

# INSTITUT NATIONAL DES SCIENCES APPLIQUÉES

## PROJET ELECTRONIQUE

### CAHIER DES CHARGES

---

## Un drone au bout des doigts

*Système de pilotage de drone à partir des gestes de la main*

---

*Auteurs :*

GADON Timothee

BOUDET Arthur

PEREL Thibault

CAILLARD Valentin

AUSCHER Octave

BUTTIN Domitille

NIESCIEREWICZ Benjamin

LE CALVEZ Antoine

4 janvier 2016

# Remerciements

Nous tenons à remercier et à témoigner toute notre reconnaissance à l'équipe pédagogique du département Systèmes et Réseaux de Communications, pour le soutien et l'aide précieuse qu'elle nous a apportés dans la réalisation de la première partie de ce projet.

Nous tenons également à la remercier pour avoir accepté d'investir dans le drone, sans lequel rien n'aurait pu aboutir.

Merci à Mme Fabienne NOUVEL, M. Christophe LEMOINE et M. Yvon DUTERTRE pour leurs remarques qui nous ont permis d'affiner notre projet et ainsi faciliter son élaboration.

Un grand merci à M. Gilles PICOULT pour son écoute, ses conseils précieux et son investissement dans la commande du drone.

Merci également à Mme Fanny GOURRET pour son aide dans la planification des tâches de notre projet.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Présentation du projet</b>	<b>6</b>
2.1	Expression du besoin . . . . .	6
2.2	Etat de l'art . . . . .	7
2.2.1	Projets et systèmes existants . . . . .	7
2.2.2	MultiWii . . . . .	8
2.3	Présentation de notre solution . . . . .	9
<b>3</b>	<b>Système de reconnaissance des gestes</b>	<b>11</b>
3.1	Présentation du module . . . . .	11
3.1.1	Principe d'une télécommande de drone . . . . .	11
3.1.2	Description générale . . . . .	12
3.2	Description technique . . . . .	13
3.3	Solution alternative . . . . .	16
<b>4</b>	<b>Drone</b>	<b>18</b>
4.1	Présentation . . . . .	18
4.2	Choix du drone . . . . .	19
4.2.1	Critères généraux . . . . .	19
4.2.2	Présentation des solutions . . . . .	20
4.2.3	Solution choisie . . . . .	21
4.3	Traitement des données reçues . . . . .	22
4.4	Pilotage . . . . .	24
<b>5</b>	<b>Transmission des données</b>	<b>26</b>
5.1	Émission des données . . . . .	26
5.2	Premiers tests . . . . .	27
<b>6</b>	<b>Gestion du projet</b>	<b>29</b>
6.1	Diagramme de Gantt . . . . .	29
6.2	Budget . . . . .	31
<b>7</b>	<b>Démonstration</b>	<b>32</b>
<b>8</b>	<b>Conclusion</b>	<b>33</b>

## Table des figures

1	MQ1 Predator : drone militaire . . . . .	6
2	DJI Phantom 2 Vision+ : drone civil . . . . .	7
3	Projet "Danse avec les drones" . . . . .	8
4	Pilotage à l'aide d'une manette Wiimote . . . . .	8
5	Manette de Wii . . . . .	8
6	Multiwii . . . . .	9
	a     Arduino IDE . . . . .	9
	b     Interface logiciel MultiWii . . . . .	9
7	Contrôle manuel . . . . .	10
8	Télécommande de drone classique . . . . .	11
9	Axes d'aéronautique . . . . .	11
10	Contrôle du Roll, Pitch et Throttle . . . . .	12
	a     Roll . . . . .	12
	b     Pitch . . . . .	12
	c     Throttle . . . . .	12
11	Implantation des modules . . . . .	13
12	MPU-6050 . . . . .	13
13	Communication MPU-6050 serial . . . . .	14
14	Schéma bloc du MPU6050 . . . . .	15
15	Connexion Arduino - MPU6050 . . . . .	16
16	Mouvements de contrôle - Solution alternative . . . . .	17
	a     Pitch . . . . .	17
	b     Roll et Gaz . . . . .	17
17	Schéma simplifié du drone . . . . .	18
18	Partie pilotage du drone . . . . .	18
19	Partie réception du drone . . . . .	19
20	Schéma global du drone . . . . .	19
21	AR drone . . . . .	20
22	Drone open source complet . . . . .	21
23	Kit FPV250 Racing . . . . .	21
24	Schéma global pour la réception des données . . . . .	22
25	Degrés de liberté de vol . . . . .	22
26	Modification des signaux PWM . . . . .	23
27	Schéma global pour le pilotage . . . . .	24
28	Matériel pour tests . . . . .	27
	a     Arduino Uno . . . . .	27
	b     MPU-6050 . . . . .	27
	c     nRF24l01 . . . . .	27
29	Affichage yaw, pitch, roll en réception . . . . .	27
30	Diagramme de Gantt Cahier des charges . . . . .	29
31	Diagramme de Gantt Réalisation . . . . .	30

## Liste des tableaux

1	Tableau comparatif des Arduino . . . . .	23
2	Caractéristiques du Multiwii 328P . . . . .	25
3	Tableau comparatif des modules 2,4 Ghz . . . . .	26
4	Liste des composants . . . . .	31

# 1 Introduction

Dans le cadre du projet d'électronique de 4<sup>ème</sup> année au sein du département Systèmes et Réseaux de Communication de l'INSA Rennes, notre équipe est constituée des étudiants suivants : Octave AUSCHER, Arthur BOUDET, Domitille BUTTIN, Valentin CAILLARD, Timothée GADON, Antoine LE CALVEZ, Benjamin NIESCIEREWICZ, Thibault PEREL. En raison de la mobilité à l'international induite par la formation INSA, nous précisons que Valentin et Timothée seront absents au second semestre.

Notre idée pour ce projet est de piloter un drone seulement avec les gestes de la main. Le but est de rendre le pilotage ludique, intuitif et plus simple. La main remplace alors la télécommande. Le pilotage devient ainsi accessible au plus grand nombre.

Ce projet s'étend sur toute l'année universitaire et s'articule autour de deux parties. Pendant le premier semestre, de septembre à janvier, nous avons rédigé ce cahier des charges. Ce document présente la synthèse de notre réflexion au cours des derniers mois ainsi que les solutions qui ont été retenues. Sa rédaction permet de préparer le plus précisément possible notre plan d'action pour le second semestre, c'est-à-dire l'élaboration du prototype. La deuxième partie du projet est l'exécution du cahier des charges, c'est-à-dire la mise en place de la solution technique choisie. Le but est de valider un concept de fonctionnement par une démonstration finale courant mai 2016.

L'intérêt de ce projet est multiple. Il nous permet d'asseoir les compétences théoriques et pratiques acquises au sein du département SRC, mais aussi de nous auto-former plus en détail sur des notions techniques. Le but est également de nous former à notre futur métier d'ingénieur : il s'agit d'un travail en équipe, de la conception à la réalisation, sur un sujet technique et innovant tout en respectant des contraintes budgétaires et temporelles.

Ce rapport s'articule autour de cinq grands axes : une présentation générale, une partie dédiée à la reconnaissance des gestes, une autre pour la partie "drone", une autre concernant la transmission sans fil des données et enfin les outils mis en place pour une bonne gestion du projet.

La première partie explique ce que notre projet apporte de nouveau dans le domaine. Elle présente également l'état de l'art ainsi que les solutions retenues. Les trois parties suivantes (gant, drone et canal de transmission) présentent les solutions techniques possibles et celles que nous avons sélectionnées. Nous avons également détecté les sources d'erreurs ou de complications que nous pourrions rencontrer au second semestre. Enfin, nous expliquons les moyens mis en œuvre afin de respecter les contraintes de temps et de budget. Le diagramme de Gantt décrit comment nous avons géré notre temps pendant ce semestre. Il s'agit également d'une prévision de notre organisation pour le second semestre. Finalement, la partie concernant le budget établit la liste des composants qui seront utilisés pour concevoir notre prototype.

## 2 Présentation du projet

### 2.1 Expression du besoin

Les drones sont de plus en plus utilisés dans de diverses applications. Nous avons décidé de développer un gant permettant de simuler une télécommande de drone.

