



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

Rapport de projet			
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	5/02/2016	Version de dépôt initial	

Remerciements

Je tiens tout d'abord à remercier mon tuteur, Monsieur Ameer Soukhal pour ses conseils, et m'avoir suivi tout au long de ce projet de fin d'études.

Je remercie également les membres du service informatique de Polytech Tours et tout particulièrement Sébastien Beaufils pour sa disponibilité et son investissement. Monsieur Mayaud pour son aide et ses conseils avisés sur la réalisation de la partie mécanique du prototype. Monsieur Gaucher pour des conseils qu'il m'a apporté sur les technologies de capteurs.

Je tiens également à remercier l'ensemble des personnes qui ont contribué à la finalisation de ce rapport et de l'ensemble des documents établis.

Enfin, je tiens à remercier ma compagne pour avoir supporté de transformer ponctuellement la cuisine de notre appartement en atelier de soudure, de découpe et de perçage.

Table des matières

1.	Introduction.....	4
1.1.	Contexte du projet.....	4
1.2.	Objectif.....	4
1.3.	Existant	4
2.	Cahier des charges	4
3.	Problématique	5
4.	Conduite de projet	5
4.1.	Méthodologie	5
4.2.	Découpage en tâches	7
4.3.	Choix stratégiques	7
4.4.	Planification	8
4.5.	Livrables du projet	8
5.	Schéma général du système	9
6.	Architecture mise en œuvre	10
7.	Matériel utilisé	11
8.	Conception et réalisation	12
8.1.	Conception de la structure	12
8.2.	Conception des éléments de protection	13
8.3.	Réalisation de la structure	13
8.4.	Partie électronique	14
8.5.	Partie logicielle	17
8.5.1.	Phase d'initialisation du programme	17
8.5.2.	Phase de fonctionnement normal du programme	18
8.5.3.	Configuration du capteur	18
8.5.4.	Mise en œuvre des moteurs brushless BLDC	18
8.5.5.	Exploitation des données du capteur	19
8.5.6.	Asservissement en position	19
8.5.7.	Architecture du programme	19
8.6.	Intégration.....	20
9.	Problèmes rencontrés	21
10.	État d'avancement du projet.....	21
11.	Perspectives d'évolution	22
12.	Conclusion	22
13.	Annexes	23
14.	Table des figures	24

1. Introduction

Ce document est un bilan de projet de fin d'études à Polytech Tours. Il a pour objectif de faire le lien entre les différents documents et livrables réalisés au cours de ce projet, d'exposer et justifier les choix qui ont été faits et d'effectuer un bilan sur le déroulement du projet.

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.

Le sujet du projet est la réalisation d'un système de stabilisation motorisé pour une caméra d'action sur trois axes.

1.1. Contexte 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 caméras, 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. C'est donc dans ce contexte que la MOA a proposé ce projet afin de trouver une solution à ce besoin.

1.2. Objectif

L'objectif de ce projet était 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. Cette stabilisation sera assurée par des effecteurs électriques. Son utilisation se devait simple et intuitive.

1.3. Existant

Certains projets ont vu le jour avec la prolifération des drones pourvus de caméras sur nacelles stabilisées. Le choix qui a été pris dans le cadre de ce projet est de considérer que le projet n'a pas d'existant ni matériel ni logiciel. En effet, dans le cadre pédagogique de ce projet, l'objectif était de se confronter à l'ensemble des problématiques mécaniques, électroniques et logicielles.

2. Cahier des charges

L'objectif du projet est de réaliser un système de stabilisation motorisé et autonome pour caméra. Usuellement appelé « Steady cam » pour « Steady camera », ce type de matériel est utilisé notamment dans le cinéma pour réaliser des travelings fluides en absorbant les mouvements parasites. Le système doit permettre d'assurer la stabilisation de caméras d'action type GoPro.

Contraintes du projet

- Le système doit être énergétiquement autonome
- La stabilisation doit se faire sur au moins deux axes : Roll + Pitch
- Effort de conception : Prototype avec un faible poids et encombrement
- L'utilisateur doit avoir la possibilité d'ajuster les paramètres d'amortissement

Livrables attendus

- Cahier de spécification
- Document d'analyse et modélisation
- Documentation de reprise de projet (Guide utilisateur, administrateur et développeur)
- Prototype fonctionnel

3. Problématique

Pour répondre aux spécifications qui ont été établies, nous avons établi la problématique suivante afin d'identifier les principales contraintes de ce système.

- **Énergétiquement autonome**

Le système doit être alimenté par batterie pour permettre à l'utilisateur de pouvoir l'utiliser sans contrainte d'alimentation secteur. Il est également nécessaire de prendre en compte le besoin important en courant pour l'alimentation de trois moteurs de façon simultanée.

- **Réactivité**

Destiné à stabiliser une caméra pour effectuer des prises de vue, le système doit être très réactif pour effacer les mouvements parasites induits par son utilisateur. Il est donc nécessaire d'avoir un traitement de l'information rapide, mais aussi d'avoir des effecteurs électriques suffisamment réactifs.

- **Évolutivité**

Dans le but de pouvoir modifier la structure mécanique pour l'adapter à un autre appareil de prise de vue, la partie électronique et la partie logicielle doivent être conçues pour s'adapter à ce changement. Cette contrainte implique un bon dimensionnement pour la partie électronique et une conception modulaire du programme afin de prévoir ces changements.

- **Encombrement**

Comme indiqué dans le cahier des charges, le système doit être le plus contenu possible afin de pouvoir être embarqué sur une moto, ou tenu à la main sans être une contrainte pour son utilisateur. Pour cela, un effort doit être fait sur la partie mécanique pour concevoir une structure la plus minimale possible. De plus, la partie électronique doit être parfaitement intégrée au système et respecter les mêmes dimensions que la structure.

- **Robustesse**

Enfin, la robustesse est un élément important à considérer. En effet, le produit doit être utilisable dans des conditions normales d'utilisation et doit être durable dans le temps. Cela implique une structure mécanique solide, des moteurs légèrement surdimensionnés ainsi qu'une protection des parties électroniques.

4. Conduite de projet

Pour mener à bien ce projet, nous avons dû sélectionner une méthode de gestion de projet. Ensuite, nous avons effectué un découpage des tâches pour réaliser un planning prévisionnel.

Notre choix de méthode s'est porté sur la méthode SCRUM des méthodes AGILE. En effet, étant sur des technologies que nous ne connaissions pas, le fait d'ajouter des fonctionnalités au système au fur et à mesure de l'avancement nous paraissait parfaitement adapté à ce projet.

4.1. Méthodologie

Comme présenté dans le livrable « Cahier de spécifications », nous avons organisé le projet en deux sous-projets. Cette décision a été faite après une étude de faisabilité au cours de laquelle nous avons identifié les risques. Le risque majeur de ce projet a été identifié comme l'obtention de l'orientation du capteur dans l'espace. En effet, aujourd'hui ces capteurs d'orientation intégrant un accéléromètre, un gyroscope et éventuellement un magnétomètre se trouvent dans tous les smartphones et tablettes du marché. Malgré le fait que ces capteurs soient très répandus, il y a encore très peu d'algorithmes Open Source permettant d'exploiter ces données. C'est pourquoi nous avons choisi dans un premier de nous consacrer à l'acquisition et au traitement des données sur un premier prototype. Puis, dans une deuxième phase du projet, après validation du fonctionnement de ces fonctionnalités de base,

nous avons prévu de créer le prototype final, répondant à l'ensemble de la problématique et des exigences exprimées dans le cahier des charges.

- **Prototype initial**

Pour commencer le projet, nous avons choisi de nous concentrer sur la prise en main du capteur et le traitement associé en mettant de côté la commande des moteurs brushless. Pour effectuer des tests, nous avons mis en place un prototype équipé de trois servomoteurs de modélisme pour remplacer les moteurs. En effet, la commande de ce type d'effecteur est grandement simplifiée par l'utilisation des bibliothèques Arduino.

