épaisseur de la batterie, celle-ci constitue une véritable poignée pour l'utilisateur.

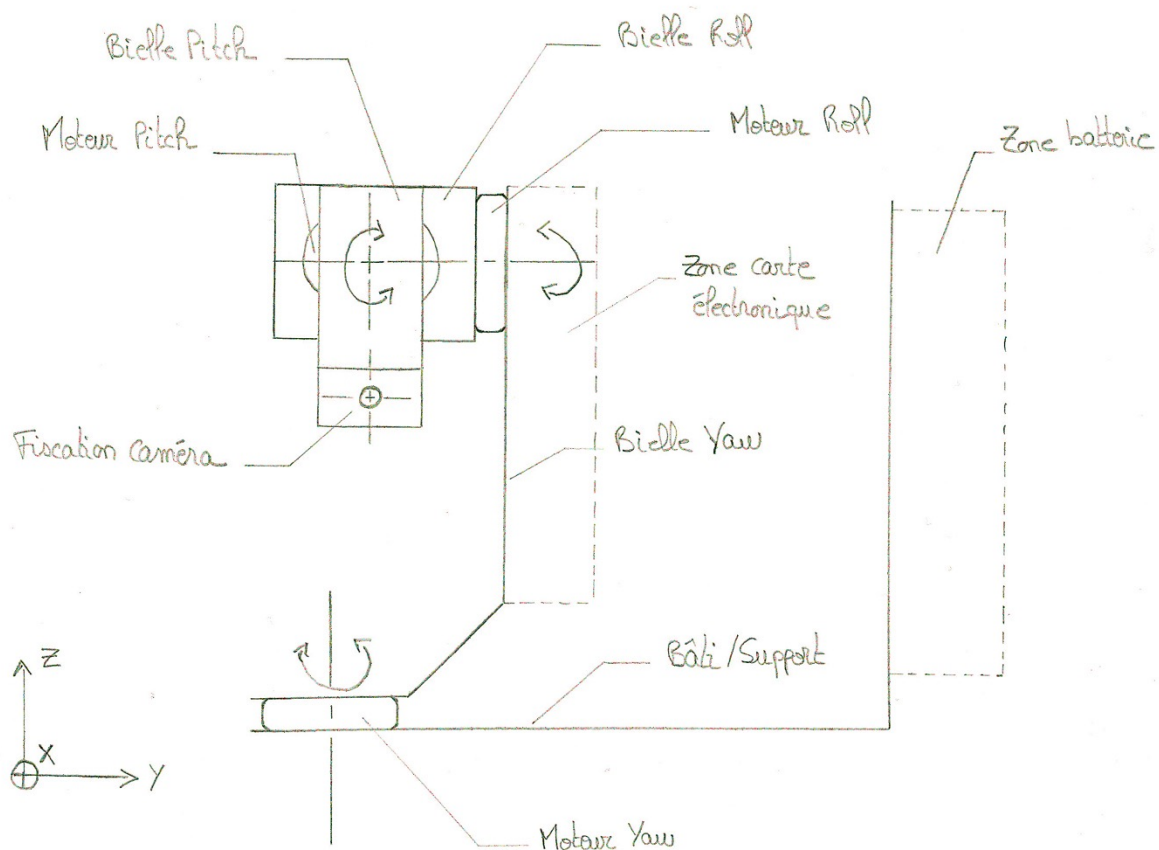


Figure 14 : Schéma structure mécanique - Vue de côté

Le schéma ci-après présente le système en vue de face. On retrouve les différents éléments présentés précédemment. L'objectif est d'obtenir un équilibre mécanique lorsque la caméra est installée pour limiter les efforts des moteurs. Cette contrainte est à prendre en compte pour la CAO.

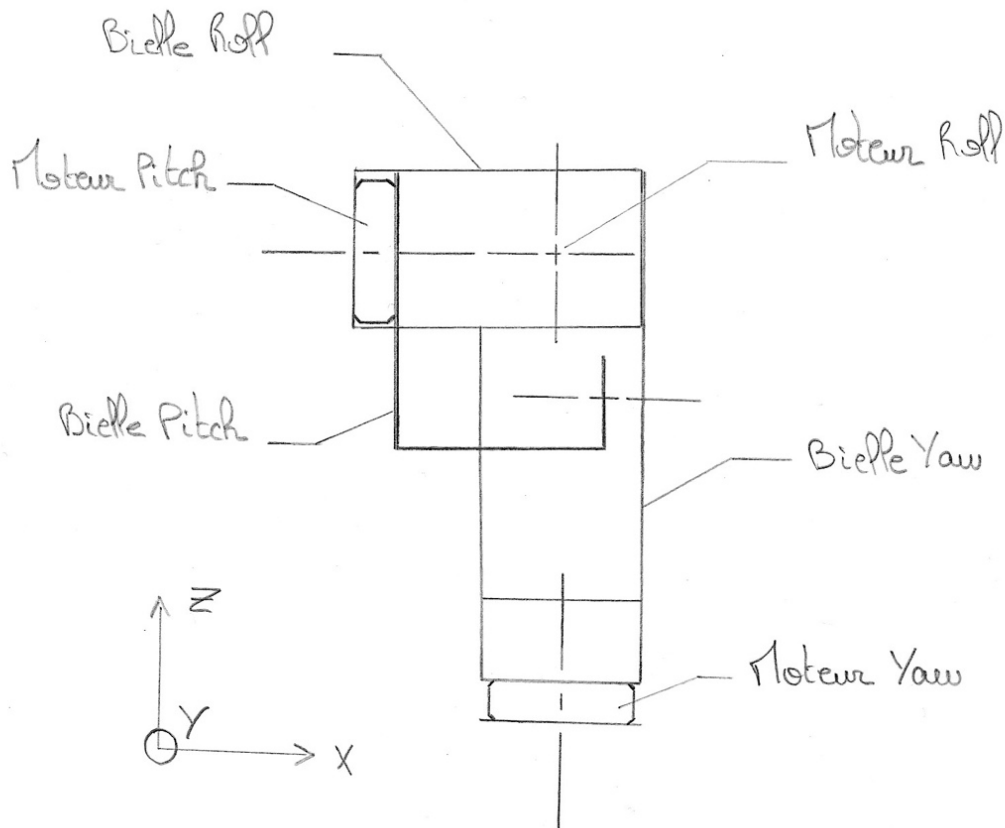
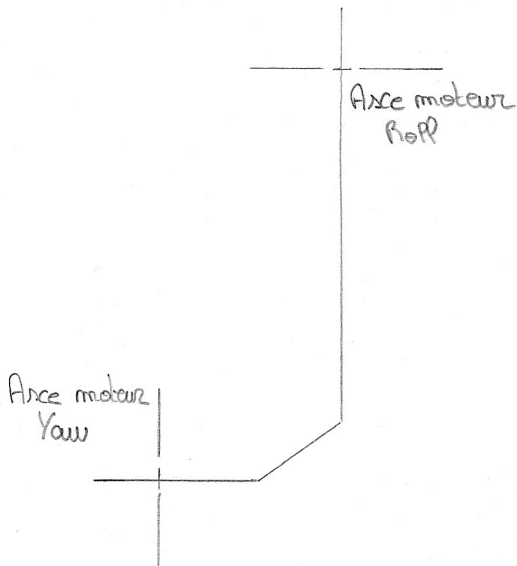
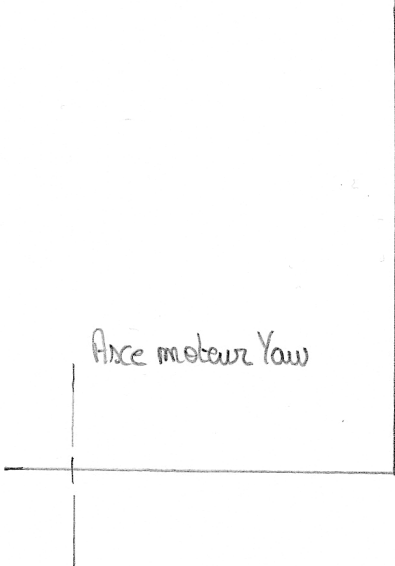
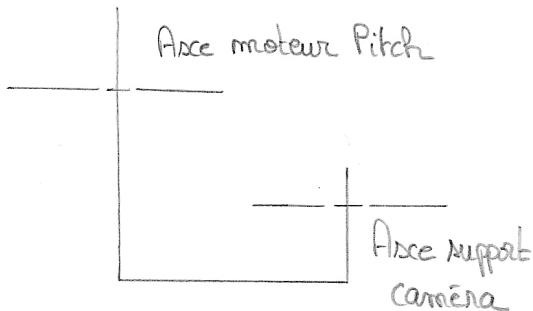
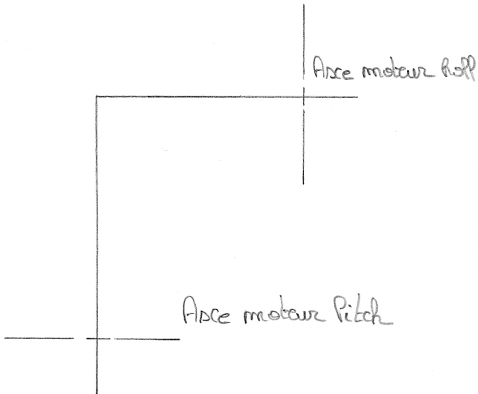