A l'origine, les premiers drones ont été développés par les armées britannique et américaine pour effectuer des missions de reconnaissance sans devoir mettre en danger la vie d'un pilote.



FIGURE 1 – MQ1 Predator : drone militaire

Aujourd'hui, les drones sont à la fois utilisés par les mondes civil et militaire. Ainsi, en France, l'armée de l'air commença à utiliser des aéronefs sans pilote à partir des années 1950. Les sapeurs pompiers de Paris ont testé en 2009 un drone permettant de compter le nombre de personnes présentes sur le lieu d'un incendie, sans avoir à y pénétrer. Cette année, la Gendarmerie Nationale a passé un appel d'offres afin d'acquérir des petits drones pour effectuer des missions de reconnaissance.

Les drones sont souvent munis d'une caméra embarquée et sont utilisés pour filmer des événements sportifs (courses, matchs de football ...).



FIGURE 2 – DJI Phantom 2 Vision+ : drone civil

Quelques étudiants en école d'ingénieurs décident de concevoir des drones dans le cadre d'activités extra-scolaires. Nous avons profité du projet d'électronique pour réaliser le nôtre tout en définissant un nouveau mode de pilotage. En effet, le pilotage classique d'un drone nous paraissait peu intuitif.

Une télécommande de drone est composée de deux joysticks. Celui de droite permet à la fois de contrôler le roulis et le tangage. Le joystick de gauche sert à diminuer ou à augmenter les gaz et à faire varier le lacet.

Nous avons donc pensé qu'il serait plus intuitif d'utiliser les mouvements de la main pour contrôler à la fois le roulis, le lacet et le tangage pendant que l'autre main servirait à la gestion des gaz. C'est ainsi qu'est née l'idée de notre projet d'électronique.

## 2.2 Etat de l'art

### 2.2.1 Projets et systèmes existants

Comme mentionné précédemment, la plupart des projets déjà existants sur le sujet sont des projets universitaires. Au cours de nos recherches, nous avons pu identifier quelques projets intéressants comme le projet "Danse avec les drones" (Figure 3) développé par des élèves-ingénieurs de l'école Télécom Saint Étienne. Vidéo de démonstration

Ce projet permet à un drone de type quadcopter de pouvoir suivre les mouvements d'un danseur, comme si celui-ci était son partenaire. Le système fonctionne grâce à un algorithme de suivi par caméra (traitement d'image), ce qui ne convient pas pour notre projet à cause de la latence provoquée par le traitement en amont de la vidéo. Nous avons besoin en effet pour mener à bien ce projet d'un système réactif, sans aucune latence, pour pouvoir piloter correctement le drone.

Un deuxième projet cette fois développé par des étudiants de l'école Polytech' Lille, a retenu notre attention. Il s'agit de piloter un drone à l'aide d'une télécommande de Wii





FIGURE 3 – Projet "Danse avec les drones"

(Wiimote)<sup>(Figure 4)</sup>. Le but de ce projet présente des similarités avec notre projet, il s'agit de récupérer les données du gyroscope (accéléromètre) de la manette, les interpréter et diriger le drone en conséquence. La limite principale de ce projet est la connectivité entre la manette et le drone, qui doit s'effectuer par l'intermédiaire d'un smartphone Android connecté en Bluetooth <sup>source</sup>.



FIGURE 4 – Pilotage à l'aide d'une manette Wiimote

Pour notre projet, nous souhaiterions avoir une connexion directe entre le gant de pilotage et le drone, encore une fois pour éviter toute latence inutile et éviter les intermédiaires entre le système de pilotage et le drone.

### 2.2.2 MultiWii

« *MultiWii* » est un projet Arduino conçu pour contrôler des multicoptères (tels que des quadcoptères, hexacoptères, avions et même des hélicoptères).

Ce code a été à l'origine développé pour les manettes portatives *Nintendo Wii*<sup>(Figure 5)</sup>.

Ce projet a évolué et dispose désormais d'un grand nombre d'options supplémentaires. Le code a également été adapté pour fonctionner avec la plupart des capteurs. Il fonctionne notamment avec le *MPU6050* qui dispose d'un gyroscope à trois axes ainsi que d'un accéléromètre à trois axes.



FIGURE 5 – Manette de Wii

Un des éléments essentiels à noter est que le MultiWii est un projet Arduino hautement personnalisable. Il doit donc être configuré en fonction du matériel à utiliser et de l'utilisation prévue (commandes de vol, stabilisation, etc). Tout cela se fait au sein de l'environnement de programmation *Arduino IDE* (Figure 6a).

La modification du code source, tout en assurant le support de la stabilisation et l'asservissement des moteurs, est gérée directement par le programme. C'est la raison pour laquelle nous nous sommes penchés sur cette solution.

Le programme est également fourni avec un petit logiciel permettant le réglage de différents paramètres : armement du drone et de ses moteurs, réglages des seuils concernant les communications avec la télécommande (ou ici de notre système de pilotage) (Figure 6b) ... Il nous permet également de visualiser en temps réel les différentes voies d'entrée (inputs) de notre système de pilotage, ce qui nous sera très utile pour de futurs tests.

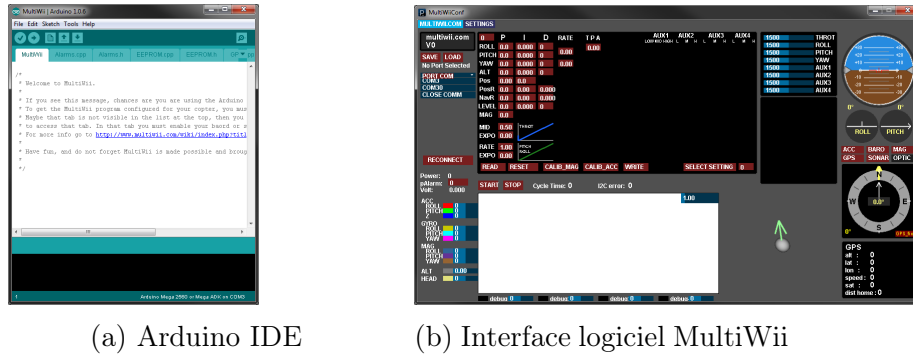


FIGURE 6 – Multiwii

## 2.3 Présentation de notre solution

L'idée principale repose sur la reconnaissance des mouvements de la main pour piloter le drone. Cette capture se fera au niveau de l'utilisateur par un accéléromètre et un gyroscope. Les données seront alors envoyées par radio au drone qui pourra les interpréter en conséquence afin de se déplacer conformément aux excitations. Comme sur la plupart des télécommandes, aucun retour d'information n'est mis en place.



FIGURE 7 – Contrôle manuel

La suite de ce document développe les moyens et technologies choisis pour chaque module qui pourront mener à l'aboutissement du projet avec succès.

## 3 Système de reconnaissance des gestes

### 3.1 Présentation du module

Nous allons présenter dans cette partie la solution retenue concernant le choix du mode de contrôle du drone.

L'innovation de ce projet réside principalement dans cette partie car nous proposons un mode de contrôle totalement innovant. Contrôler un drone avec le mouvement des mains n'est pas quelque chose que l'on peut trouver chez son marchand de modélisme préféré. En effet, tous les drones proposés dans les commerces peuvent se piloter à l'aide d'une télécommande. Cette télécommande est composée au minimum de 2 joysticks.

#### 3.1.1 Principe d'une télécommande de drone



FIGURE 8 – Télécommande de drone classique

Le premier joystick permet le contrôle du throttle (gaz) et du yaw. Le throttle est particulièrement difficile à doser car il faut tout le temps l'ajuster. On ne peut pas se contenter de pousser légèrement le joystick pour faire décoller le drone et le stabiliser en l'air. Pour avoir un drone stable en altitude, il faut en permanence augmenter légèrement le throttle puis le diminuer. Le yaw permet de faire faire une rotation sur lui-même au drone. La tête, donc l'avant du drone, peut alors se retrouver face au pilote. Cela implique que tous les mouvements disponibles sur le second joystick sont inversés. Cet axe de contrôle entraînant des difficultés supplémentaires de pilotage, nous avons décidé de ne pas l'implémenter pour le moment, le but étant de simplifier au maximum le pilotage. Nous utiliserons donc le drone en mode headless, c'est à dire que quel que soit l'orientation du drone (donc la valeur de son yaw), il gardera toujours le même cap, donc aura toujours la même direction. L'avant et l'arrière du drone seront alors inchangés pour le pilote.