Ce premier prototype a permis de mettre en œuvre :

- La configuration du capteur
- De comprendre et d'implémenter un algorithme permettant d'effectuer la calibration du capteur
- D'implémenter un algorithme de filtre PID
- De tester plusieurs techniques de fusion de données d'un tel capteur
- De valider le fonctionnement du prototype sur deux axes : Roll et Pitch

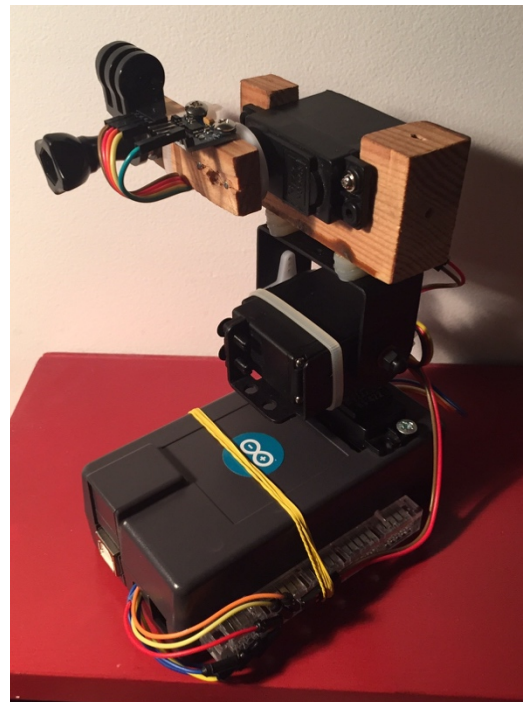


Figure 1 : Prototype initial

Après la validation de fonctionnalités, nous avons réalisé des vidéos pour valider le fonctionnement auprès de la MOA :

- Prototype initial sur deux axes : <https://www.youtube.com/watch?v=oEpsLbolnE>
- Ajout de l'axe Z (Yaw) : <https://www.youtube.com/watch?v=FzccmuE41do>
- Prototype initial sur trois axes : https://www.youtube.com/watch?v=J3hMNFk_600

Dans cette première partie, nous avons également réalisé un banc d'essai moteur pour prendre en main la commande de ce type de moteur brushless et implémenter la fonction permettant la commande.

Pour réaliser la commande du moteur, nous avons utilisé une Breadboard intégrant le composant L6234 afin de concentrer le travail sur l'implémentation de la fonction au niveau du microcontrôleur. La partie électronique sera mise en œuvre pour le prototype final.

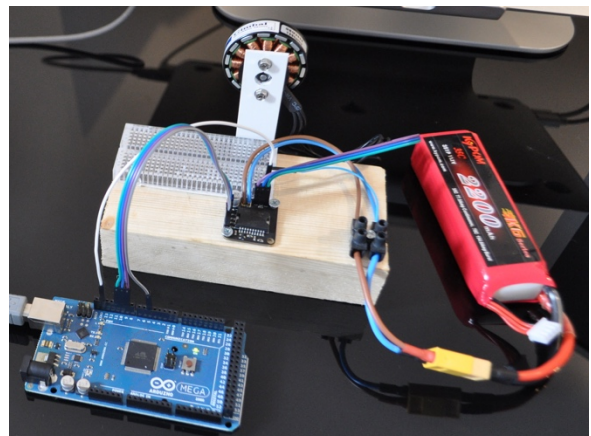


Figure 2 : Banc d'essai moteur

Le fonctionnement de la commande d'un moteur brushless a également été validé avec un vidéo comme livrable. Celle-ci montre le fonctionnement du moteur avec variation de la vitesse : https://www.youtube.com/watch?v=QIONZgU_zDE

À la fin de cette première partie du projet, nous avons mis en œuvre l'ensemble des techniques nécessaire à la réalisation du prototype final.

- **Prototype final**

La seconde partie du projet a pour objectif de capitaliser les connaissances et techniques développées lors de la première partie du projet pour les intégrer dans un prototype final. Dans cette partie du projet, nous avons donc toutes les connaissances nécessaires à la conception et à la réalisation du système.

Lors de cette phase, nous avons effectué la conception de la carte électronique, la conception mécanique et le code optimisé pour le microcontrôleur sélectionné pour le prototype final.

4.2. Découpage en tâches

Le découpage en tâches a été réalisé dans la partie 8 du livrable « Cahier de spécifications ».

4.3. Choix stratégiques

Au cours du projet, nous avons dû effectuer des choix techniques qui sont liés à la gestion de projet.

- **Choix de l'algorithme de fusion des données du capteur**

Pour obtenir l'orientation d'un solide dans l'espace il existe plusieurs types de capteurs. Dans le cadre de ce projet, nous avons établis lors de la phase de spécification d'utiliser un accéléromètre, un gyroscope et un magnétomètre. Pour obtenir l'orientation du solide, il est nécessaire de fusionner ces valeurs afin d'obtenir par exemple une représentation d'Euler. Lors de nos recherches, nous avons observé qu'il existe essentiellement deux méthodes pour obtenir l'orientation avec le capteur MPU-9250 :

- Soit utiliser le DMP (processeur spécialisé dans le calcul intégré au capteur). Cette Solution permet d'obtenir de manière simple et précise l'orientation du capteur.
- Soit mettre en œuvre un algorithme au niveau du microcontrôleur en récupérant les valeurs brutes des différents capteurs. Cette solution nécessite l'utilisation d'un algorithme mathématique déjà existant, car cela requiert de grandes compétences très spécialisées.

La solution avec l'utilisation du DMP n'a pas été retenue du fait de devoir flasher à chaque démarrage du système le code binaire d'un programme propriétaire dans le DMP. En effet, nous avons comme volonté de maîtriser l'ensemble des fonctions que nous utilisons pour ce projet. Nous avons donc choisi d'implémenter l'algorithme de Madgwick au niveau du microcontrôleur. Nous verrons par la suite que ce choix n'a pas permis d'aboutir à une solution pleinement satisfaisante en terme de performances.

- **Choix sur la structure mécanique**

De nombreuses structures avec moteur brushless intégrés sont en vente dans des boutiques spécialisées ou sur internet. Ces Structures sont destinées à être utilisées sur des drones pour stabiliser des caméras d'action. Ce type de matériel aurait très bien pu convenir pour ce projet. Néanmoins, ayant des connaissances en conception mécanique, nous avons comme volonté de réaliser nous même l'ensemble dans la partie mécanique.

- **Choix sur la partie électronique**

Comme pour la partie mécanique, le marché des drones a fait apparaître des cartes électroniques intégrant un microcontrôleur, des drivers de moteurs brushless sur une même carte pour un encombrement et un prix très réduit. Là encore, nous souhaitons réaliser nous même notre propre carte électronique afin de nous confronter aux différentes problématiques induites par ce type de projet.

4.4. Planification

La gestion du planning établie en début de projet (voir cahier de spécification, partie 10) a été suivie à quelques exceptions en fin de projet en raison de tests non satisfaisants sur une tâche du projet qui nous a contraints à revenir sur l'implémentation d'une fonction. Certaines tâches non prioritaires n'ont donc pas pu être réalisées :

- Réalisation du code natif pour la fonction « Commande de l'orientation »
- Intégration du code natif pour le prototype final

Nous avons environ 120h dédiées sur notre emploi du temps pour la réalisation du PFE. Pour mener à bien ce projet, nous avons consacré entre 470 et 500 heures ce qui totalise un minimum de 450 heures sur le temps personnel. Cet investissement personnel nous a permis de ne pas prendre de retard sur le planning que nous nous étions fixé.

4.5. Livrables du projet

Ce rapport est accompagné de plusieurs livrables détaillant les spécifications et l'analyse qui a été faite pour ce projet. Les guides doivent permettre une reprise du projet par niveau d'utilisateur.

- **Cahier de spécifications**

Ce document a pour objectif de définir les besoins, l'environnement et les objectifs pour la réalisation du projet. Il contient le découpage des tâches ainsi que la planification.

- **Document d'analyse et modélisation**

Ce document a pour objectif d'expliquer en détail les choix et solutions mises en place pour la réalisation du projet de fin d'études. Ce rapport est organisé en plusieurs sous-parties : Électronique, mécanique et logicielle.

- **Guide de l'utilisateur**

Ce document est destiné à l'utilisateur du système. Il décrit les démarches à suivre pour l'utilisation du système.

- **Guide l'administrateur**

Ce document est destiné à l'administrateur du système. Il décrit les démarches à suivre pour la configuration du système. Les configurations présentées dans ce document doivent être effectuées par l'administrateur pour préparer le système après sa fabrication. Ce paramétrage est à effectuer une seule fois dans la vie du système. Ce document peut être également utile aux développeurs.

- **Guide du développeur**

Ce document est destiné à décrire l'architecture du code et présente les différents codes sources associés au projet.

- **Code source du projet**

Ce projet est livré avec le code source et son environnement nécessaire à la cross-compilation.

- **Codes exemples associés au projet**

L'ensemble des codes sources des programmes utilisés au cours de la vie du projet est également inclus dans les livrables du projet. Ils sont décrits dans le guide du développeur.

5. Schéma général du système

Le schéma ci-dessous permet de définir le nom des axes que nous avons utilisé dans l'ensemble des rapports et documents fournis. Il permet également de définir le repère géométrique.