Figure 15 : Schéma structure mécanique - Vue de face

5.2.2. Schémas des pièces

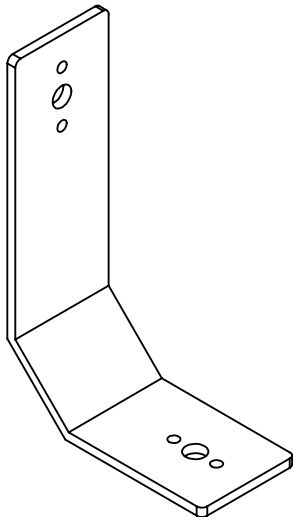
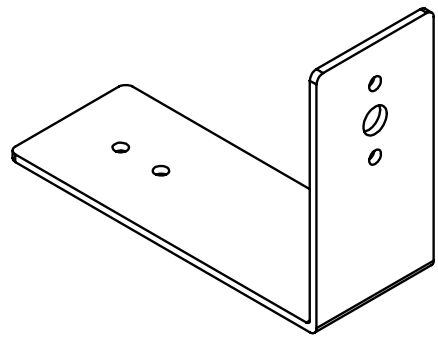
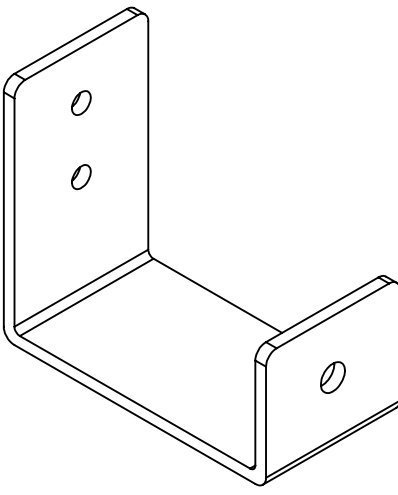
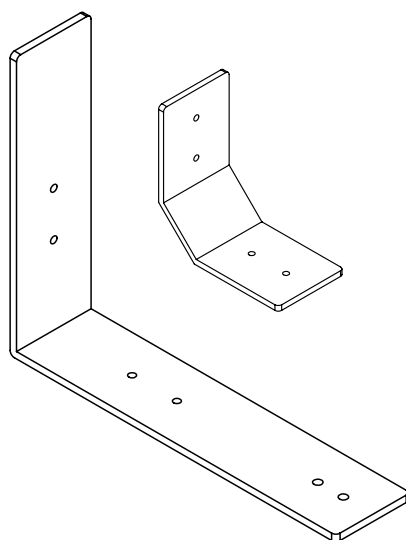
Bielle Yaw	Bâti / Support
 <p>Figure 16 : Modélisation mécanique - Bielle Yaw</p>	 <p>Figure 17 : Modélisation mécanique - Bâti / Support</p>

Bielle Pitch	Bielle Roll
 <p>Figure 18 : Modélisation mécanique - Bielle Pitch</p>	 <p>Figure 19 : Modélisation mécanique - Bielle Roll</p>

5.3. Conception des éléments de la structure

Nous avons utilisé le logiciel SolidWorks pour modéliser et dimensionner les pièces précédemment présentées. Cette conception permet de produire des plans de chaque pièce pour la réalisation.



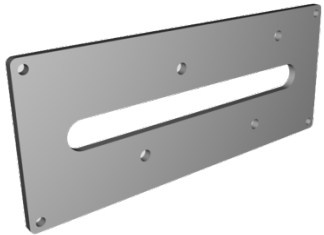
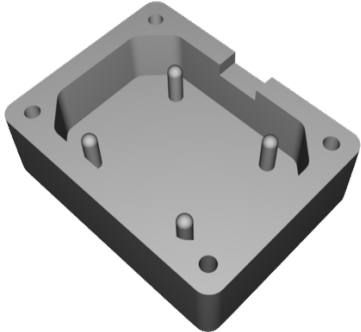

Remarque : L'ensemble des plans et fichiers 3D se trouvent en annexe de ce document.

Bielle Yaw	Bielle Roll
 <p>Figure 20 : CAO - Bielle Yaw 3D</p>	 <p>Figure 21 : CAO - Bielle Roll 3D</p>
Bielle Pitch	Bâti et renfort
 <p>Figure 22 : CAO - Bielle Pitch 3D</p>	 <p>Figure 23 : CAO - Bâti et renfort 3D</p>

5.4. Conception des éléments de protection

Afin de protéger la carte électronique et le capteur, nous avons conçu deux boîtiers de protection toujours avec le logiciel SolidWorks. L'objectif est de réaliser ces pièces par impression 3D. Lors de la conception, nous avons donc pris en compte les contraintes liées à ce genre de réalisation : éviter les angles vifs, épaisseur minimum des parois de 3mm.

Remarque : Un lien vers les fichiers STL est en annexe.

Boîtier carte électronique (Dessus)	Boîtier carte électronique (Dessous)	Couvercle carte électronique
		
Boîtier capteur IMU		Couvercle capteur
		

5.5. Choix techniques pour la réalisation

5.5.1. Matériaux utilisés pour la structure

La fabrication du prototype doit être réalisable avec les équipements de l'école. En effet, afin de limiter le coût du projet la réalisation sera effectuée par nos soins. Pour réaliser les pièces précédemment présentées, nous avons choisi d'utiliser des profilés en aluminium. Ce matériau est léger et peut être plié sans machine spéciale. L'épaisseur et les dimensions des profilés seront à déterminer au cours de la CAO, en prenant en compte les paramètres de résistance (poids de la caméra et des moteurs) et d'encombrement.

5.5.2. Technique de réalisation des éléments de protection

Pour la réalisation des boîtiers, il a été fait le choix d'utiliser l'impression 3D. En effet, ce type de pièce se prête très bien à ce type de réalisation car cela permet d'obtenir une pièce relativement complexe avec un très faible coût.

4.4.2. Fixation de la caméra

Pour fixer la caméra sur le système, nous avons choisi d'utiliser un boîtier GoPro type « Frame » comme illustré ci-dessous pour maintenir efficacement la caméra. La bielle Pitch est pourvue d'un trou de fixation pour l'assemblage. Cette solution permet également de s'adapter à plusieurs caméras de la marque GoPro.



Figure 24 : Partie mécanique - Fixation caméra

6. Partie logicielle

Cette partie présente l'ensemble des techniques et méthode utilisées pour remplir les fonctions du système. Ce chapitre présente les différentes méthodes envisagées et mises en place pour un prototype sur deux axes puis un prototype final sur trois axes.

Les deux techniques d'exploitation des données ont été mises en place sur un prototype. L'ensemble des techniques qui ont permis d'aboutir au choix de la solution ont également été testé et implémenté.

6.1. Exploitation des données d'un capteur IMU 6 DoF

Dans la phase de recherche de solution, nous avons dans un premier temps entrepris des tests avec un capteur IMU 6 DoF : le MPU-6050. Afin d'appréhender cette technologie, nous nous sommes limités à faire l'acquisition des données sur deux axes : Pitch et Roll.

Pour rappel, le capteur MPU-6050 possède deux type de capteur : un gyroscope et un accéléromètre. Pour obtenir l'orientation du capteur avec Pitch et Roll il est nécessaire d'effectuer un traitement sur les données brutes issues des capteurs. Pour cela, nous avons essayé deux méthodes : Le filtre le Kalman et le filtre complémentaire.