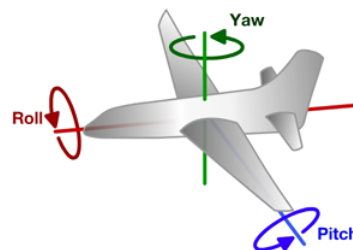


FIGURE 9 – Axes d'aéronautique

Le second joystick permet le contrôle du roll et du pitch. Cela permet un déplacement du drone dans un plan horizontal, par de légères inclinaisons. Lorsque l'utilisateur souhaite déplacer le drone dans ce plan, on remarque une légère diminution de la rotation des moteurs qui sont placés vers la direction souhaitée, tandis que les 2 autres moteurs vont tourner plus vite. Cela permet au drone de s'incliner vers la direction souhaitée et donc de se déplacer. Il faut toutefois continuer à ajuster le throttle avec le premier joystick, sinon le drone perd de l'altitude très rapidement.

### 3.1.2 Description générale

La première solution que nous allons vous présenter permettra un contrôle du drone à l'aide des deux mains. Cette solution présente des avantages à la fois au niveau de la prise en main de l'utilisateur mais aussi au niveau de la conception et de la réalisation technique.

La première main permettra de transmettre directement les mouvements du roll et du pitch. En plaçant la main à plat, celle-ci enverra des valeurs d'angles qui seront transmises directement au drone. Le drone pourra alors interpréter ces valeurs comme des instructions classiques.

Concernant la seconde main, la première piste de développement envisagée est de l'assimiler à une pédale d'accélération. Les angles d'inclinaison de la main seront identifiés comme une position du joystick du throttle d'une télécommande.

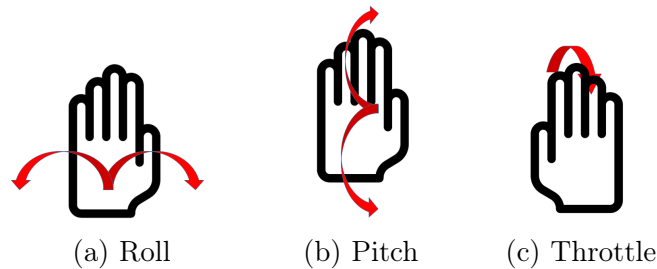


FIGURE 10 – Contrôle du Roll, Pitch et Throttle

Chaque main aura donc sa fonction spécifique. De plus, étant donné que l'utilisateur enfle de simples gants pour piloter le drone, il lui sera possible de choisir le type de contrôle sur chaque main en fonction de ses préférences.

Les gants seront reliés à un Arduino qui effectuera les opérations de filtrage des données. Les données seront alors émises vers le drone à l'aide d'un émetteur à 2.4GHz.

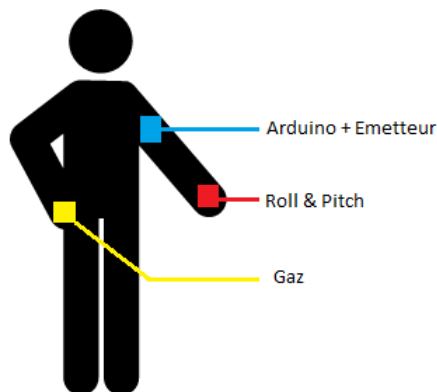


FIGURE 11 – Implantation des modules

L'intérêt d'un contrôle à l'aide des deux mains est donc de transmettre directement les mouvements de la main de l'utilisateur au drone. De plus, la prise en main du drone ne peut être plus intuitive, notamment sur le pilotage du roll et du pitch.

### 3.2 Description technique

Pour réaliser les fonctions de pilotage avec les mains que nous avons décrites précédemment, la solution retenue est d'utiliser des gyroscopes qui seront fixés sur des gants.

Ces gyroscopes permettront de détecter l'inclinaison que prennent les mains. Ainsi se pose la question du choix des composants qui seront utilisés. Le circuit qui a rapidement été retenu est le MPU-6050. En effet, ce dernier a l'avantage de posséder à la fois un gyroscope ainsi qu'un accéléromètre. Il dispose de trois axes pour le gyroscope et trois autres axes pour l'accéléromètre.

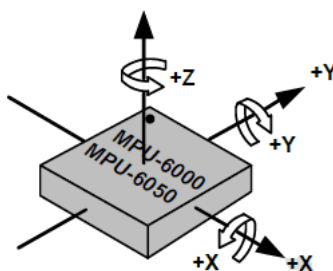
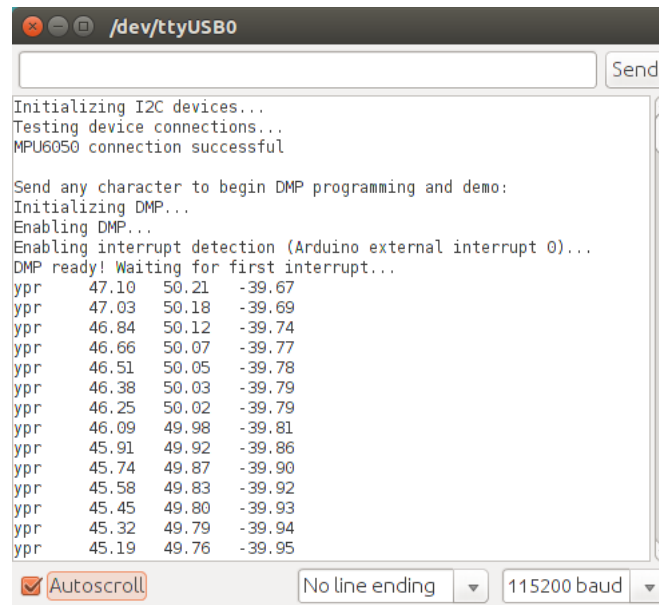


FIGURE 12 – MPU-6050

L'implémentation finale ne sera donc pas modifiée en fonction du mode de pilotage retenu. Les données fournies par ce composant sont très précises. Le MPU-6050 réalise une conversion analogique/digitale sur 16 bits. Un autre avantage de ce composant est son prix qui est très modeste (cf Budget). Enfin le MPU-6050 étant très utilisé (en particulier par la communauté Arduino), beaucoup d'informations et de codes sont disponibles sur ce composant.

Ce dernier point est très important. En effet il est relativement simple d'accéder aux mesures brutes de ce capteur (on possède directement les angles YAW, PITCH et ROLL). Cependant ces données sont très compliquées à interpréter. Heureusement, il existe des codes sur Arduino qui permettent de sortir ces données sous formes plus intuitives (angles compris entre -180 et +180 degrés). Sur le terminal suivant on peut observer les valeurs du YAW, PITCH et ROLL (respectivement sur les colonnes 1, 2 et 3) obtenues lors d'un test.



```
/dev/ttyUSB0
Send

Initializing I2C devices...
Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
Initializing DMP...
Enabling DMP...
Enabling interrupt detection (Arduino external interrupt 0)...
DMP ready! Waiting for first interrupt...
ypr  47.10  50.21 -39.67
ypr  47.03  50.18 -39.69
ypr  46.84  50.12 -39.74
ypr  46.66  50.07 -39.77
ypr  46.51  50.05 -39.78
ypr  46.38  50.03 -39.79
ypr  46.25  50.02 -39.79
ypr  46.09  49.98 -39.81
ypr  45.91  49.92 -39.86
ypr  45.74  49.87 -39.90
ypr  45.58  49.83 -39.92
ypr  45.45  49.80 -39.93
ypr  45.32  49.79 -39.94
ypr  45.19  49.76 -39.95

Autoscroll No line ending 115200 baud
```

FIGURE 13 – Communication MPU-6050 serial

Il sera aussi important de créer (ou de trouver dans une bibliothèque déjà codée) une fonction de filtrage des mesures brutes. En effet, il est nécessaire de ne pas prendre en compte les micro-fluctuations de la main et de lisser les valeurs obtenues. Il existe pour cela des méthodes plus ou moins simples (moyenne glissante, filtre de Kalman...).