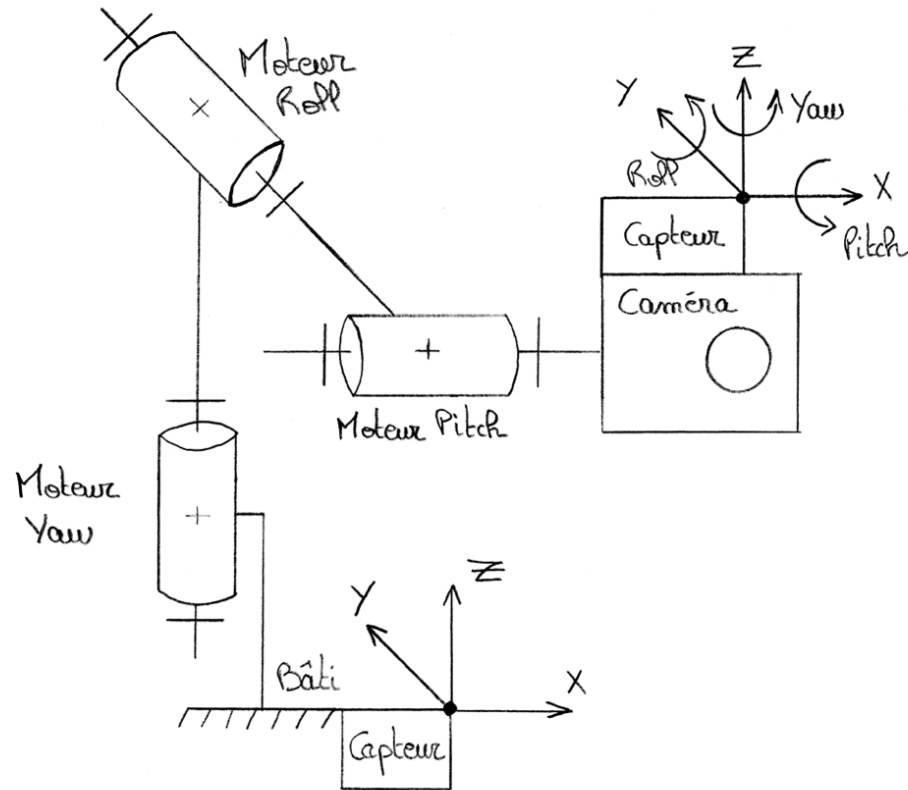


Figure 3 : Schéma général du système

6. Architecture mise en œuvre

Pour réaliser la fonction de stabilisation de la caméra, nous avons découpé le système en trois sous-parties : Captage, Traitement et partie opérative.

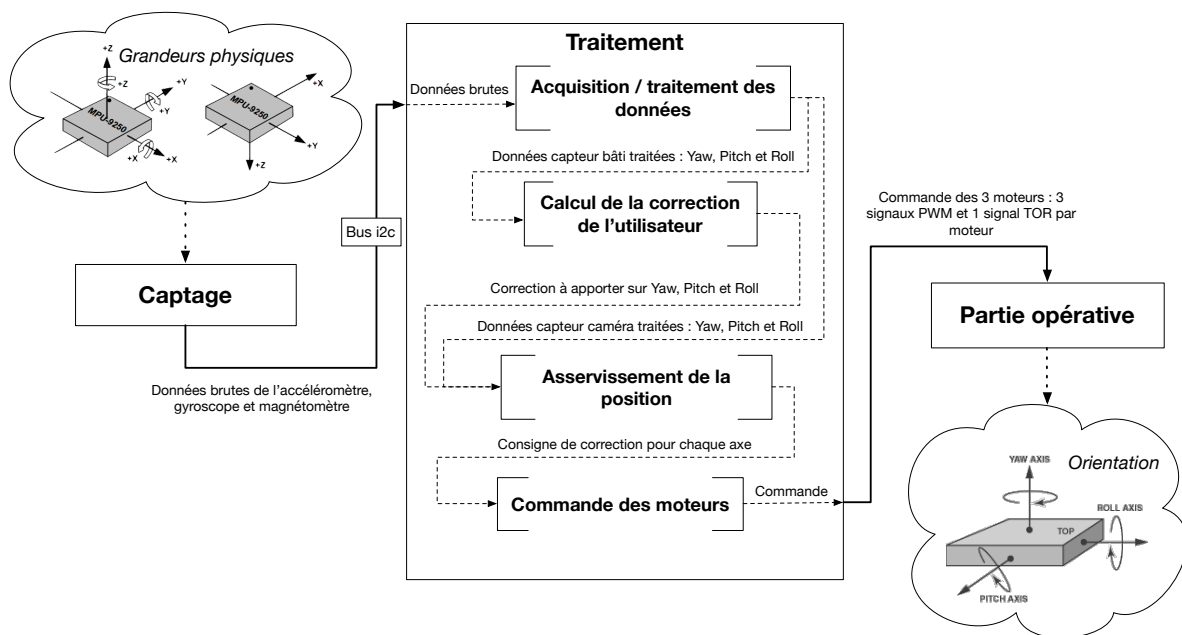


Figure 4 : Architecture du système

- **Captage** : Sous-ensemble permettant de fournir au système les données brutes nécessaires à la modélisation de l'orientation de la caméra et du système dans un espace 3D. C'est grâce à l'orientation de la caméra que nous effectuons l'asservissement en position.
- **Traitement** : Sous-ensemble permettant de réaliser plusieurs fonctions : Traitement des données, calcul de la correction de l'utilisateur, asservissement en position et commande des moteurs.
 - **Fonction « Acquisition / traitement des données »** : Permet d'obtenir l'orientation exacte de la caméra et du bâti à partir des données brutes envoyées par le sous-ensemble « Captage ».
 - **Fonction « Calcul de la correction de l'utilisateur »** : Cette fonction permet de calculer la correction à apporter pour modifier l'orientation de la caméra en fonction de celle du bâti.
 - **Fonction « Asservissement de la position »** : Permet de calculer la correction à appliquer sur le sous-ensemble « Partie opérative » pour corriger l'orientation de la caméra en fonction des données produites par la fonction « Traitement des données ».
 - **Fonction « Commande des moteurs »** : Permet d'effectuer la commande des moteurs en fonction de la consigne calculée par la fonction « Asservissement de la position ».
- **Partie opérative** : Sous-ensemble constitué d'une structure mécanique et de moteurs permettant d'orienter la caméra dans un espace 3D.

7. Matériel utilisé

Le choix et la justification des composants du système a été réalisé dans le livrable « Analyse et modélisation » dans la partie 4.1. La liste de matériel qui suit concerne le matériel nécessaire à la réalisation du prototype final. Au cours du projet, un premier prototype a été réalisé qui sera décrit dans la partie gestion de projet.

Le tableau ci-après fait une synthèse du matériel qui a été nécessaire à la réalisation électronique.

Désignation	Quantité	Prix unitaire TTC
Capteur MPU-9250	1	11,90€
Moteur brushless BLDC GBM3508-130	3	24,43€
Microcontrôleur Teensy 3.1	1	19,80€
Batterie LIPO 3S 2200 mAh	1	23,90€
Régulateur LM2574	3	7,80€
Support DIP20	3	1,03€
Condensateur chimique 22 uF	1	0,50€
Condensateur chimique 220 uF	1	0,80€
Condensateur céramique 100 nF	3	0,10€
Condensateur chimique 100 uF	3	0,50€
Condensateur céramique 220 nF	3	0,80€
Diode Schottky MBR160G	1	0,20€
Bobine 330 uH	1	1,00€
Condensateur céramique 10 nF	3	0,10€
Condensateur céramique 1 uF	3	0,10€
Diode 1N4148	6	0,10€
PCB Eurocircuit	1	71,91€
Interrupteur 2 positions	1	2,90€
Total TTC		238,39€

Le tableau ci-après constitue la la liste du matériel pour la réalisation de la partie mécanique.

Désignation	Quantité	Prix TTC
Plat aluminium brut (Longueur 1 m – Largeur 35 mm)	1	7,45€
Plat aluminium brut (Longueur 1 m – Largeur 20 mm)	1	5,90€
Vis M3 15 mm à tête cylindrique + écrous + rondelles	4	2,00€
Vis M3 5 mm à tête plate	9	2,00€
Fixation GoPro « The Frame »	1	9,90€

Coût total du projet en matériel : 265,64€

8. Conception et réalisation

8.1. Conception de la structure

Afin de réaliser la structure mécanique, nous avons réalisé dans un premier temps un schéma permettant de définir les éléments de la structure.