Le filtre de Kalman est un outil complexe à utiliser. Tout comme l'autre filtre que nous avons utilisé, il permet d'obtenir une estimation de l'orientation du capteur à partir des données d'un accéléromètre et d'un gyroscope. Les résultats obtenus avec ce filtre sont tout à fait satisfaisant. Cependant l'implémentation d'un tel filtre n'est vraiment pas aisé et l'ajustement des paramètres de celui-ci requiert un travail non négligeable pour arriver à une estimation correcte. Dans un second temps, nous nous sommes intéressé au filtre complémentaire qui est décrit dans la littérature comme beaucoup plus simple que celui de Kalman.

Le filtre complémentaire est relativement simple à mettre en place. En effet, le calcul de ce filtre tient en une seule ligne de calcul. De plus, l'ajustement des paramètres est assez rapide à effectuer. Ci-dessous, voici l'algorithme permettant de faire une estimation de l'orientation sur les deux axes : Pitch et Roll.

Dans un premier temps il est nécessaire d'appliquer une formule mathématique permettant d'obtenir Roll et Pitch. Ces valeurs sont non filtrées et ne sont pas exploitable :

$$\begin{aligned} roll &= \text{atan2}(\text{accY}, \text{accZ}) * \text{RAD_TO_DEG}; \\ pitch &= \text{atan}(-\text{accX} / \sqrt{\text{accY} * \text{accY} + \text{accZ} * \text{accZ}}) * \text{RAD_TO_DEG}; \end{aligned}$$

Figure 25 : Équations roll et pitch

Pour obtenir l'estimation de l'orientation, on applique le filtre complémentaire qui est de la forme suivante :

$$\begin{aligned} \text{compAngleX} &= 0.96 * (\text{compAngleX} + \text{gyroXrate} * dt) + 0.04 * roll; \\ \text{compAngleY} &= 0.97 * (\text{compAngleY} + \text{gyroYrate} * dt) + 0.03 * pitch; \end{aligned}$$

Figure 26 : Équations filtre complémentaire

Les variables gyroXrate et gyroYrate correspondent au données brutes du gyroscope en degrés par seconde. Les variables numériques correspondent au paramètres du filtre. L'addition de ces deux paramètres doit obligatoirement être égal à 1.

Au cours des tests, nous nous sommes aperçu que les résultats obtenus avec un filtre complémentaire sont presque identiques à ceux obtenus avec un filtre de Kalman. En conclusion, pour estimer l'orientation d'un solide sur les deux axes Pitch et Roll, la solution retenue est le filtre complémentaire.

Remarque : Les codes utilisés pour l'implémentation d'un filtre de Kalman et d'un filtre complémentaire sont disponibles en annexe de ce document.

6.2. Exploitation des données d'un capteur IMU 9 DoF

Dans la précédente partie, nous avons traité l'exploitation des données issues d'un capteur 6 axes pour modéliser l'orientation d'un solide sur deux axes. Cependant, l'utilisation d'un tel capteur pour modéliser l'orientation sur trois axes est problématique. En effet, l'ajout de l'axe Yaw complexifie le problème. Pour effectuer l'acquisition des données sur 3 axes, l'utilisation d'un simple filtre complémentaire ne suffit plus. C'est pour cela que nous avons essayé un autre capteur : le MPU-9250. Celui-ci intègre un magnétomètre en plus d'un accéléromètre et gyroscope.

Pour obtenir l'orientation du capteur sur les trois axes à partir des données des trois capteurs nous avons utilisé un algorithme de fusion nommé « [Algorithme de Madgwick](#) ». Sébastien Madgwick est un ancien étudiant de l'université de Bristol qui a mis au point un algorithme permettant de faire l'estimation de l'orientation d'un objet en 3D. Dans [son rapport](#) figure l'implémentation en langage C de son algorithme. Nous avons utilisé cet algorithme comme une boîte noire pour exploiter le capteur MPU-9250 et ainsi obtenir l'orientation sur Yaw, Pitch et Roll. En effet, ses travaux s'appuient sur les quaternions, un outil mathématique que nous ne maîtrisons pas. Ci-dessous, le diagramme représentant le filtre mis en place, prenant en compte les données d'un gyroscope, accéléromètre et d'un magnétomètre.

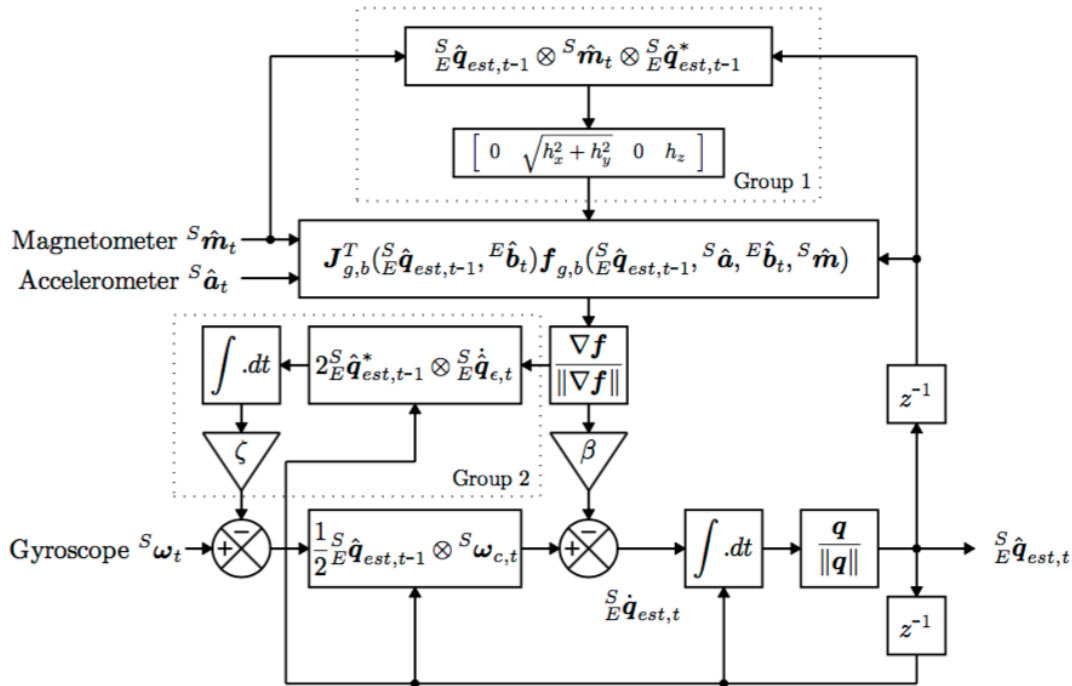


Figure 27 : Diagramme du filtre de Madgwick

Le magnétomètre permet de compenser le glissement naturel du gyroscope. En sortie du filtre, on obtient le résultat de l'orientation sous la forme d'un quaternion. Les formules suivantes permettent d'obtenir les composantes Yaw, Pitch et Roll à partir des quaternions.