Il faut à présent étudier le câblage qui devra être fait entre l'Arduino et le MPU-6050. Les broches du composant qui doivent être utilisées se trouvent sur la datasheet de ce dernier.

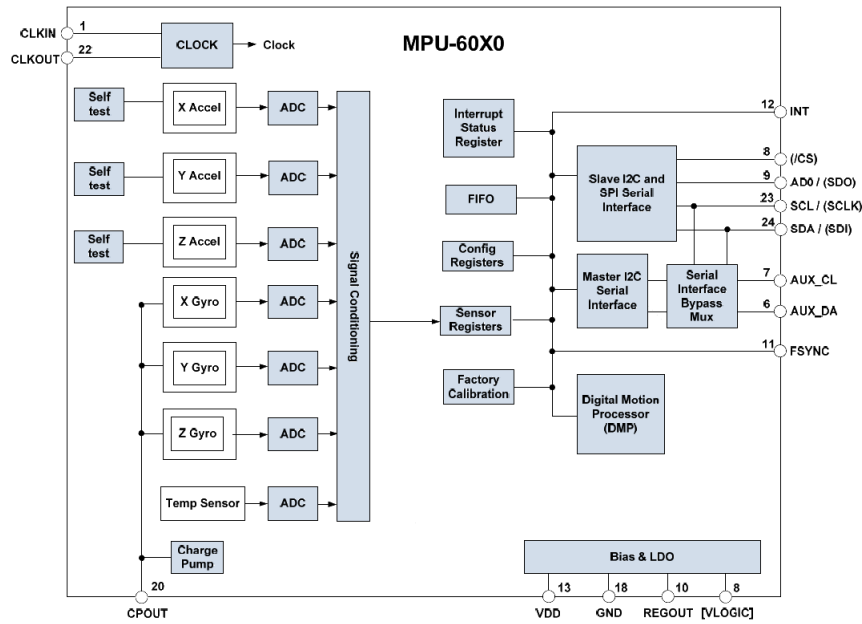


FIGURE 14 – Schéma bloc du MPU6050

Les sorties VCC et GND sont évidemment nécessaires pour l'alimentation du composant. Le MPU-6050 communique les données via les pins SDA et SCL en utilisant un protocole I<sup>2</sup>C.

La pin SDA (Serial Data Line) correspond à la ligne de données bidirectionnelle ; la pin SCL (Serial Clock Line) correspond à la ligne d'horloge de synchronisation bidirectionnelle. Ces deux broches seront connectées aux entrées analogiques de l'Arduino (par exemple les entrées A4 et A5). Il faut aussi brancher le signal d'interruption digital (pin INT) du MPU-6050 sur l'Arduino. On obtient ainsi le montage présenté en Figure 15.



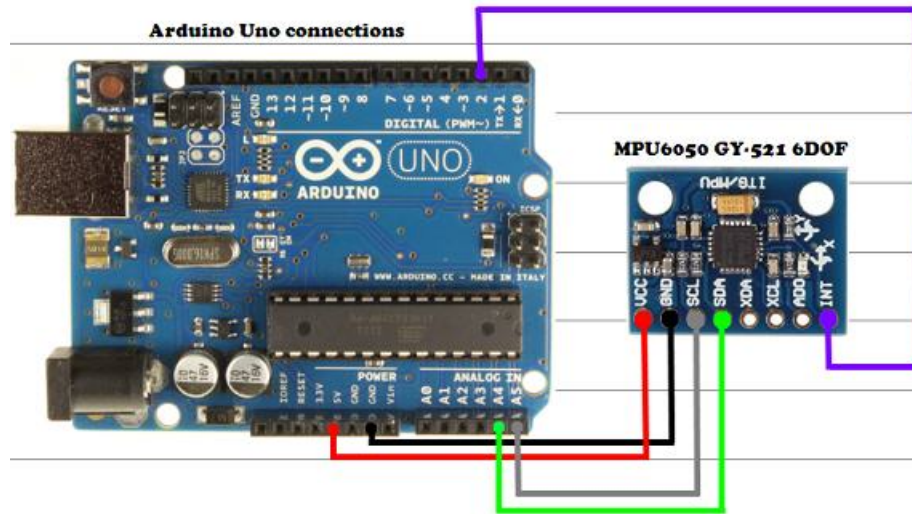


FIGURE 15 – Connexion Arduino - MPU6050

La carte possédant le MPU-6050 sera fixée sur un gant, les connexions avec l'Arduino se feront alors avec de simple fils (ce dernier sera fixé par exemple sur le bras de l'utilisateur).

### 3.3 Solution alternative

Si nous disposons du temps et des ressources nécessaires, nous souhaitons développer une deuxième solution de pilotage. Celle-ci a pour but principal de contrôler le drone à l'aide d'une seule main, et donc de gérer l'intégralité des commandes sur un seul accéléromètre.

Le drone sera piloté par "visée" de la main. En effet, l'utilisateur devra viser le drone avec sa main, bras tendu. Pour déplacer le drone vers la gauche ou la droite, il devra simplement déplacer son bras. De même pour le throttle, il devra lever ou abaisser son bras. En ce qui concerne l'avance ou le recul du drone (pitch), l'utilisateur effectuera des flexions du bras. Nous devons également placer un interrupteur, contrôlé à l'aide des doigts, afin que l'utilisateur puisse replacer son bras dans la bonne position. Cela indiquerait au drone de se placer en mode stabilisé.

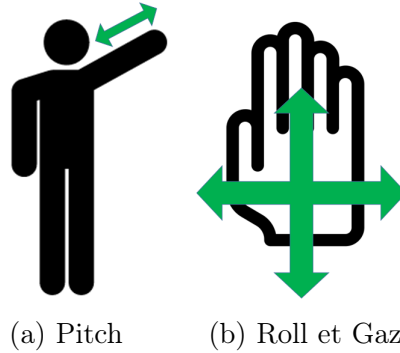


FIGURE 16 – Mouvements de contrôle - Solution alternative

Cette solution est totalement différente de la première. En effet, nous utilisons beaucoup moins les angles du gyroscope (uniquement pour de la vérification), tandis que l'accéléromètre sera exploité dans son intégralité (les 3 axes seront utilisés). Un travail sur les valeurs renvoyées en fonction des accélérations subies est à prévoir, afin de faire réagir le drone à la main le plus fidèlement possible.

De plus, le code implémenté sur l'Arduino sera complètement différent car les données à interpréter seront différentes, tout comme les fonctions de filtrage des valeurs parasites qui seront à réajuster. Un vrai travail d'interprétation des données envoyées par l'accéléromètre sera donc à réaliser.

Le mode de vol du drone serait également différent. Nous souhaitons maintenir le drone à altitude constante, sans avoir à ajuster en permanence le throttle. Il existe le mode de vol "Altitude Hold", mais il faudrait placer un baromètre sur le drone. Pour éviter cela, le programme dans l'Arduino situé sur le bras enverrait, lorsqu'aucun mouvement ne serait détecté, soit une valeur constante pour les gaz, soit les pleins gaz pendant une durée et une fréquence précises. Ces valeurs seront à déterminer expérimentalement, en fonction des conditions extérieures, de la gravité, du poids du drone, etc.

Cependant, l'avantage de cette solution par rapport à la précédente est son coût. En effet, il n'y a plus qu'un seul accéléromètre utilisé, car il n'y a qu'une main pour piloter. Cela permet alors de diminuer les dépenses pour le projet. En contrepartie, ce mode de pilotage est plus complexe à réaliser.

Finalement, la principale difficulté dans la partie de gestion des mouvements réside dans le code que nous devrons fournir. En effet, nous devons tenter de retranscrire les données que peut envoyer une télécommande le plus fidèlement possible. L'interprétation et l'analyse des données envoyées par l'accéléromètre représentera une partie importante dans l'élaboration de ce projet.

## 4 Drone

### 4.1 Présentation

Le drone est un élément essentiel du projet. Il doit être capable de recevoir les informations envoyées par le gant de l'utilisateur, de les interpréter et de piloter les moteurs. Un schéma de principe est donné Figure 17.