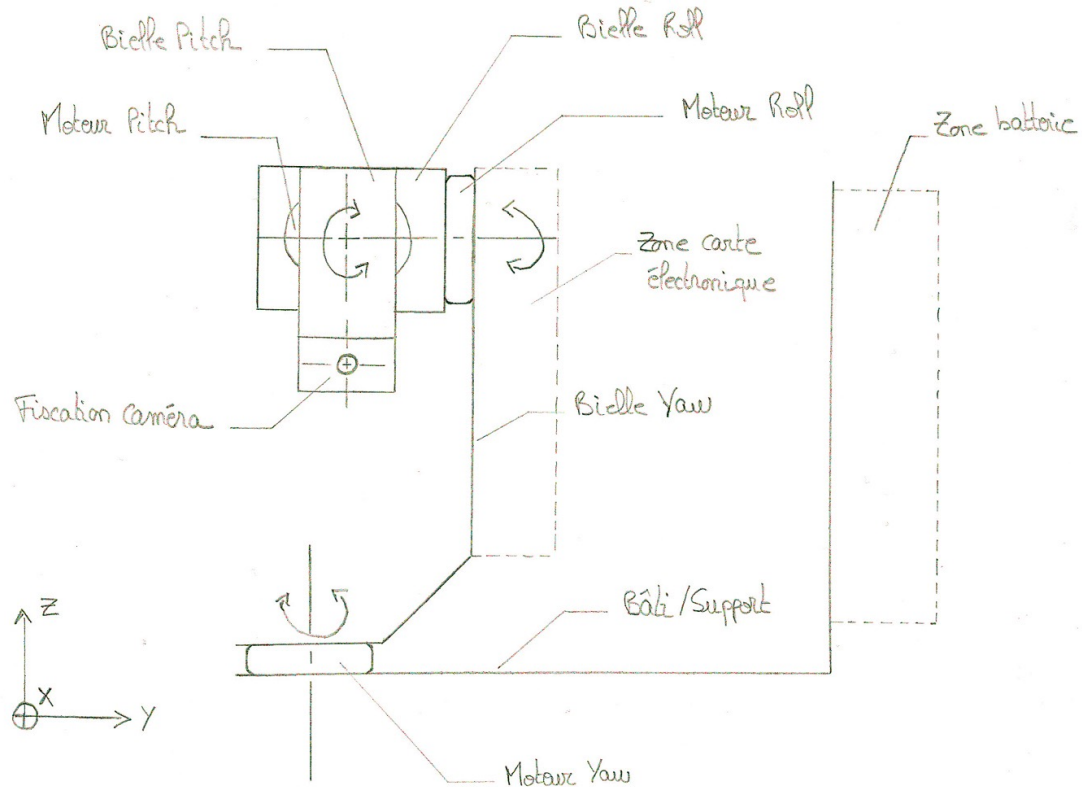


Figure 5 : Schéma de la structure mécanique

Nous avons ensuite entrepris une phase de conception sur ordinateur à l'aide du logiciel SolidWorks pour obtenir un 3D. Cela a également permis de définir les dimensions exactes de chaque pièce et de s'assurer qu'il n'y a pas de collisions entre les différents éléments. Les plans de l'ensemble de la structure se trouvent en annexe du document « Analyse et modélisation ».

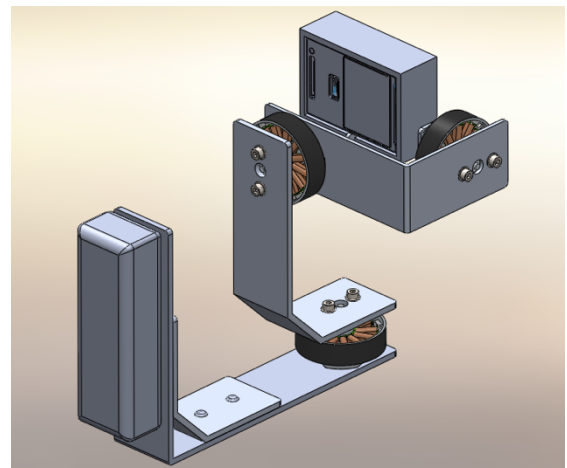
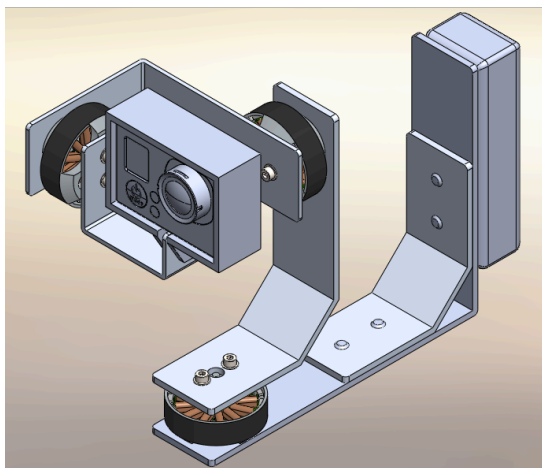


Figure 6 : Rendu 3D structure mécanique

8.2. Conception des éléments de protection

Afin de protéger la carte électronique, nous avons réalisé un boîtier de protection. Nous avons effectué la conception sur ordinateur, puis réalisé les pièce grâce à une imprimante 3D.

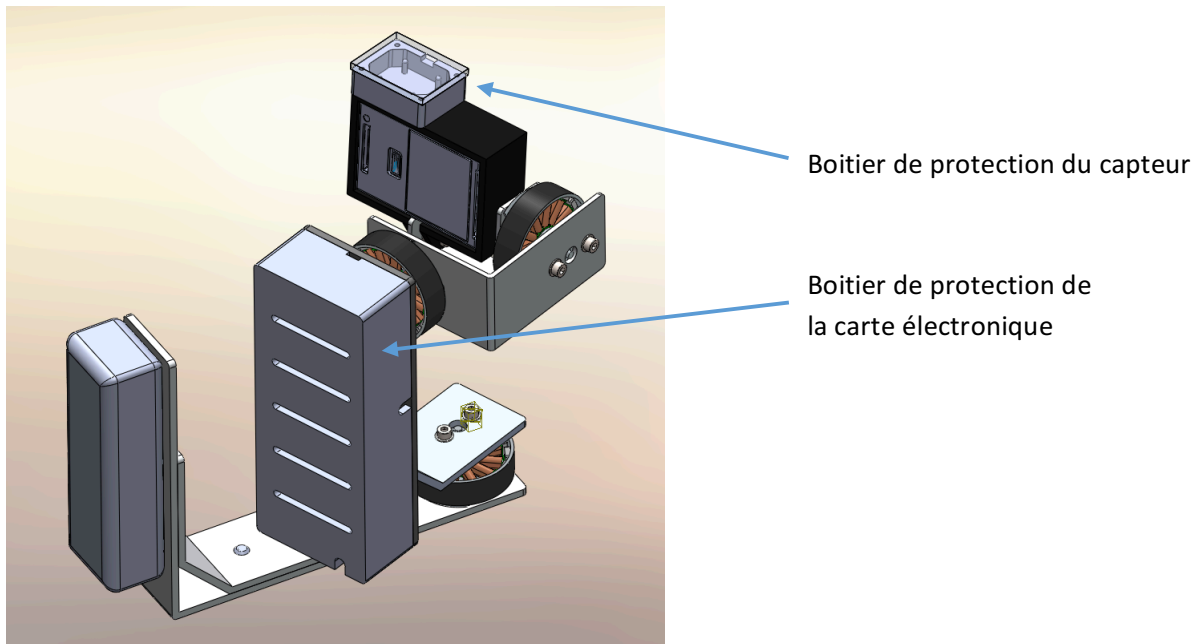


Figure 7 : Éléments de protection

8.3. Réalisation de la structure

Pour réaliser la structure mécanique, nous nous sommes appuyé entièrement sur les plans que nous avons réalisé pour chaque pièce grâce à la modélisation 3D. En procédant ainsi, nous avons l'assurance de réaliser les pièces avec les bonnes dimensions.

Cependant, lors de la modélisation nous n'avions pas les compétences nécessaires pour prendre en compte le poids de la caméra pour calculer le point d'équilibre de la structure.



Figure 8 : Réalisation de la structure

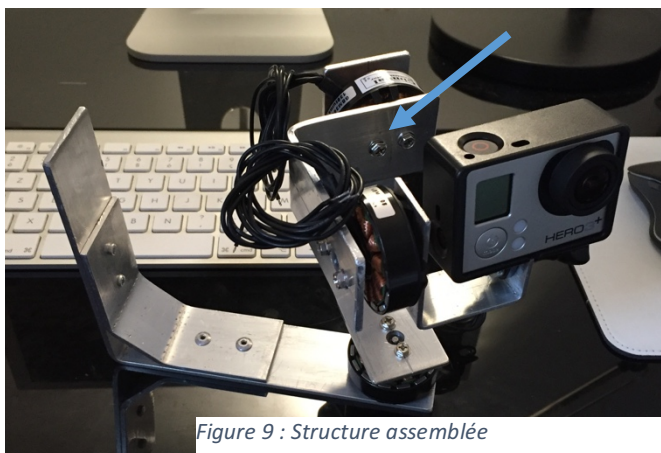


Figure 9 : Structure assemblée

En effet, comme on le voit sur l'illustration ci-contre, le système est en équilibre, même sans l'aide des moteurs. Cela permet d'éviter aux moteurs d'appliquer un effort lorsque le système est parfaitement à l'horizontal.

Nous avons donc dû adapter la dimension de la bielle pointée par la flèche sur l'illustration afin d'obtenir le résultat suivant.