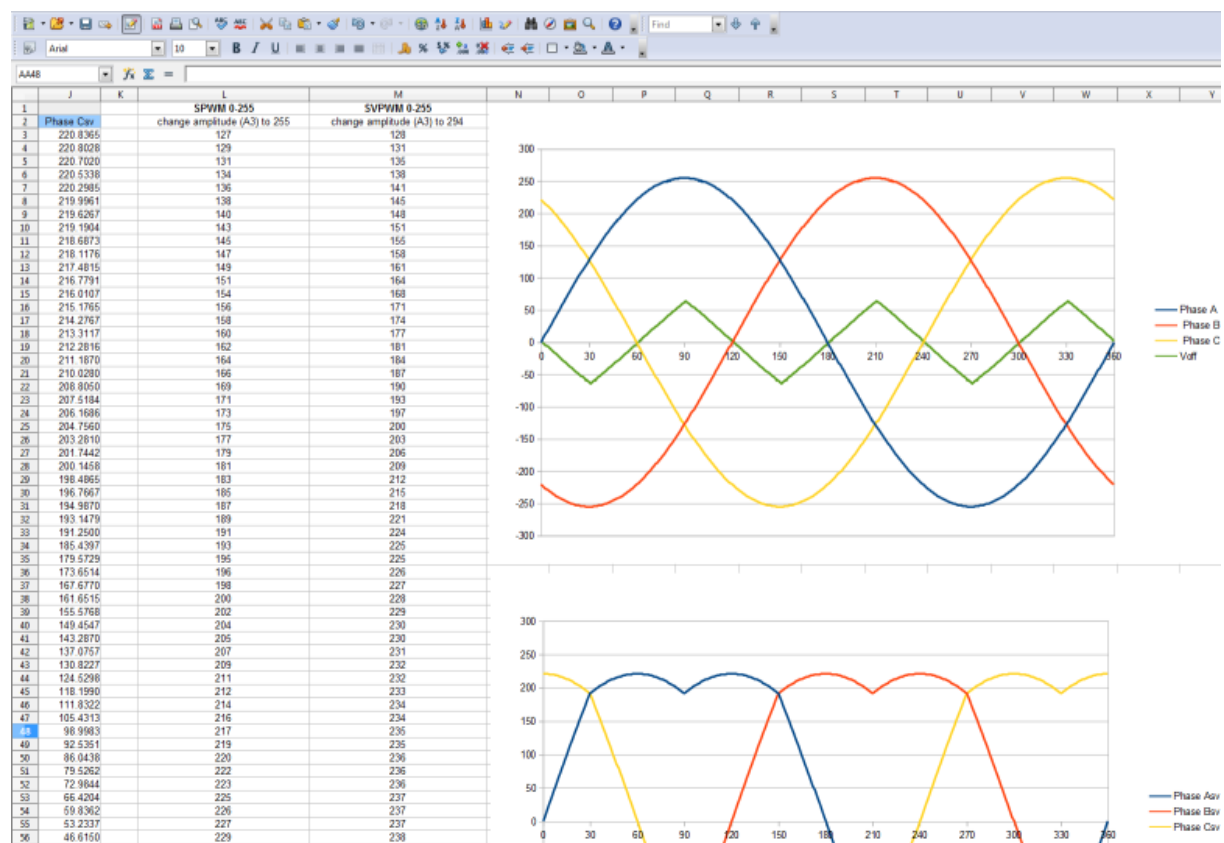
$$\begin{aligned} yaw &= \text{atan2}(2.0f * (q[1] * q[2] + q[0] * q[3]), q[0] * q[0] + q[1] * q[1] - q[2] * q[2] - q[3] * q[3]) \\ pitch &= -\text{asin}(2.0f * (q[1] * q[3] - q[0] * q[2])) \\ roll &= \text{atan2}(2.0f * (q[0] * q[1] + q[2] * q[3]), q[0] * q[0] - q[1] * q[1] - q[2] * q[2] + q[3] * q[3]) \end{aligned}$$

Figure 28 : Yaw Pitch Roll à partir des quaternions

6.3. Technique de commande d'un moteur BLDC

La commande d'un moteur brushless DC s'effectue grâce au driver le moteur L6234 précédemment présenté. Ce composant permet uniquement d'apporter la puissance nécessaire au moteur. Les trois signaux nécessaires à son fonctionnement sont générés par le microcontrôleur. Ce sont trois signaux de type PWM qui permettent de commander le moteur en position. Un trinôme de valeur permet de positionner le moteur. Il n'y a pas de point de référence : Habituellement, pour effectuer un asservissement, on place un capteur à effet hall sur l'axe du moteur. Le type de moteur utilisé permet de s'affranchir de cette contrainte matérielle.

Ces valeurs de PWM sont générées grâce à un tableur comme le montre l'illustration ci-dessous.



Cette technique de « Lookup table » permet d'éviter d'effectuer des calculs au niveau du microcontrôleur. Pour effectuer les premiers tests, nous avons dans un premier temps généré une table contenant 71 valeurs. Les moteurs utilisés possèdent 7 trinômes d'enroulements ce qui permet d'obtenir 497 pas pour 360 degrés ce qui représente une précision de 0,72 degrés. Pour augmenter la précision, il suffit de générer plus de valeurs (avec une limite de 255 valeurs : résolution des PWM configurées sur 8 bits).

Remarque : L'utilisation de signaux PWM pour la commande des ces moteurs peut provoquer des perturbations sonores en fonction de la fréquence. Pour que ces perturbations ne soient pas perceptibles par l'humain, des tests on montré qu'une fréquence de 18 KHz convient.

6.4. Modélisation des fonctions

La spécification de chaque fonction se trouve dans le cahier de spécifications.

6.4.1. Configuration du capteur

La configuration du capteur est une fonction essentielle au bon fonctionnement du système. En effet, la stabilisation du système se base uniquement sur les données issues du capteur. Il est donc nécessaire de le configurer correctement. Après étude de la datasheet et du référentiel des registres du capteur nous avons identifié les points suivants à configurer. Cette partie présente la marche à suivre pour l'implémentation.

Remarque : La datasheet du MPU-9250 et la référence des registres se trouvent en annexe.

6.4.1.1. Configuration accéléromètre et gyroscope : Les offsets

Les offsets permettent d'ajouter une valeur d'offset directement dans le capteur. Nous avons décidé de ne pas utiliser cette solution pour gérer ces valeurs directement au niveau du microcontrôleur.

- **Offsets du gyroscope** (0x13 à 0x18) : XG_OFFSET_H, XG_OFFSET_L, YG_OFFSET_H, YG_OFFSET_L, ZG_OFFSET_H, ZG_OFFSET_L
- **Offsets de l'accéléromètre** (0x77 à 0x7E) : XA_OFFSET_H, XA_OFFSET_L, YA_OFFSET_H, YA_OFFSET_L, ZA_OFFSET_H, ZA_OFFSET_L

6.4.1.2. Configuration accéléromètre et gyroscope : L'échelle

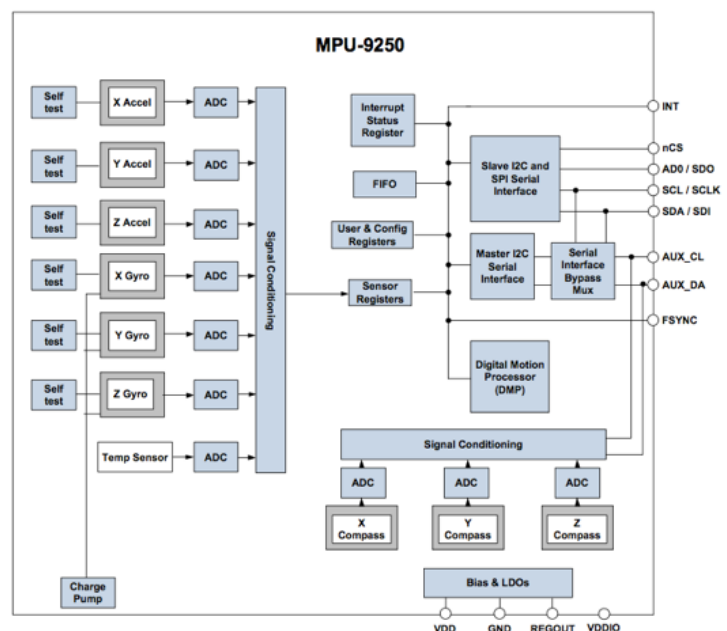
Les échelles pour définissent la plage d'utilisation du capteur. Les valeurs de configuration de ces échelles seront à définir lors des tests.

- **Échelle du gyroscope** (GYRO_CONFIG : 0x1B) : Bit 3 et 4 (GYRO_FS_SEL)
- **Échelle de l'accéléromètre** (ACCEL_CONFIG : 0x1C) : Bit 3 et 4 (ACCEL_FS_SEL)

6.4.1.3. Configuration de l'accès au magnétomètre

Le capteur MPU-9250 est divisé en deux partie : Un capteur gyroscope + accéléromètre et un magnétomètre. Le schéma ci-contre présente son architecture.

Pour accéder au magnétomètre, il existe deux solutions : La première consiste à utiliser la fonction de « Bypass » pour joindre le magnétomètre au bus i2c ce qui fait apparaître du point de vue du microcontrôleur deux périphériques i2c avec deux adresses distinctes. La deuxième solution est de paramétrer le MPU-9250 pour que son magnétomètre soit accessible par des registres dédiés à des esclaves. En effet, ce capteur peut accueillir d'autres capteurs dits « esclaves » (jusqu'à 4). Avec cette solution, le microcontrôleur peut accéder à l'ensemble



des données du capteur avec une seule adresse i2c ce qui limite l'accès au bus. C'est cette deuxième solution qui a été choisie.

Pour effectuer la configuration du magnétomètre en tant que « esclave », il faut renseigner les registres suivants :

- **I2C_SLV0_ADDR** (0x25) : Adresse i2c du périphérique identifié en tant que « Esclave 0 »
- Configuration du magnétomètre : Il est nécessaire de passer par les registres de transfert de données associés à l'esclave. Les deux registres ci-dessous sont nécessaires pour effectuer la configuration de chaque registre du capteur.
 - o **I2C_SLV0_REG** (0x26) : Registre permettant de spécifier l'adresse du registre du périphérique esclave sur lequel transférer des données.
 - o **I2C_SLV0_DO** (0x63) : Valeur du registre du périphérique que l'on souhaite modifier.
 - o **I2C_SLV0_CTRL** (0x27) : Permet d'activer l'i2c et d'effectuer l'écriture.