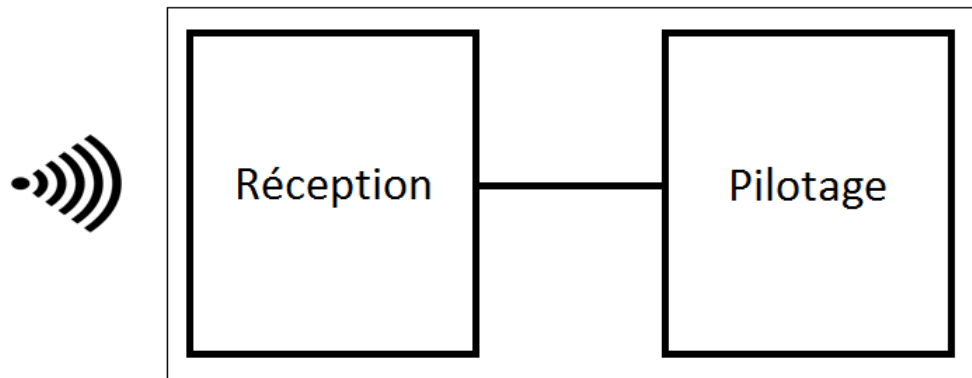


FIGURE 17 – Schéma simplifié du drone

Ce schéma simplifié permet d'identifier 2 tâches principales dans la fabrication du drone : réception et pilotage. La partie réception doit permettre de récupérer les données envoyées par le gant de l'utilisateur. De plus, il devra également effectuer des traitements pour fournir les bonnes informations à la partie pilotage. Il est donc logique de traiter la partie pilotage dans un premier temps et d'adapter la partie réception à celle-ci.

La première tâche à réaliser est de définir les éléments nécessaires à la partie pilotage. Le choix des composants sera détaillé ultérieurement. Les différents modules sont présentés Figure 18.

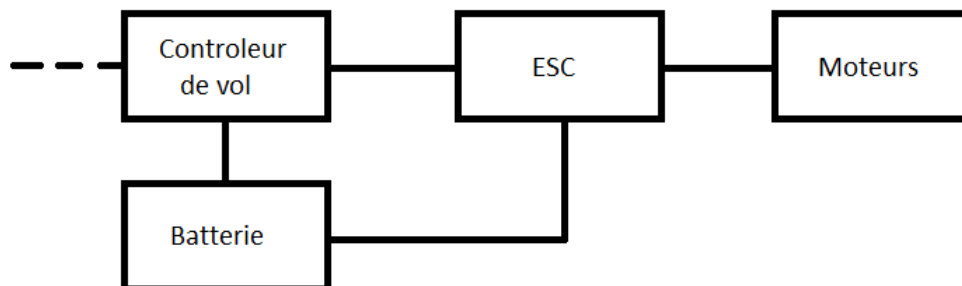


FIGURE 18 – Partie pilotage du drone

Le contrôleur de vol impose une mise en forme particulière des signaux à la partie réception. La deuxième tâche est donc de réaliser une partie réception capable de récupérer des données et de les mettre en forme. Ce principe est illustré Figure 19. Le schéma global du drone présenté Figure 20 est donc obtenu en combinant la partie réception et la partie pilotage.

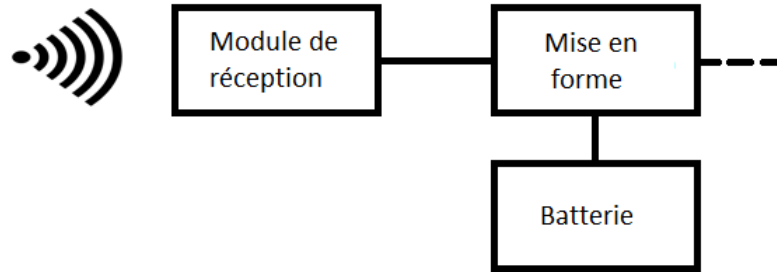


FIGURE 19 – Partie réception du drone

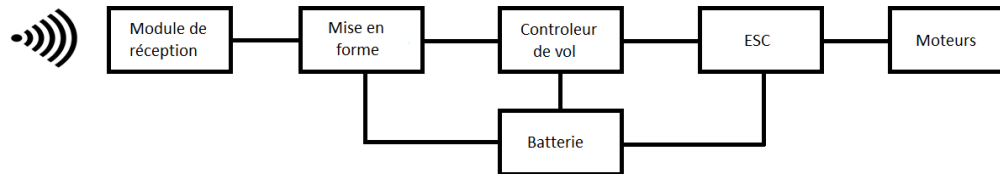


FIGURE 20 – Schéma global du drone

Le schéma du drone étant à présent défini, il faut choisir les composants à utiliser. Dans un premier temps, nous nous attarderons sur le choix du châssis et sur la partie pilotage. Ensuite, nous détaillerons nos choix pour la partie réception.

## 4.2 Choix du drone

Bien que la fabrication du drone ne soit pas une partie innovante de notre projet, celui-ci en conditionne le bon fonctionnement. Aussi, il est important de bien le choisir. Dans un premier temps, nous nous intéresserons à des critères généraux. Ensuite, nous nous focaliserons sur les solutions répondant aux exigences évoquées précédemment. Ces dernières possèdent toutes des caractéristiques différentes, notamment en termes de coût et de flexibilité.

### 4.2.1 Critères généraux

La première caractéristique sur laquelle nous nous sommes focalisés est la taille. En effet, la taille du châssis est déterminante car elle conditionne celle des moteurs et de la batterie, ce qui impacte considérablement les coûts. Un châssis trop grand entraînerait un prix

trop élevé. En revanche, un châssis trop petit compromet l'intégration facile de modules, ce qui est un handicap pour notre projet. De plus, la majorité des prototypes réalisés par des particuliers sont des drones de taille moyenne, la documentation y est donc plus conséquente.

Le deuxième critère concerne le type de drone utilisé. Dans le commerce, on distingue notamment les tricopters (3 hélices) des quadcopters (4 hélices) et des octocopters (8 hélices). Pour chaque hélice, un moteur et un ESC sont nécessaires. Le prix du drone est donc directement impacté par ce choix. Un bon compromis est d'utiliser un quadcopter : il s'agit du type de drone majoritairement utilisé par les amateurs et son coût est nettement inférieur aux octocopters.

Dans le cadre de ce projet, l'idéal est donc d'utiliser un drone standard de type quadcopter de taille moyenne. Il permet une intégration facile des modules à un coût raisonnable tout en garantissant une documentation fournie.

#### 4.2.2 Présentation des solutions

Pour fabriquer le quadcopter, nous envisageons plusieurs solutions. Il est donc nécessaire de comparer ces solutions afin de déterminer celle qui est la plus appropriée.

##### *AR drone*

Lors de la proposition de notre sujet aux encadrants du projet, il nous a été proposé d'utiliser un AR drone<sup>(Figure 21)</sup> que possède le département Systèmes et Réseaux de Communication de l'INSA Rennes.



FIGURE 21 – AR drone

Cette solution a pour avantage de réduire considérablement le coût. En revanche, ce drone appartient également au département Informatique de l'INSA Rennes, ce qui signifie que l'on est limité en matière de modifications. De plus, ce drone commercialisé par Parrot est de type « propriétaire », il n'est donc pas prévu pour être modifié. Ce manque de flexibilité est un handicap pour notre projet. Cette démarche met en évidence la nécessité d'utiliser un drone open source.

##### *Drone open source complet*

Une autre solution est d'acheter un drone open source complet<sup>(Figure 22)</sup> dans le commerce. Ces drones ont l'avantage d'être facilement modifiables. Cependant, le coût particulièrement élevé est un frein non négligeable.



FIGURE 22 – Drone open source complet

### *Drone en kit*

Les contraintes de prix et de flexibilité nous ont orientés vers l'achat d'un drone en kit. Il doit respecter les critères généraux évoqués précédemment (quadcopter de taille moyenne). L'idéal serait qu'il possède tous les éléments nécessaires à l'exception de la partie réception et du contrôleur de vol. En effet, nous souhaitons utiliser un contrôleur de vol compatible avec le code open source Multiwii, il est donc plus simple de l'acheter séparément. La réception sera réalisée avec un module NRF24L01 .