8.4. Partie électronique

La conception de la partie électronique a été réalisée à partir des schémas fonctionnels réalisés dans le document « Analyse et modélisation » à la partie 4.2. Ci-dessous on retrouve l'ensemble des fonctions électroniques nécessaires au système :

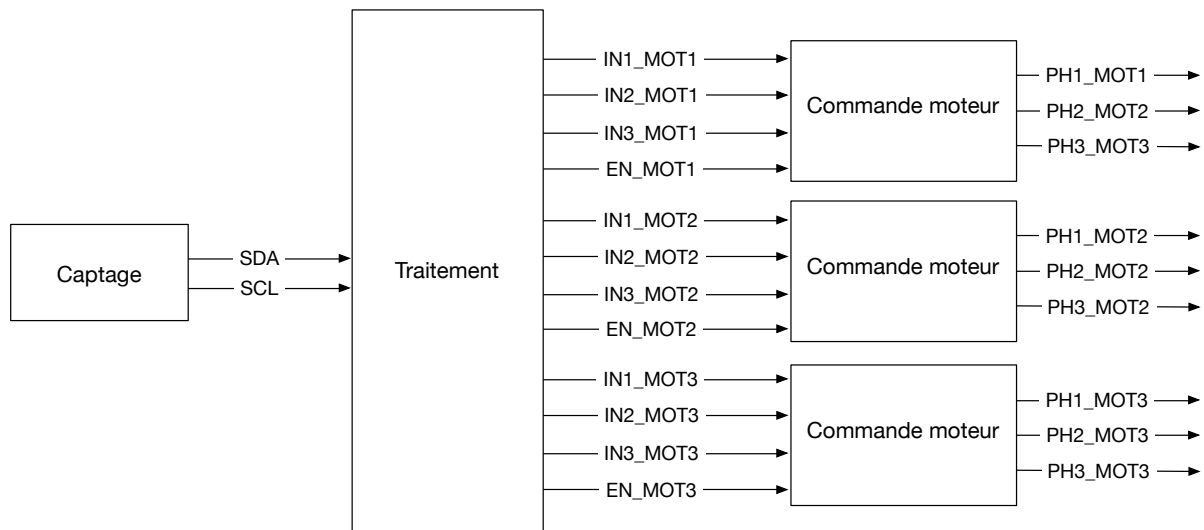


Figure 10 : Partie électronique - Schéma fonctionnel général

Nous avons réalisé le schéma de principe et conçu le PCB avec le logiciel Multisim. Sur le schéma ci-dessous, on retrouve les différents blocs fonctionnels identifiés avec le schéma de principe :

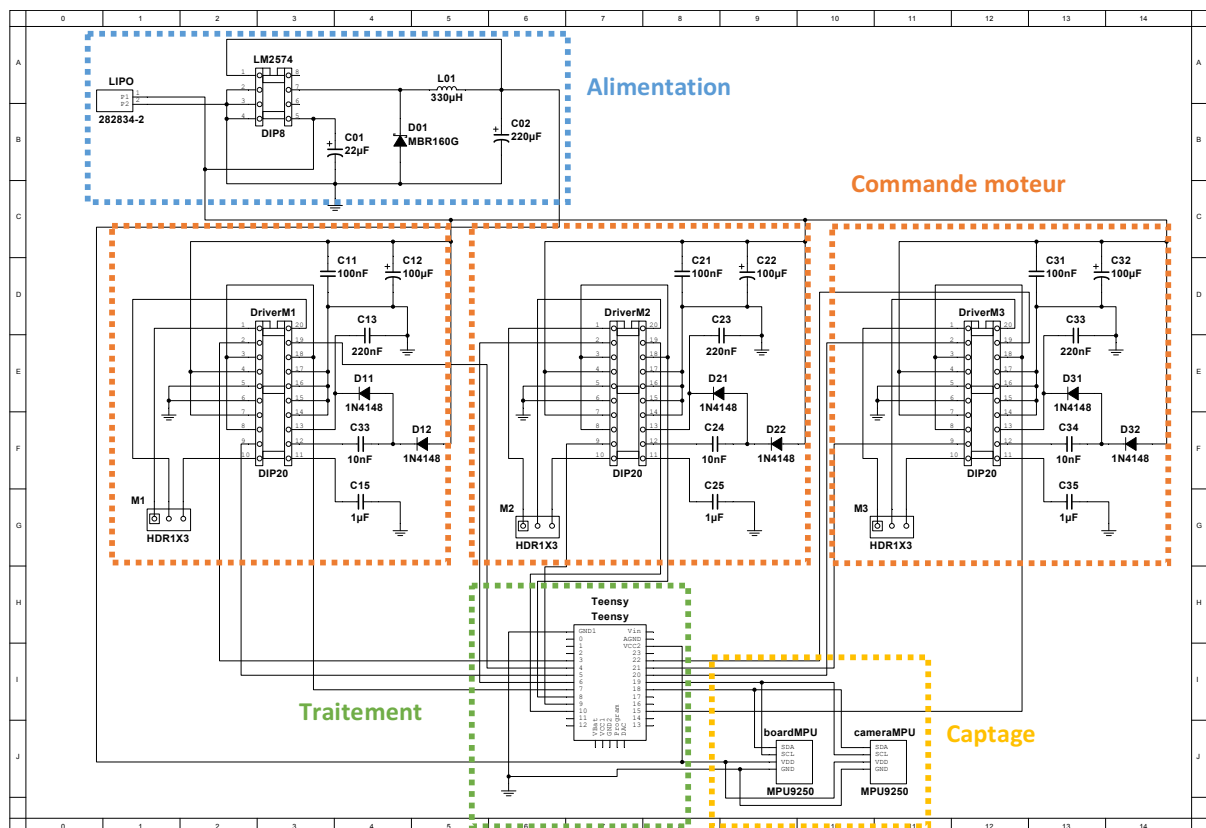


Figure 11 : Partie électronique - Schéma de principe

Pour valider cette conception que nous avons réalisée en nous appuyant sur les datasheet des composants, nous avons utilisé une platine de développement reproduit chaque fonction.

Ainsi, nous avons pu valider chaque fonction électronique indépendamment en réalisant un programme de test sur le microcontrôleur. Après la validation de chaque fonction, nous avons effectué un test avec l'ensemble des fonctions pour nous assurer qu'il n'y a pas d'effet de bords.

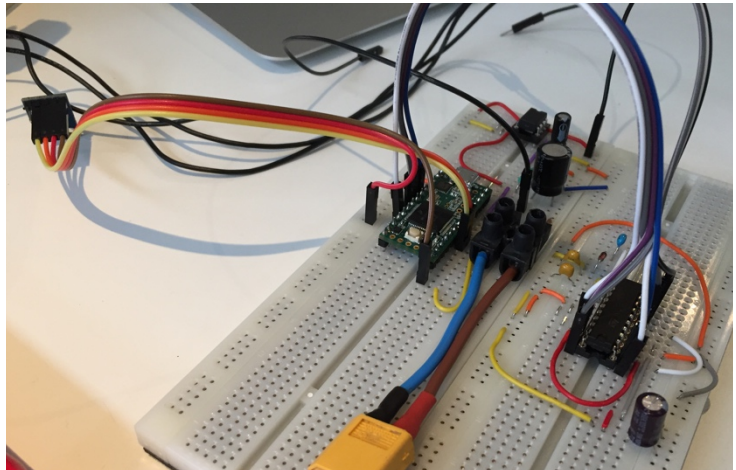


Figure 12 : Conception électronique - Tests

La contrainte principale pour la conception de la carte électronique était l'encombrement. En effet, nous devons contenir au maximum les dimensions du PCB pour qu'il puisse être fixé sur le système sans gêner l'utilisateur.

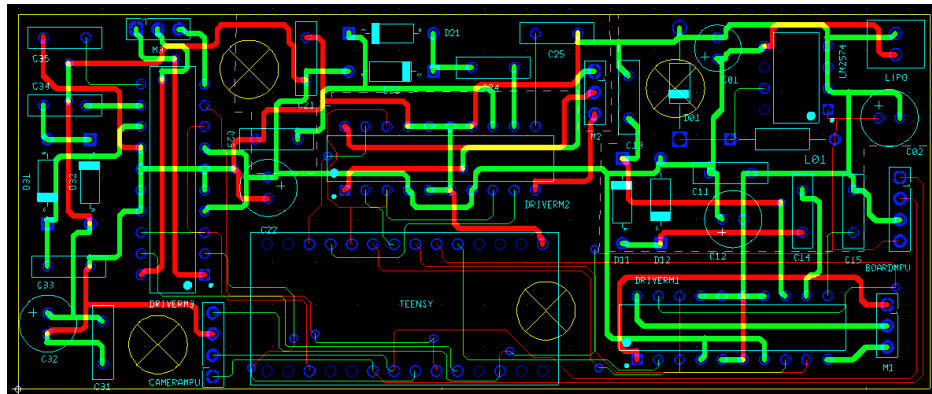


Figure 13 : Partie électronique - Conception du PCB

Le PCB a été réalisé en 2 couches pour permettre un gain de place. Toujours dans une démarche d'optimisation, nous avons utilisé deux tailles de piste :

- Les pistes de données : 0,254 mm
- Les pistes de puissance : 0,762 mm

Afin de limiter les perturbations électromagnétiques entre les pistes de puissance et les pistes de données, nous avons créé des zones « keep-out » pour éloigner au maximum de deux types de pistes.

Le placement des composants a également nécessité un travail d'optimisation pour trouver l'arrangement qui permet un encombrement minimum tout en limitant les proximités de certaines pistes et en rapprochant au maximum les condensateurs de découplages des composants concernés.