6.4.1.4. Configuration du magnétomètre

Pour rappel, la configuration des registres du magnétomètre doit être réalisée en utilisant les registres I2C_SLV0_REG, I2C_SLV0_DO et I2C_SLV0_CTRL. En effet, le capteur n'est pas relié directement au bus i2c menant au microcontrôleur.

- **Reset du capteur** : AK8963_CNTL2 (0x0B) à 0x00
- **Configuration** : AK8963_CNTL1 (0x0A) à 0x12
 - o **Mesures continue** : Bits 0 à 3 à 0x02
 - o **Résolution 16 bit** : Bit 4 à 1

6.4.2. Acquisition des données

La fonction d'acquisition des données doit permettre de récupérer les données brutes issues du capteur. Elle doit donc renvoyer trois trinômes de valeur correspondants aux données du gyroscope, de l'accéléromètre et du magnétomètre.

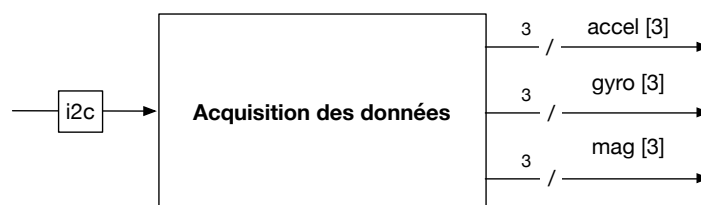


Figure 29 : Fonction "Acquisition des données"

La fonction d'acquisition des données est divisée en trois sous-fonction pour dissocier chaque type de capteur.

6.4.2.1. Acquisition des données du gyroscope

L'acquisition des données du gyroscope s'effectuant en lisant 6 octets à partir de l'adresse GYRO_XOUT_H (0x43). Les données du gyroscope sont codées sur 16 bit, il est donc nécessaire d'assembler les données une fois celles-ci lues. En effet, la donnée est divisée en deux parties : GYRO_...OUT_H (pour les bits de poids fort) et GYRO_...OUT_L (pour les bits de poids faible).

6.4.2.2. Acquisition des données de l'accéléromètre

L'acquisition des données de l'accéléromètre est similaire à celle des données du gyroscope. Il s'agit de lire 6 octets à partir de l'adresse ACCEL_XOUT_H (0x3B). Tout comme pour le gyroscope, les données sont codées sur 2 x 8 bits, il est donc nécessaire d'assembler les deux parties après lecture.

6.4.2.3. Acquisition des données du magnétomètre

L'acquisition des données du magnétomètre est différente. Il est nécessaire d'utiliser les registres du périphérique esclave :

- **I2C_SLV0_ADDR** (0x25) : On spécifie l'adresse i2c du magnétomètre tout en indiquant qu'on souhaite effectuer une lecture : 0x0C.
- **I2C_SLV0_REG** (0x26) : On indique l'adresse à partir de laquelle on souhaite effectuer la lecture : AK8963_XOUT_L (0x03).
- **I2C_SLV0_CTRL** (0x27) : On active l'i2c et demande la lecture de 6 octets : 0x87.
- On effectue une lecture de 6 octets à partir de l'adresse **EXT_SENS_DATA_00** qui contiennent les données codées sur 16 bit des 3 axes du magnétomètre.

6.4.3. Calibration du capteur

Le capteur doit être calibré pour que les données puissent être fiables. Chaque composant du capteur nécessite une calibration particulière. C'est grâce à ces données de calibration que l'on obtient les offsets à appliquer.

6.4.3.1. Calibration du gyroscope et de l'accéléromètre

Pour la calibration du gyroscope et de l'accéléromètre, on constitue un échantillon de données que l'on va ensuite moyenner. L'acquisition de cet échantillon doit se faire lorsque le capteur est immobile.

- Pour un **gyroscope**, lorsqu'il est statique, celui-ci doit renvoyer 0 sur les trois axes. La moyenne de l'échantillon correspondra donc à l'offset à appliquer et ceci pour chaque axe.
- Pour un **accéléromètre**, lorsqu'il est statique, celui-ci doit renvoyer 0 sur X, Y et renvoyer 1 sur l'axe Z qui est l'axe perpendiculaire au sol. Ainsi, en effectuant la moyenne des échantillons sur les trois axes, on obtient l'offset à appliquer.

$$offset_{axe} = \frac{\sum erreurs}{Nb_echantillons}$$

Figure 30 : Calibration du gyroscope et accéléromètre

6.4.3.2. Calibration du magnétomètre

La calibration du magnétomètre est différente. Elle nécessite d'orienter le capteur sur les 3 axes pour la constitution d'un échantillon suffisant de valeurs. Pour chaque axe, on doit enregistrer la valeur maximale et la valeur minimale renvoyée par le capteur lors des mouvements. Une fois les valeurs stables, on doit moyenne ces deux valeurs pour obtenir l'offset à appliquer :

$$offset_{axe} = \frac{val_{max} + val_{min}}{2}$$

Figure 31 : Calibration du magnétomètre

6.4.4. Traitement des données

Le traitement des données consiste à appliquer le filtre de Madgwick puis d'utiliser les formules permettant d'obtenir l'orientation du capteur sur Yaw, Pitch et Roll à partir des quaternions.

6.4.5. Correction de l'orientation

La correction de l'orientation permet de corriger l'orientation en fonction des mouvements volontaires de l'utilisateur. L'algorithme consiste à réduire au cours du temps la différence entre la position normale du système et la position obtenues pour le stabiliser.

6.4.6. Asservissement en position

L'asservissement en position utilise l'algorithme du PID comme décrit ci-dessous pour chaque axe.

Tous les x millisecondes, faire :

$\text{erreur} = \text{consigne} - \text{mesure}$

$\text{somme_erreurs} += \text{erreur}$

$\text{variation_erreur} = \text{erreur} - \text{erreur_précédente}$

$\text{commande} = K_p * \text{erreur} + K_i * \text{somme_erreurs} + K_d * \text{variation_erreur}$

$\text{erreur_précédente} = \text{erreur}$

Figure 32 : Algorithme PID - Asservissement

6.4.7. Commande de la motorisation

La commande de la motorisation s'appuie sur la « Lookup table » précédemment présentée. La commande de la motorisation consiste à se déplacer dans cette table pour faire tourner le moteur et donc changer l'orientation de la caméra.

6.5. Processus / fonctionnement

Le processus est découpé en deux parties. Une première phase d'initialisation permet de configurer le système et de préparer le fonctionnement normal. Ensuite, une routine de fonctionnement normale permet d'effectuer la stabilisation de la caméra et est constituée de plusieurs fonctions.

Remarque : Pour permettre d'effectuer l'étalonnage du capteur et le paramétrage de l'asservissement PID, le système repose sur un ensemble de paramètres. Il a été fait le choix de stocker ces paramètres en mémoire EEPROM. Cette solution a l'avantage de pouvoir modifier / mettre à jour ces valeurs pour effectuer des réglages au cours des tests. De plus, il est donc possible d'effectuer un nouvel étalonnage du capteur sans devoir modifier le code du programme.

6.5.1. Initialisation

La phase d'initialisation est une procédure qui permet d'initialiser le système. Le grafcet suivant décrit son fonctionnement. Le système utilise un grand nombre de paramètre, notamment en lien avec l'étalonnage du capteur. Ces paramètres sont chargés lors de cette phase. De plus, il est nécessaire de configurer le matériel avec notamment les interruptions (permettant de cadencer les différentes fonctions par la suite) ainsi que les entrées / sorties.