#### 4.2.3 Solution choisie

Nos recherches nous ont finalement orienté vers le kit « FPV250 Racing quadcopter »<sup>(Figure 23)</sup> proposé par HobbyKing, un des acteurs majeurs du domaine du modélisme.

Voici ses caractéristiques :

- quadcopter (4 moteurs, 4 esc, 4 hélices)
- batterie LiPo 1000mAh 11.1V
- taille du châssis : 25cmx25cm
- absence de module de réception et de contrôleur de vol



FIGURE 23 – Kit FPV250 Racing

Le choix concernant le contrôleur de vol est détaillé ultérieurement.

### 4.3 Traitement des données reçues

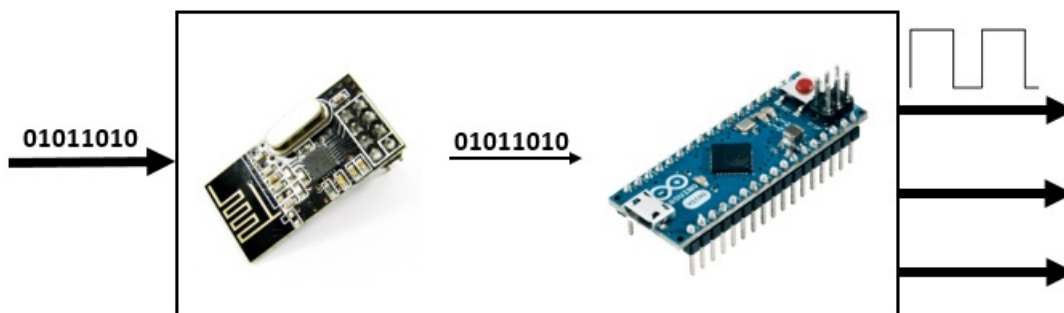


FIGURE 24 – Schéma global pour la réception des données

En ce qui concerne la réception des données, nous allons devoir exploiter les informations qui ont été émises sur le canal de propagation sous forme de trames. La première étape est donc la réception de ces trames à proprement parlé. Pour cela, nous utilisons le récepteur du module NRF24L01. Son fonctionnement est totalement transparent pour nous, c'est-à-dire que nous allons l'utiliser sans modification. Il va nous fournir en sortie la trame qui a été émise. L'étape suivante consiste à interpréter ces trames.

En aval du récepteur, nous allons placer un module Arduino, l'ensemble étant bien sûr directement implanté sur le drone. Il est important de souligner que le poids de l'Arduino et du récepteur est tout à fait négligeable devant la puissance du drone que nous comptons utiliser. En effet, il est conçu pour pouvoir supporter une caméra embarquée, et donc quelques grammes ne viendront pas altérer les possibilités de vol du drone. En réalité, l'Arduino devra traiter des trames par groupes de trois. Selon notre algorithme de capture sur le gant, les trois trames seront l'image de la tension récupérée par les accéléromètres, et ceci avec un ordre prédéfini, pour chacun des trois canaux : pitch, roll, ou throttle<sup>(Figure 25)</sup>.



FIGURE 25 – Degrés de liberté de vol

Chaque trame reçue représente donc une valeur correspondant à une tension. En fonction de cette valeur, l'Arduino va modifier le duty cycle d'un signal PWM codant un des trois canaux, qui sera ensuite envoyé vers le contrôleur Multiwii. L'exploitation de la trame suivante modifie le duty cycle d'un autre signal PWM codant un deuxième canal, et de même pour le canal restant. Une fois les trois trames traitées, d'autres trames sont reçues, et on recommence en boucle. La Figure 26 explique le fonctionnement pour une trame, et sur un canal particulier.

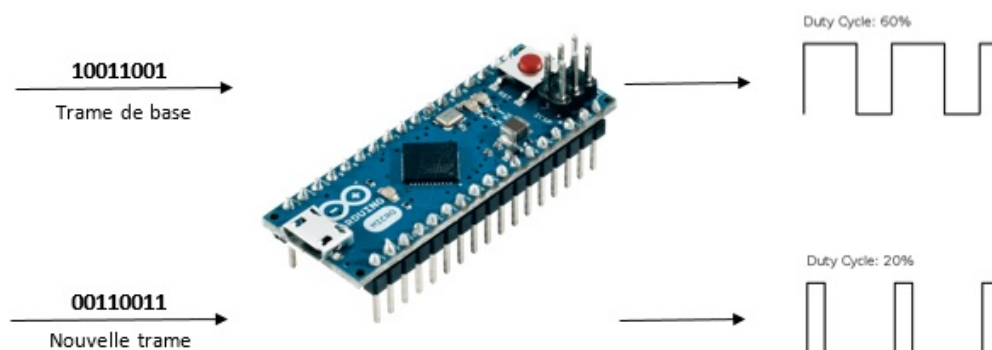


FIGURE 26 – Modification des signaux PWM

En ce qui concerne l'Arduino, plusieurs choix s'offrent à nous. En effet, il existe une gamme assez développée de microcontrôleurs Arduino, et nous avons donc dû nous attarder sur les spécifications de chacun pour pouvoir faire un choix. Compte tenu de la taille du drone, nous avons conclu qu'il était préférable d'utiliser le module le plus petit possible pour ne pas altérer les déplacements, sachant que la carte sera certainement rajoutée sur le drone sans avoir d'emplacement spécifique. Ce sont donc les Arduino Micro et Uno qui ont attiré notre attention. Voici leurs spécificités :

	Arduino Uno	Arduino Micro
Consommation (V)	5	5
Mémoire SRAM/Flash (KB)	2/32	2.5/32
Digital IO/PWM	14/6	20/7
Taille physique (mm)	74x53x15	50x18x19
Prix (€) (Atlantique Composants)	20	18

TABLE 1 – Tableau comparatif des Arduino



Dans les deux cas, la consommation est identique, et le module peut être alimenté via les ESC en 5V. Également, la mémoire est presque la même pour les deux modules, ainsi que le nombre de sorties PWM (nous en avons besoin de trois, les deux peuvent donc convenir).

Cependant, le Micro a les avantages de présenter un prix un peu plus bas que le Uno, et surtout une taille nettement inférieure :  $900 \text{ mm}^2$  d'occupation en surface contre  $3922 \text{ mm}^2$  pour le Uno. Naturellement, notre choix s'est donc porté sur le Micro, que nous implanterons donc sur le drone.

## 4.4 Pilotage

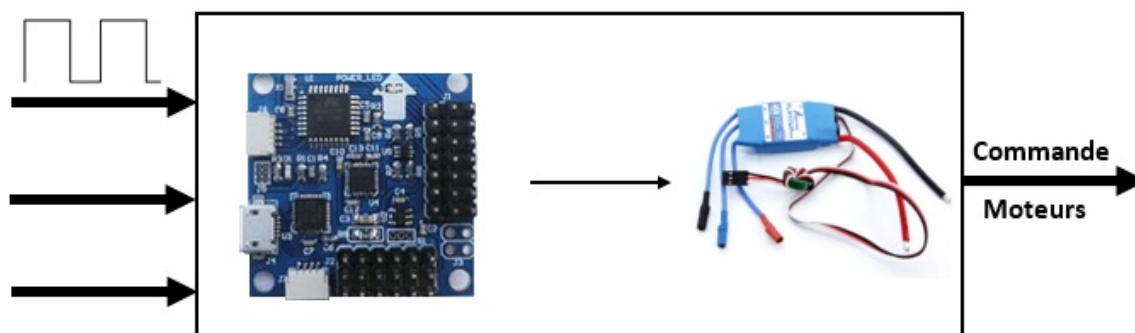


FIGURE 27 – Schéma global pour le pilotage

Le pilotage du drone va être principalement basé sur l'utilisation d'un organe fondamental : le contrôleur. Le fonctionnement de cet ensemble est tout à fait transparent pour nous, mais il est important d'en connaître quelques détails, notamment pour savoir quel type de données amener en entrée du contrôleur. Ce dernier va contenir un code open source, le code Multiwii, qui va permettre directement de contrôler le comportement des moteurs. Nous réutilisons ici un code qui a déjà été développé. Il est complètement autonome, nous n'avons pas besoin de le modifier. Le contrôleur prend en entrée des signaux PWM (d'où la sortie de trois signaux PWM à la réception des données). À partir des duty cycle de ces signaux, le code Multiwii envoie une commande aux ESC qui assurent le pilotage des moteurs.