Le PCB a été réalisé par la société EURO circuits. Afin de supporter les forts courants induits par les moteurs, nous avons spécifié une épaisseur de piste à 35µm pour supporter les forts courants.

Figure 14 : Partie électronique - PCB

L'ensemble des tests ont été réalisés avec succès.

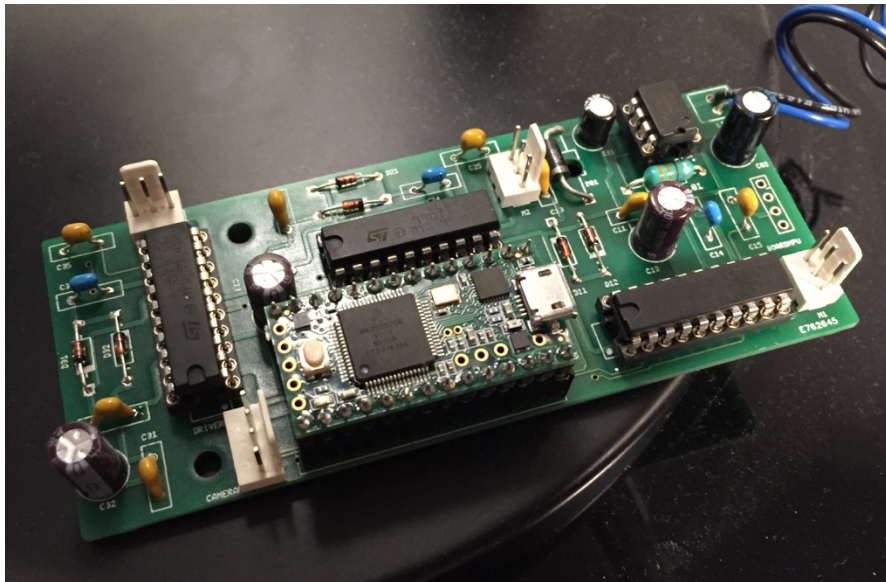


Figure 15 : Partie électronique - Carte électronique assemblée

Le programme du microcontrôleur est organisé en deux partie :

- Une phase d'initialisation permet de configurer le matériel, charger les paramètres et éventuellement d'effectuer l'étalonnage du capteur. Cette phase est décrite dans le document « Analyse et modélisation » dans la partie 6.5.1.
- La phase de fonction normal permet d'effectuer la stabilisation de la caméra. Le processus est constitué d'un ensemble de fonctions qui s'enchainent de façon séquentielle. Cette phase est décrite dans le document « Analyse et modélisation » dans la partie 6.5.2.

8.5.1. Phase d'initialisation du programme

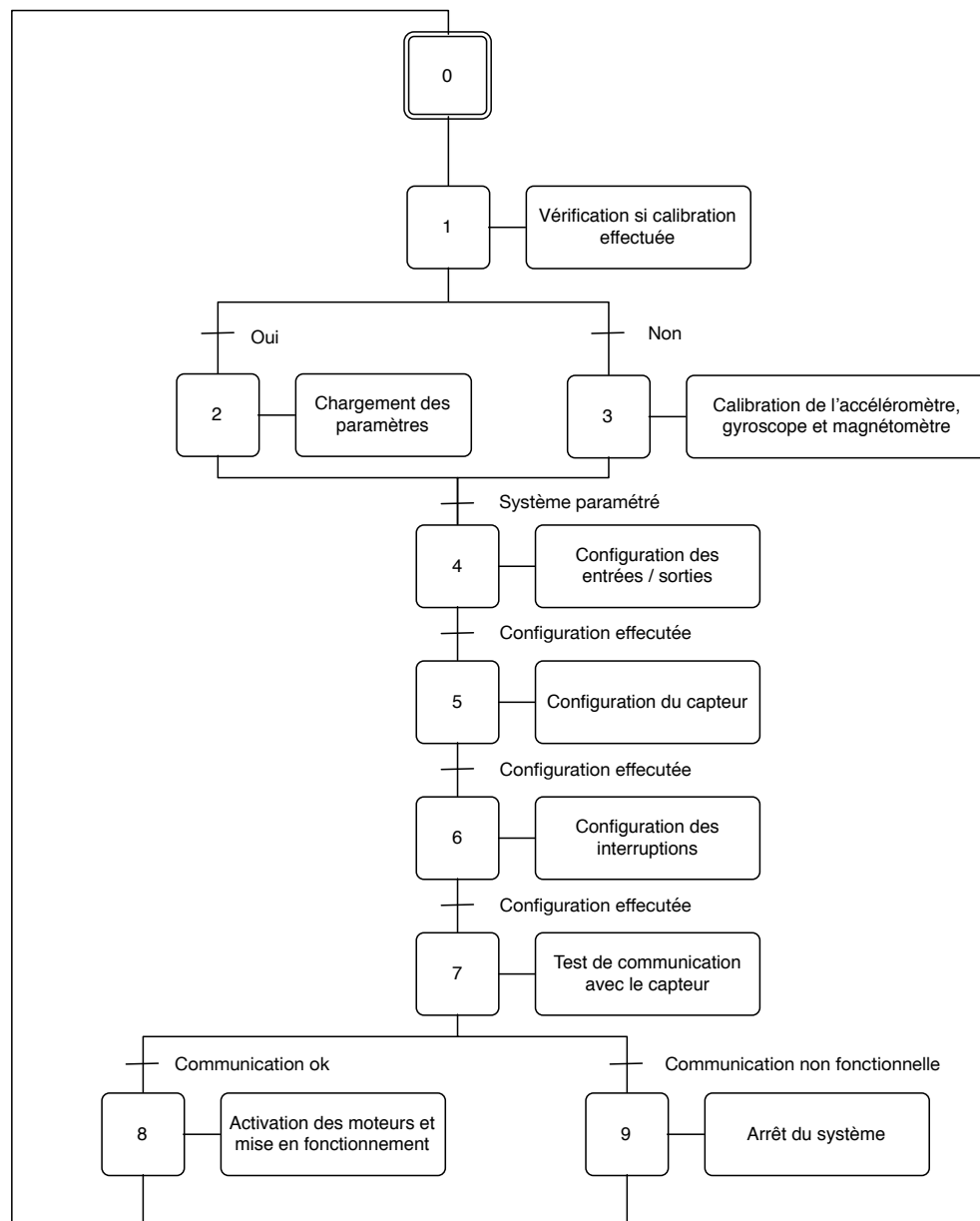
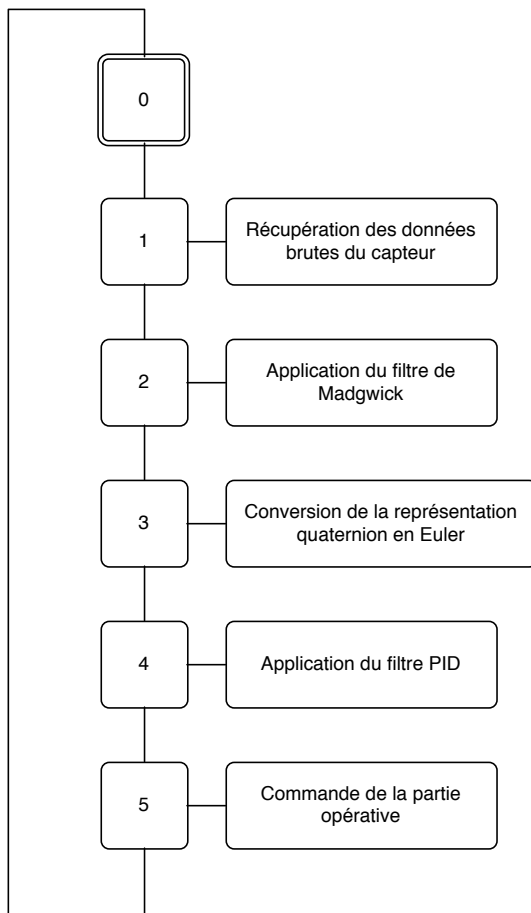


Figure 16 : Partie logicielle - Phase d'initialisation

8.5.2. Phase de fonctionnement normal du programme



Le processus de fonctionnement normal du programme du microcontrôleur peut être décrit par le grafcet ci-contre. Les fonctions représentées s'enchaînent successivement en partant de la récupération des données brutes du capteur jusqu'à la commande de la partie opérative. Cependant, les opérations sont cadencées par des interruptions avec les fréquences suivantes :

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

Pour plus de détails sur le cadencement de ces interruptions, se référer à la partie 6.5.2 du livrable « Analyse et modélisation ».