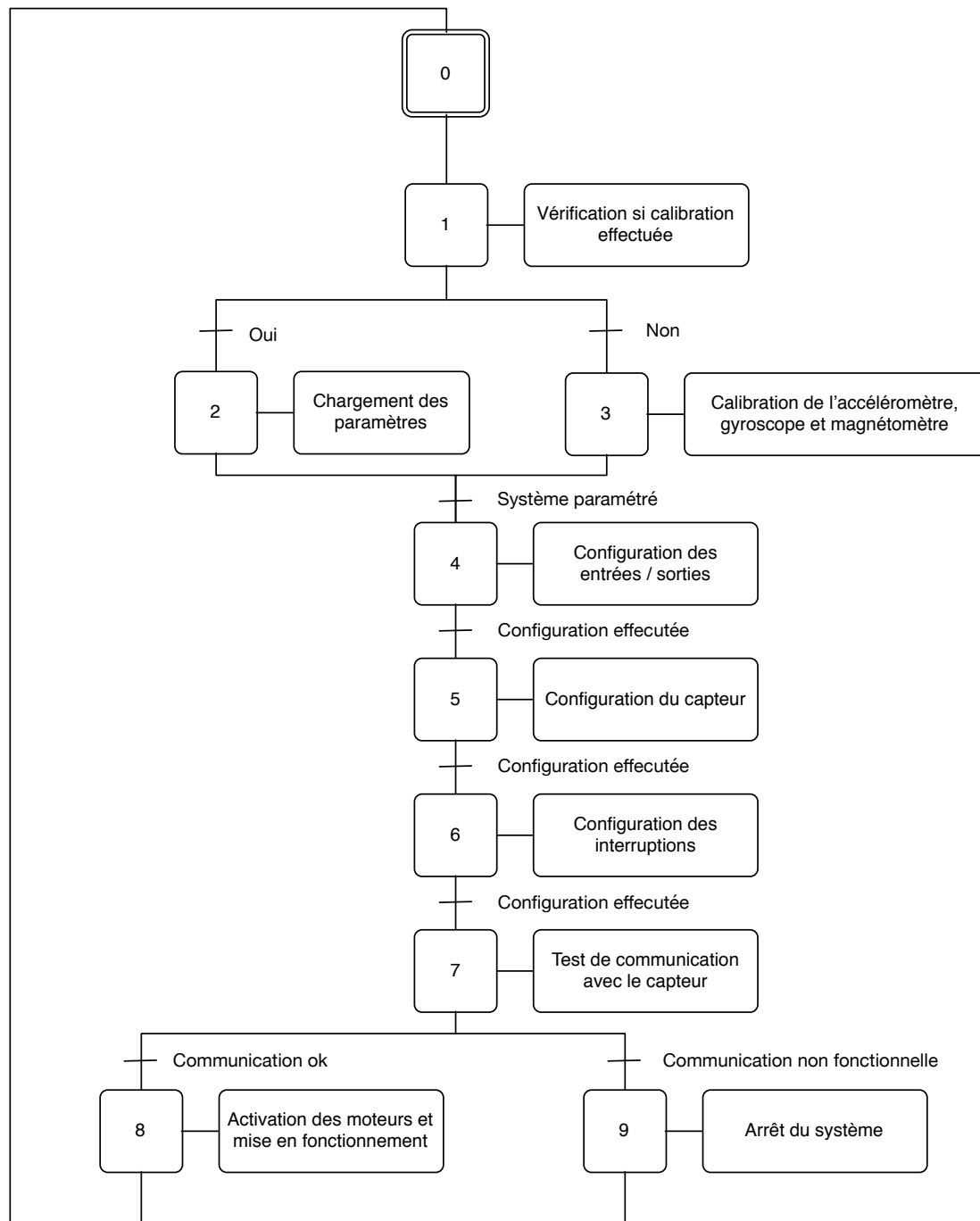


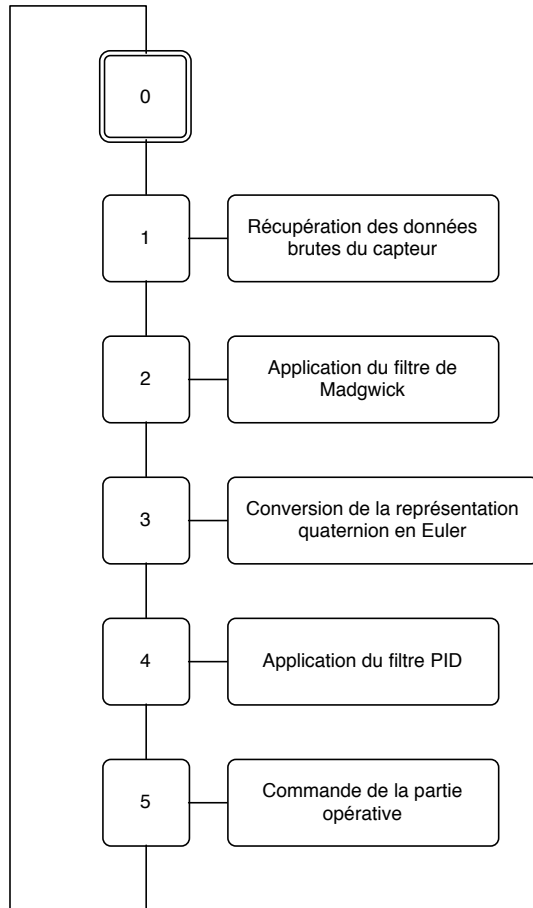
Figure 33 : Grafcet d'initialisation

À l'issue de cette procédure d'initialisation, le système est configuré et peut entrer dans la phase de fonctionnement normale décrit ci-après.

6.5.2. Fonctionnement normal

Le fonctionnement normal du système est assuré par vecteurs d'interruptions. En effet, il a été choisi de cadencer chaque fonction en utilisant les interruptions du microcontrôleur pour s'assurer que les opérations décrites ci-dessous sont effectuées avec une fréquence constante.

Le grafcet ci-contre illustre le fonctionnement normal du système. Cette représentation correspond aux opérations à effectuer pour pouvoir effectuer la stabilisation de la caméra.



L'ensemble de ces fonctions sont cadencées de la façon suivante :

Interruption 1 : Cadencée à 200 Hz

- Récupération des données brutes du capteur
- Conversion de la représentation quaternion en Euler

Interruption 2 : Cadencée à 50 kHz

- Application du filtre PID
- Commande de la partie opérative

Interruption 3 : Cadencée à 1 kHz

- Application du filtre de Madgwick

Remarque : Le filtre de Madgwick est un algorithme itératif. Pour effectuer l'estimation de l'orientation du capteur à partir d'un échantillon de valeur, la fonction doit être appelée 5 fois. Cela signifie que la fréquence d'appel de cette fonction doit être 5 fois plus importante que celle permettant l'acquisition des données. En effectuant des tests, nous avons constaté que le microcontrôleur utilisé peut faire fonctionner ce filtre à une fréquence de 1 kHz. Nous avons donc fixé la fréquence des données à 1 KHz / 5, soit 200 Hz.

Figure 34 : Grafcet de fonctionnement normal

7. Annexes

7.1. Liste du matériel et références

Microcontrôleur Teensy 3.1	http://www.robotshop.com/eu/fr/platine-developpement-microcontroleur-usb-teensy-32.html
IMU 6DoF - MPU-6050 sur PCB par Drotek	http://www.drotek.com/shop/fr/home/42-mpu6050-gyro-accelerometre.html?search_query=MPU6050&results=18
IMU 9 DoF – MPU-9250 sur PCB par Drotek	http://www.drotek.com/shop/fr/home/421-mpu9250-gyro-accelerometre-magnetometre.html?search_query=MPU9250&results=3
Moteur Brushless GBM3508-130	http://www.drotek.com/shop/fr/home/167-moteur-brushless-nacelle-gbm3508-130.html
Breakout Board L6234 par Drotek	http://www.drotek.com/shop/fr/home/212-controleur-nacelle-brushless-l6234.html
Batterie LIPO 2200 mAh	http://www.drotek.com/shop/fr/home/313-batterie-lipo-kypom-2200mah-3s-35c.html
Driver de moteur Brushless L6234	http://www.rs-particuliers.com/WebCatalog/Circuit_integre_pour_comman_de_moteur_Driver_de_moteur_triphas PDIP_20-8111016.aspx

7.2. Références

- Comparaison de différents algorithmes IMU : <http://www.olliw.eu/2013/imu-data-fusing/>
- Implémentation des algorithmes de Madgwick : <https://github.com/kriswiner>
- Commande d'un moteur BLDC : <http://www.berryjam.eu/2015/04/driving-blcd-gimbals-at-super-slow-speeds-with-arduino/>

7.3. Ressources : Liens Git Hub

- Exemple sur 2 axes avec un filtre de Kalman :

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Exemple_2_axes_kalman

- Exemple sur 2 axes avec un filtre complémentaire :

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Exemple_2_axes_filtre_complementaire

- Commande d'un moteur BLDC :

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Commande_moteur_brushless

- **Commande d'un moteur BLDC comme servo en c natif :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Commande_moteur_brushless_servo_c_natif