Pour choisir le contrôleur adéquat, une des conditions était de pouvoir y implanter directement le code Multiwii. Nous nous sommes rendus compte qu'il existait des contrôleurs de vol où le code était déjà implanté, et était exploité à l'aide d'un Arduino. Nous avons également repéré que certaines cartes possédaient des capteurs intégrés : gyroscope et accéléromètre principalement. Ceci est fondamental car ce sont ces capteurs qui vont permettre de stabiliser le vol du drone, en effectuant un retour d'information sur le contrôleur. Forts de ces constatations, et en comparant les prix des différents contrôleurs existants, nous avons

finalement choisi le Multiwii 328P possédant entre autres les caractéristiques suivantes :

Capteurs intégrés	Gyroscope, Accéléromètre, Baromètre, Magnétomètre
Alimentation carte/capteurs	5V/3.3V
Contenu additionnel	Sortie pour moteur de contrôle d'une caméra
Dimensions (mm)	50x50x12
Masse (g)	13.9

TABLE 2 – Caractéristiques du Multiwii 328P

On peut remarquer que les dimensions de la carte ont été prévues pour permettre l'implantation direct du contrôleur de vol du drone que nous allons monter. De plus, sa masse est très faible et ne gênera donc pas le vol. Enfin, le contrôleur pourra être réutilisé par le département SRC ou par d'autres projets car il possède un panel de capteurs assez développé, et surtout un servomoteur permettant le contrôle des mouvements d'une caméra embarquée, ce qui pourrait aboutir à un autre projet.

En ce qui concerne les ESC (aussi appelés variateurs), ils permettent de contrôler la vitesse de rotation des moteurs, c'est-à-dire la puissance que nous allons leur fournir. A partir d'une commande en entrée issue du contrôleur (une commande des gaz par exemple), ils vont directement modifier le comportement des moteurs auxquels ils sont reliés. Une fois de plus, leur fonctionnement nous est transparent. Ils vont présenter en entrée deux fils permettant de les alimenter, et un fil de connexion au contrôleur. Ce fil, appelé BEC, permet à la fois de recevoir les instructions du contrôleur, et peut-être aussi utilisé pour alimenter d'autres éléments sur le drone (5,6 V en sortie de ce fil). Dans notre cas, nous l'utiliserons pour alimenter notamment l'Arduino Micro. Les ESC possèdent en sortie trois autres fils faisant la liaison avec le moteur.

Les ensembles que nous venons de décrire, à savoir la réception des données et le pilotage, vont donc nous permettre d'obtenir un contrôle total des mouvements du drone. La difficulté principale que nous pourrions rencontrer serait l'implantation de l'Arduino de réception sur le drone. En effet, tous les autres composants ont un fonctionnement transparent pour nous, et possèdent physiquement une place prévue sur le drone. Pour cet Arduino, il faudra donc élaborer le code de traitement des données reçues, et l'ajouter au drone en altérant au minimum son aérodynamisme.

## 5 Transmission des données

### 5.1 Émission des données

Pour assurer une réception correcte des données sur le drone, nous allons tout d'abord devoir créer une trame qui sera émise sur le canal de propagation. Nous avons pensé créer des paquets contenant les 3 informations indispensables pour contrôler le drone : pitch, roll et throttle. Ces valeurs seront issues de l'accéléromètre et seront mappées avec une valeur comprise entre 1000 (valeur min) et 2000 (valeur max).

Voici un comparatif des différents modules que nous avons trouvé pour réaliser la fonction de transmission des données :

	NRF24L01	Module Xbee
Consommation (mA)	13,50	50
Alimentation (V)	3.3	3.3
Débit (kb/s)	200	250
Protocole	Zigbee	Enhanced Shockburst
Prix (€)	2,72	25

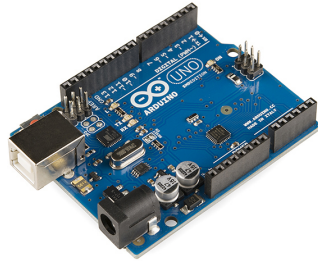
TABLE 3 – Tableau comparatif des modules 2,4 Ghz

Tout comme en réception, nous allons utiliser le module NRF24L01 pour émettre les informations issues de l'accéléromètre vers le drone. En effet, ce module est un transceiver, c'est à dire un émetteur/récepteur. Ces deux modules ont une grande communauté sur internet. De plus, ce module consomme le moins de courant et coûte le moins cher, ce qui justifie notre choix.

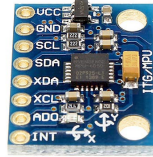
Par ailleurs, il est important de noter que la communication entre le gant et le drone est unidirectionnelle. Ainsi, le module NRF24L01 du gant fonctionnera uniquement en tant qu'émetteur de données. Rappelons par ailleurs que le fonctionnement du module est totalement transparent pour nous. Nous allons uniquement lui fournir la trame à émettre.

## 5.2 Premiers tests

Afin de valider le choix des nRF24L01 et de nous familiariser avec son fonctionnement nous avons effectué des tests de transmissions. Ceux-ci consistaient à envoyer des informations issues du gyroscope/accéléromètre MPU-6050, de les traiter par un Arduino pour les envoyer à un autre Arduino.



(a) Arduino Uno



(b) MPU-6050



(c) nRF24L01

FIGURE 28 – Matériel pour tests

Le code de traitement des données spatiales utilise la bibliothèque en langage C du MPU-6050<sup>1</sup> disponible sur Internet. Il permet, pour ces tests, d’afficher les angles de yaw, pitch et roll calculés par le module selon ses mouvements (cf figure).

```
reception_triplet_9_12_2015

Serial.println("Listening...");
}

void loop(){
  data[Mirf.payloadd];
  if(!Mirf.isSending() && Mirf.dataReady()){
    Mirf.getData((byte *) data);

    String yaw ="000";
    yaw.setCharAt(0,data[0]);
    yaw.setCharAt(1,data[1]);
    yaw.setCharAt(2,data[2]);
    iyaw = yaw.toInt();
    myaw = map(iyaw,100,460,-180,180);

    String pitch ="000";
    pitch.setCharAt(0,data[3]);
    pitch.setCharAt(1,data[4]);
    pitch.setCharAt(2,data[5]);
    ipitch = pitch.toInt();
    mpitch = map(ipitch,100,280,-90,90);

    String roll ="000";
    roll.setCharAt(0,data[6]);
    roll.setCharAt(1,data[7]);
    roll.setCharAt(2,data[8]);
    iroll = roll.toInt();
    mroll = map(iroll,100,280,-90,90);

    Serial.print("ypc");
    Serial.print("\n");
    Serial.print(myaw);
  }
}
```

ypc	115	-7	-3
ypc	116	-5	-3
ypc	116	-5	-3
ypc	117	-4	-3
ypc	107	2	13
ypc	119	0	23
ypc	117	0	23
ypc	116	-2	23
ypc	115	-2	23
ypc	114	-3	22
ypc	112	-4	22
ypc	111	-6	22
ypc	109	-7	22
ypc	108	-7	21
ypc	107	-8	21
ypc	105	-10	20

FIGURE 29 – Affichage yaw, pitch, roll en réception

1. <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>

Les programmes de transmission s'écrivent à l'aide des bibliothèques nRF24L01 et Mirf<sup>2</sup> codées pour l'utilisation du NRF24L01. Nous avons réussi à envoyer en continu un tableau contenant les valeurs des angles.

La programmation de ce système s'est avérée modérément complexe et le résultat est satisfaisant. Nous notons une portée suffisante pour notre application et un taux d'erreur très faible.

On observe cependant, avec ce code peu optimisé, une latence d'environ une demi-seconde entre l'affichage des données à l'envoi et l'affichage des données à la réception. A priori, même si d'autres traitements susceptibles d'augmenter cette latence ont lieu en réception, ceci semble acceptable pour notre application au sujet principal.