Figure 17 : Partie logicielle - Fonctionnement normal

8.5.3. Configuration du capteur

Durant ce projet, une des tâches qui nous aura pris le plus de temps est la configuration du capteur. En effet, le capteur MPU-9250 est très complexe et dispose d'un grand nombre de registres pour effectuer sa configuration. De plus, sa configuration fait appel à des notions de physique que nous ne maîtrisons pas en début de projet, et qui restent relativement abstraites malgré un important travail de recherche. Dans la partie 6.4.1 du livrable « Analyse et modélisation », nous avons décrit avec précision les registres du capteur à configurer et la démarche à adopter pour mettre en œuvre le bon fonctionnement de ce capteur. La calibration est également un point très important pour obtenir des valeurs fiables. Nous avons également expliqué les étapes à effectuer pour correctement calibrer le capteur dans le document d'analyse dans la partie 6.4.3.1 et 6.4.3.2.

8.5.4. Mise en œuvre des moteurs brushless BLDC

Ce type de moteur est particulièrement intéressant dans son fonctionnement. Il permet un positionnement extrêmement précis (résolution de $0,1^\circ$) sans utiliser de codeur ou de capteur à effet hall. Ainsi, on peut effectuer un asservissement très précis avec ces moteurs avec une grande réactivité ce qui correspond parfaitement à la problématique du système. La mise en œuvre du programme a été facilitée par de nombreux exemples trouvés sur internet. De plus, l'algorithme permettant de commander ce type de moteur est relativement simple en utilisant un « lookup-table ». Cette table contient l'ensemble des trinômes de valeurs des signaux PWM à appliquer aux entrées du moteur. Ainsi, pour faire tourner le moteur, il suffit de se déplacer dans la table.

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 degré. Pour augmenter la précision, il suffit de générer plus de valeurs (avec une limite de 255 valeurs : résolution des PWM configurées sur 8 bits).

Pour éviter les perturbations sonores liées à la fréquence des signaux PWM, nous avons augmenté la fréquence à 18 kHz, ce qui est inaudible à l'oreille humaine. Dans certains exemples observés lors de nos recherches sur internet, la fréquence était fixée à 32 kHz. Nous avons jugé qu'une fréquence plus basse produirait le même effet en terme d'atténuation des nuisances sonores. De plus, une fréquence de fonctionnement moins élevée permet d'éviter l'échauffement des moteurs et par conséquent, la consommation énergétique du système.

8.5.5. Exploitation des données du capteur

Afin d'obtenir l'orientation du capteur dans l'espace, nous avons sélectionné la solution qui est d'utiliser l'algorithme de Madgwick. Cet algorithme, créé par un étudiant de l'université de Bristol, permet la fusion des données d'un accéléromètre, du gyroscope et d'un magnétomètre pour obtenir l'orientation sous la forme d'un quaternion. Pour effectuer l'asservissement de notre système, nous utilisons la représentation d'Euler. Une formule mathématique permet de passer d'un quaternion à Euler.

Cet algorithme Open Source est utilisé dans notre programme comme une boîte noire à laquelle on transmet les valeurs brutes du capteur. En retour, nous obtenons un quaternion qui peut être exploité par la suite en passant à la représentation d'Euler.

Pour plus d'informations sur l'algorithme de Madgwick :

<http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/>

8.5.6. Asservissement en position

L'asservissement en position est réalisé par un filtre PID. L'algorithme est implémenté sous forme de fonction et s'appuie sur les paramètres Kp, Ki, Kd propre à chaque axe. Ces paramètres sont ajustables soit dans le programme ou bien utilisant le programme de réglage. La modification de ces paramètres est décrite dans le document « Guide de l'administrateur ».

8.5.7. Architecture du programme

L'architecture du programme est décrite dans le livrable « Guide du développeur ». Pour répondre aux à la problématique d'évolutivité de la solution, nous avons construit le programme pour qu'il soit le plus paramétrable possible. Ainsi, il est possible de modifier la taille des moteurs (nombre d'enroulements) ainsi que la géométrie de la structure métallique en adaptant les paramètres. Chaque paramètre est également décrit dans le livrable « Guide l'administrateur ».

Le code source du projet est disponible en annexe du document par l'intermédiaire d'un lien Github.

8.6. Intégration

La phase d'intégration a consisté à intégrer la partie électronique à la structure mécanique. L'assemblage des deux parties a été conçu par CAO afin que les points d'assemblage soient parfaitement identiques.

Sur l'illustration ci-contre, on peut voir les différents points de fixations de la carte sur la structure qui correspondent aux perçages qui avaient été prévus sur le PCB. Le fond du boîtier de protection est intercalé entre la structure métallique et le PCB.

Les moteurs sont branchés sur la carte grâce à des connecteurs afin que l'ensemble soit maintenable facilement en cas de changement de moteur.

Le capteur est fixé derrière la caméra et également relié à la carte par un connecteur 4 broches pour simplifier un éventuel remplacement de la carte ou du capteur. Nous avons utilisé une nappe avec 4 fils non torsadés pour éviter les perturbations électromagnétiques sur le bus i2c.

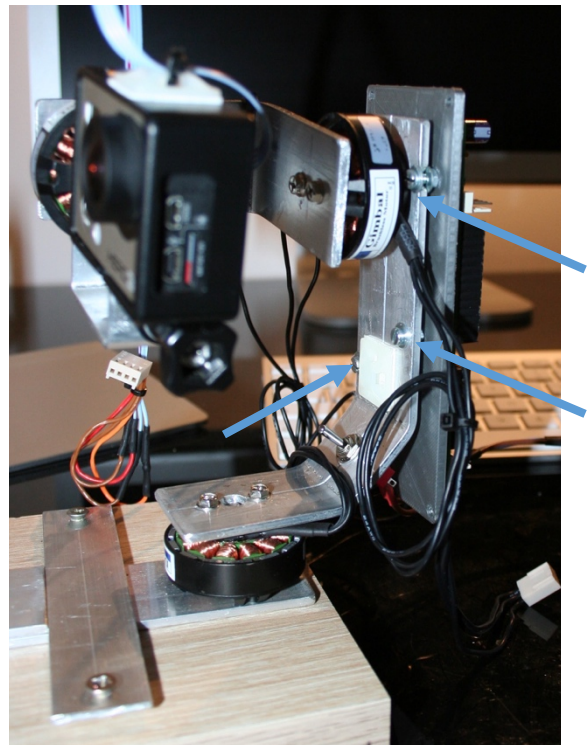


Figure 18 : Intégration - Assemblage du prototype

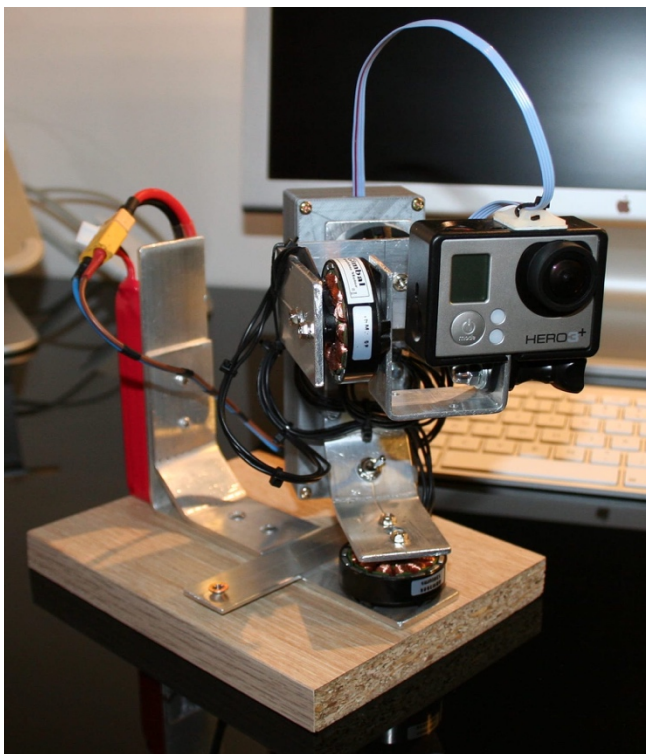


Figure 19 : Intégration - Assemblage du capot de protection

Enfin, le boîtier de protection est maintenu par quatre vis, facilement accessibles, toujours dans le but de faciliter la maintenance.

La batterie est maintenue sur le système via une bande de scratch, ce qui est couramment utilisé dans le domaine du modélisme pour fixer ce type de batterie.

L'interrupteur de mise sous tension est fixé sur la structure métallique, au plus proche de l'arrivée de l'alimentation pour limiter la longueur de câble et l'encombrement.

Pour faciliter les tests et les réglages, nous avons réalisé un support permettant de maintenir le système tel qu'illustré ci-contre.

9. Problèmes rencontrés

- **Charge de travail**

Comme évoqué dans la partie « Planification », la charge de travail requise pour la réalisation de ce projet était trop importante vis-à-vis des heures allouées au PFE sur notre emploi du temps. De plus, nous avions régulièrement des créneaux de deux heures prévues sur l'emploi du temps ce qui ne permet pas de travailler efficacement sur son projet. En effet, sur des projets comme celui-ci avec une contrainte matérielle nécessite une installation et une certaine logistique qui n'est pas compatible avec un créneau de deux heures.

Si c'était à refaire, je reverrais mes ambitions à la baisse, et je proposerais un système fonctionnant sur seulement deux axes. En effet, d'un point de vue du traitement des données d'orientation, la complexité entre un traitement des données sur deux axes varie énormément.