- **Exemple d'étalonnage du gyrosocope et accéléromètre :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Etalonnage_gyro_acc

- **Exemple d'étalonnage du magnétomètre :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Etalonnage_mag

- **Exemple d'étalonnage du capteur complet :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Etalonnage_gyro_acc_mag

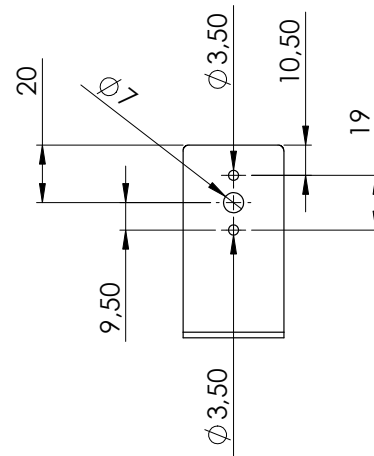
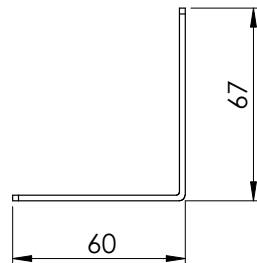
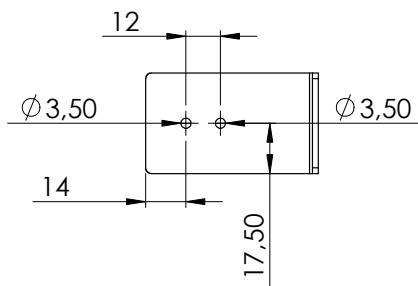
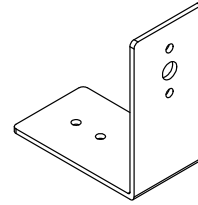
- **Asservissement en position (PID) :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Asservissement_en_position

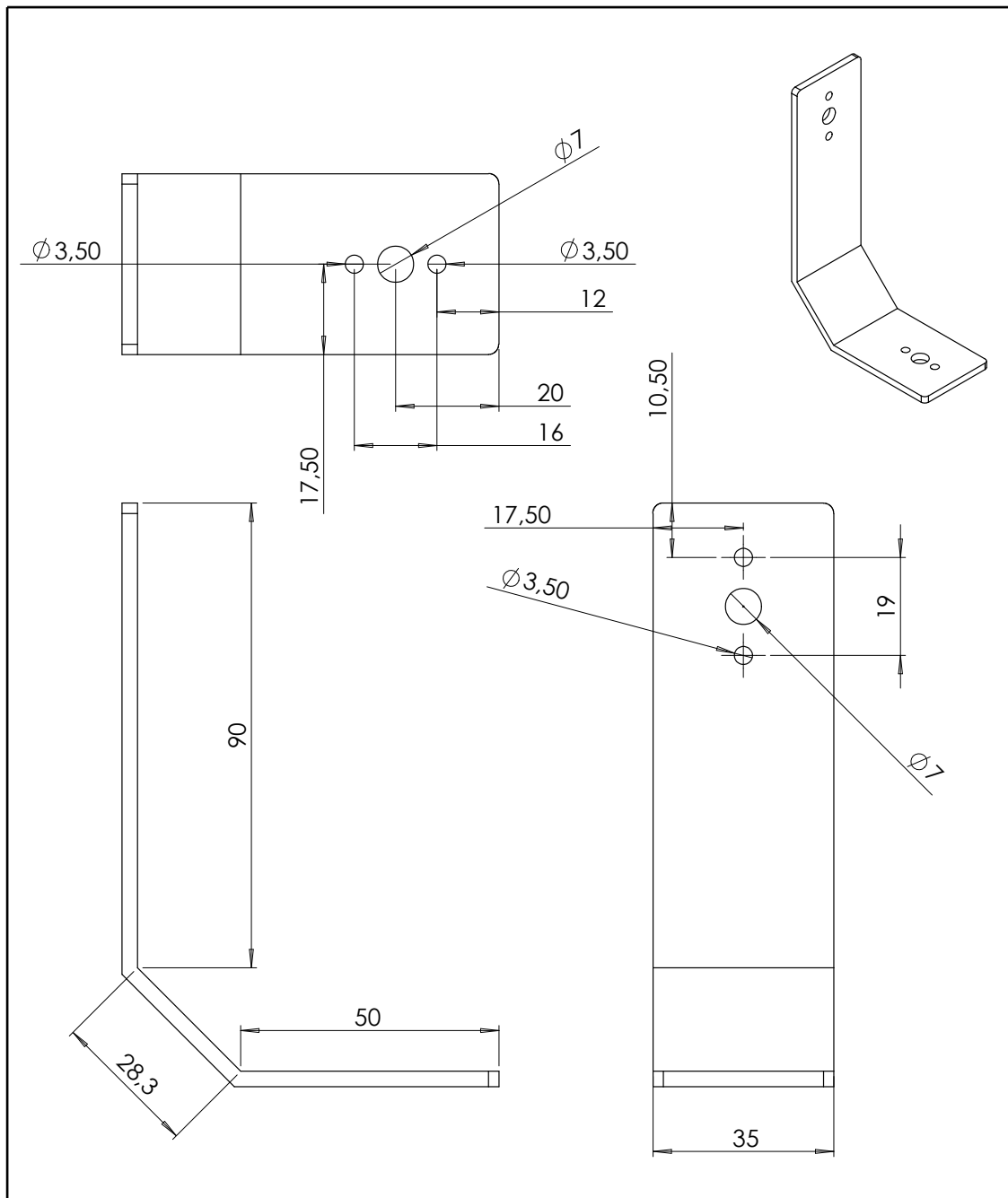
- **Exemple d'acquisition des données brutes du capteur :**

https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Acquisition_donnees_brutes

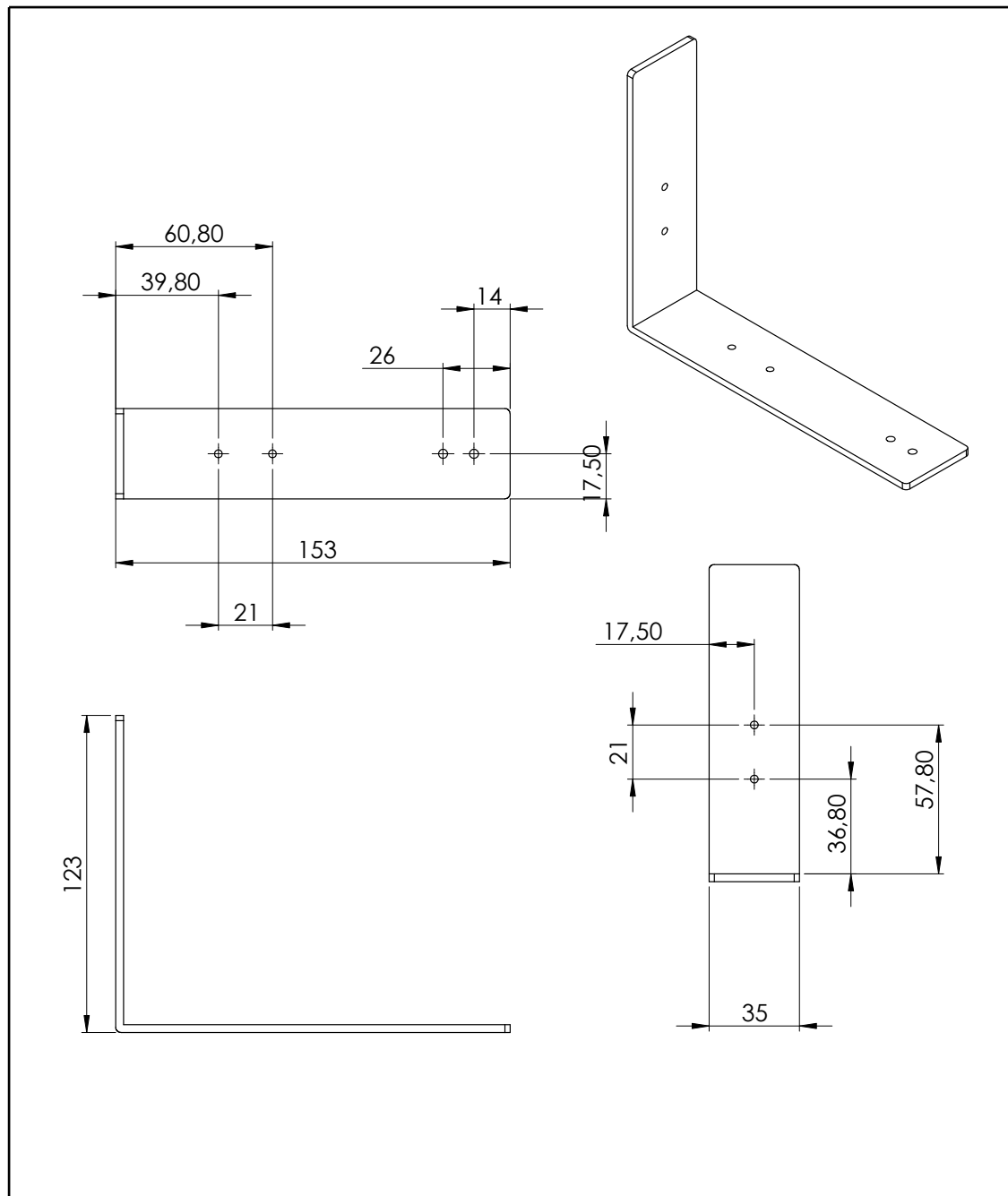
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:					FINITION:		CASSER LES ANGES VIFS		NE PAS CHANGER L'ECHELLE		REVISION
	NOM	SIGNATURE	DATE						TITRE:		
AUTEUR											
VERIF.											
APPR.											
FAB.											
QUAL.						MATERIAU:	No. DE PLAN			Bielle pitch	
										A4	
						MASSE:	ECHELLE:1:1		FEUILLE 1 SUR 1		