- **Complexité de l'algorithme de fusion**

Malgré le fait que le prototype soit aujourd'hui fonctionnel, la partie traitement des données issues du capteur ne permet pas d'obtenir l'orientation de la caméra de façon satisfaisante. Les valeurs varient et sont instables, ce qui rend l'asservissement en position instable également.

Si c'était à refaire, je choisirais d'utiliser le DMP malgré le fait que l'on doit flasher un code binaire dont on n'a aucune idée de son fonctionnement. Cependant, cette solution permettrait au prototype de fonctionner parfaitement et d'obtenir une stabilisation idéale grâce à des données fiables.

- **Perturbations électromagnétiques**

Lors des tests effectués sur la carte électronique, nous avons testé l'ensemble des fonctions de façon unitaire. Chaque test a été effectué avec succès. Néanmoins, nous nous sommes aperçus d'un dysfonctionnement lorsque nous faisons l'acquisition des données du capteur en même temps de la commande des moteurs. Ce dysfonctionnement était dû à des perturbations électromagnétiques induites par la commande des moteurs. En effet, nous utilisons trois signaux PWM pour la commande d'un moteur. Ayant sur la carte trois moteurs, cela fait au total neuf signaux PWM à une fréquence de 18 kHz qui émettent en même temps. Cela avait pour conséquence de perturber le bus i2c, ce qui rendait la communication avec le capteur impossible.

Pour résoudre ce problème, nous avons effectué des tests à l'aide d'un oscilloscope pour observer les trames sur le bus i2c. Ces observations n'ont révélé rien de probant. Nous avons fait l'hypothèse que des pics de courant pouvaient perturber le bus. Nous avons effectué des tests en ajoutant des résistances en série sur le bus et avons constaté une amélioration permettant de faire fonctionner l'acquisition des données en même temps que la communication.

10. État d'avancement du projet

Le projet s'est réalisé sur une durée de 5 mois et a abouti à un prototype fonctionnel. Cependant, comme expliqué dans la partie « Problèmes rencontrés », la fonction de traitement des données ne permet pas d'obtenir des résultats satisfaisants.

Le système répond aux spécifications établies en début de projet et peut aboutir rapidement à un prototype pleinement fonctionnel en terme de stabilisation de caméra. En effet, seule la fonction de traitement des données est à revoir. Nous avons pu tester l'ensemble des fonctions au cours du développement et avons validé le fonctionnement de chacune.

Il reste cependant une fonction à implémenter qui est la correction de l'orientation de la caméra en fonction des mouvements de l'utilisateur (fonction décrite dans le cahier de spécification dans la partie 4.2). Cette fonction est purement algorithmique et ne nécessite pas de connaissances particulières.

Une vidéo de démonstration du prototype est disponible à l'adresse suivante :

<https://www.youtube.com/watch?v=LZprc0sueDk>

11. Perspectives d'évolution

Étant à l'initiative de ce projet, je souhaite continuer son développement et résoudre le problème de traitement des données évoqué dans la partie « Problèmes rencontrés ».

Afin d'alléger le système, je prévois de remplacer certaines bielles actuellement en aluminium par des pièces en PLA imprimée en 3D. Cela permettrait de rendre le système plus maniable et moins consommateur en énergie.

Il peut également être étudié d'ajout d'une connectivité Bluetooth pour permettre d'effectuer les réglages du système depuis une application Smartphone et de pouvoir ajouter plusieurs modes de fonctionnement.

12. Conclusion

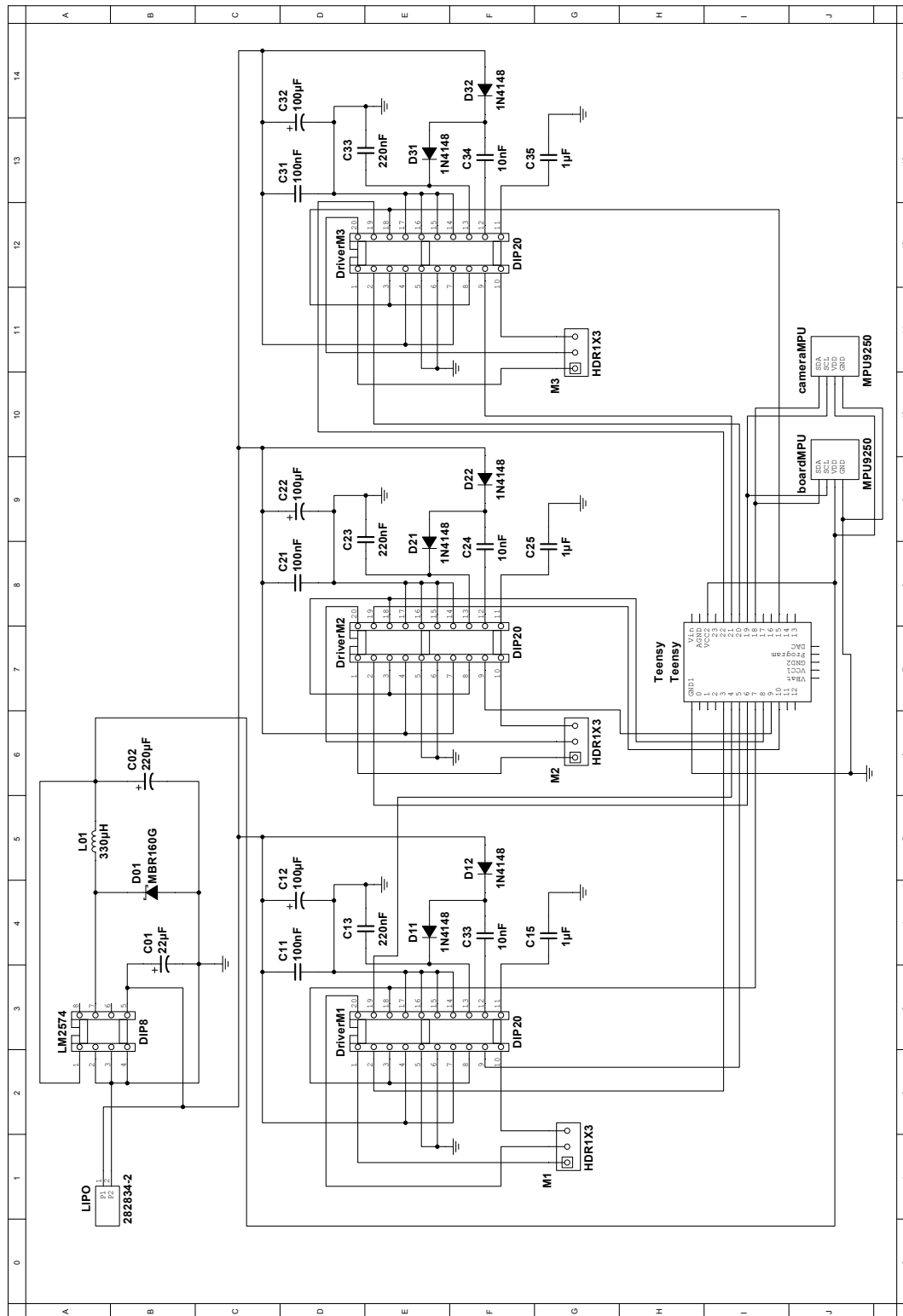
Ce projet est à mon sens un aboutissement dans ma formation d'ingénieur. Cela m'aura permis de mettre en œuvre un grand nombre de compétences techniques, mais surtout organisationnelles. Tout au long du projet, j'ai travaillé à prendre du recul sur les choix que j'ai effectués. Cette démarche m'a permis de mener à bien ce projet d'envergure, dans lequel je me suis beaucoup investi et qui m'a passionné.

13. Annexes

Ressources

Lien GitHub vers le dépôt du projet : <https://github.com/bibi03331/PFE-Polytech>

Schéma de principe de la carte électronique



14. Table des figures

Figure 1 : Prototype initial.....	6
Figure 2 : Banc d'essai moteur	6
Figure 3 : Schéma général du système	9
Figure 4 : Architecture du système	10
Figure 5 : Schéma de la structure mécanique	12
Figure 6 : Rendu 3D structure mécanique.....	12
Figure 7 : Éléments de protection	13
Figure 8 : Réalisation de la structure.....	13
Figure 9 : Structure assemblée.....	13
Figure 10 : Partie électronique - Schéma fonctionnel général	14
Figure 11 : Partie électronique - Schéma de principe	14
Figure 12 : Conception électronique - Tests.....	15
Figure 13 : Partie électronique - Conception du PCB	15
Figure 14 : Partie électronique - PCB.....	16
Figure 15 : Partie électronique - Carte électronique assemblée	16
Figure 16 : Partie logicielle - Phase d'initialisation	17
Figure 17 : Partie logicielle - Fonctionnement normal	18
Figure 18 : Intégration - Assemblage du prototype	20
Figure 19 : Intégration - Assemblage du capot de protection.....	20