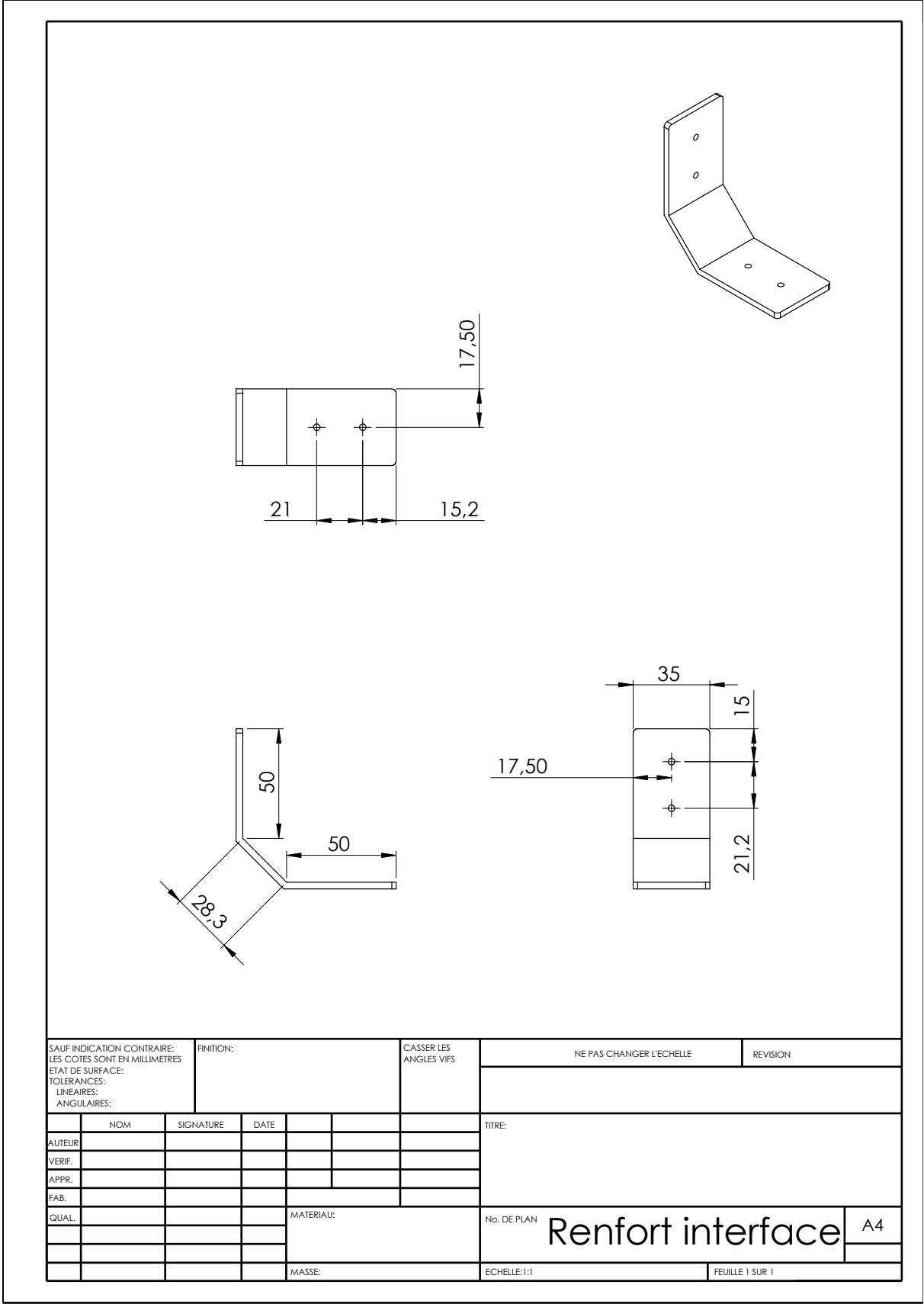
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAL:		No. DE PLAN		A4	
						Bielle roll			
				MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1	



SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
ETAT DE SURFACE:									
TOLERANCES:									
LINEAIRES:									
ANGULAIRES:									
NOM		SIGNATURE		DATE		TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.						MATERIAL:		No. DE PLAN	
								Bielle yaw	
								A4	
						MASSE:		ECHELLE:1:1	
								FEUILLE 1 SUR 1	



SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMETRES		FINITION:		CASSER LES ANGLES VIFS		NE PAS CHANGER L'ECHELLE		REVISION	
ETAT DE SURFACE:									
TOLERANCES:									
LINEAIRES:									
ANGULAIRES:									
	NOM	SIGNATURE	DATE			TITRE:			
AUTEUR									
VERIF.									
APPR.									
FAB.									
QUAL.				MATERIAU:		No. DE PLAN		Interface support	
								A4	
				MASSE:		ECHELLE:1:2		FEUILLE 1 SUR 1	



8. Table des figures

Figure 1 : Architecture générale du système	4
Figure 1 : MPU-6050 par Drotek	6
Figure 2 : MPU-9250 par Drotek	6
Figure 4 : Signaux de commande d'un moteur BLDC	7
Figure 4 : Moteur BLDC	7
Figure 5 : L6234 par Drotek.....	7
Figure 5 : Teensy 3.1.....	8
Figure 8 : Partie électronique - Schémas fonctionnels - Fonction alimentation	9
Figure 9 : Partie électronique - Schémas fonctionnels - Fonction traitement des données	9
Figure 10 : Partie électronique - Schémas fonctionnels - Fonction captage	10
Figure 11 : Partie électronique - Schémas fonctionnels - Fonction commande moteur.....	10
Figure 12 : Partie électronique - Schéma fonctionnel du système.....	10
Figure 13 : Schéma général du système.....	12
Figure 14 : Schéma structure mécanique - Vue de coté	13
Figure 15 : Schéma structure mécanique - Vue de face.....	14
Figure 16 : Modélisation mécanique - Bielle Yaw	15
Figure 17 : Modélisation mécanique - Bâti / Support	15
Figure 18 : Modélisation mécanique - Bielle Pitch	15
Figure 19 : Modélisation mécanique - Bielle Roll.....	15
Figure 20 : CAO - Bielle Yaw 3D	16
Figure 21 : CAO - Bielle Roll 3D.....	16
Figure 22 : CAO - Bielle Pitch 3D.....	16
Figure 23 : CAO - Bâti et renfort 3D.....	16
Figure 24 : Partie mécanique - Fixation caméra	18
Figure 25 : Équations roll et pitch	19
Figure 26 : Équations filtre complémentaire.....	19
Figure 27 : Diagramme du filtre de Madgwick	20

Figure 28 : Yaw Pitch Roll à partir des quaternions.....	21
Figure 29 : Fonction "Acquisition des données"	23
Figure 30 : Calibration du gyroscope et accéléromètre	24
Figure 31 : Calibration du magnétomètre	24
Figure 32 : Algorithme PID - Asservissement	25
Figure 33 : Grafcet d'initialisation	26
Figure 34 : Grafcet de fonctionnement normal	27