---

2. <https://github.com/aaronds/arduino-nrf24l01/tree/master/Mirf>

## 6 Gestion du projet

### 6.1 Diagramme de Gantt

L'organisation en différentes tâches de ce projet est synthétisée, pour le premier et second semestre, dans les diagrammes de Gantt suivants.

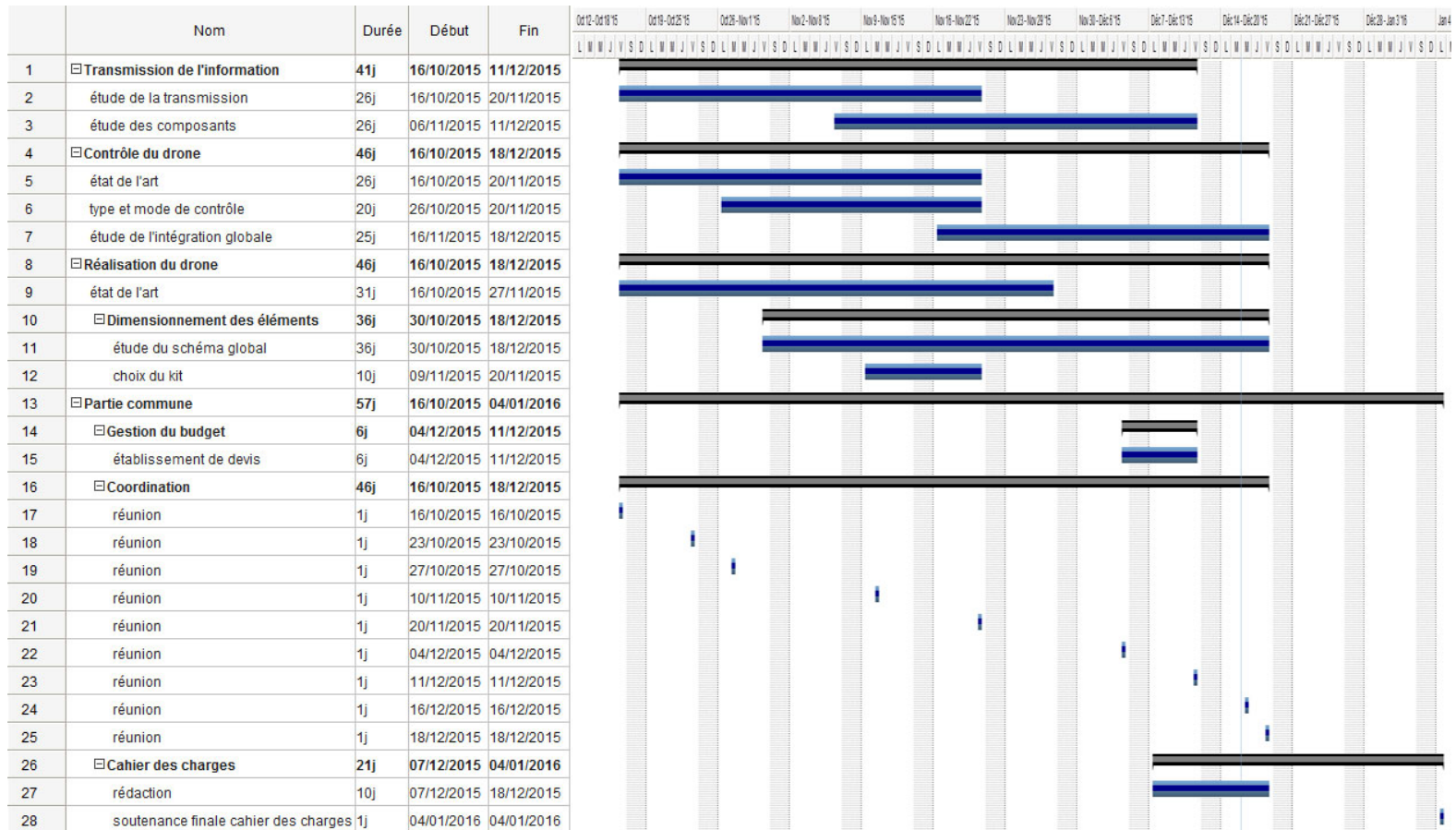


FIGURE 30 – Diagramme de Gantt Cahier des charges

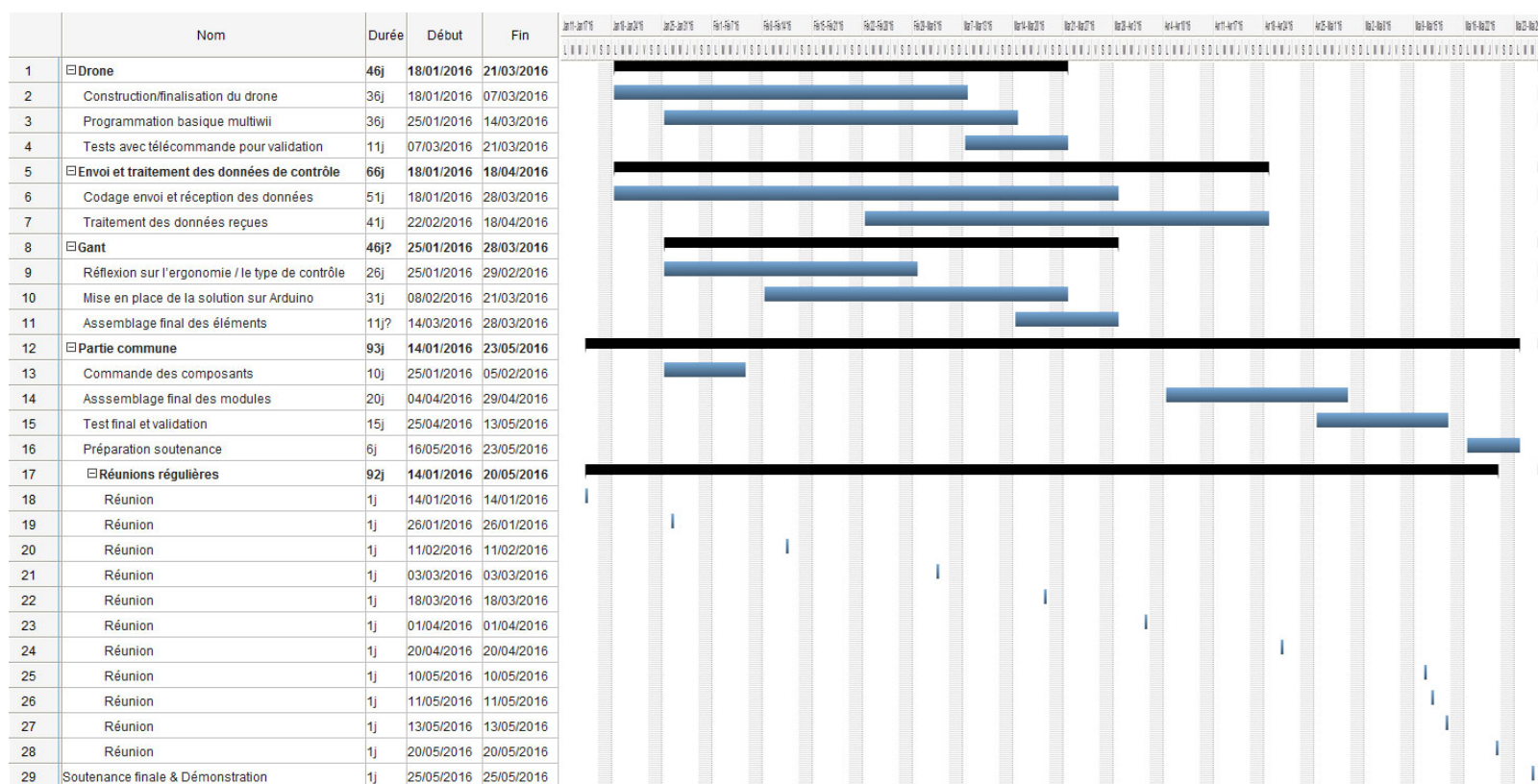


FIGURE 31 – Diagramme de Gantt Réalisation

## 6.2 Budget

Le département SRC nous a alloué un budget de 120€. La justification des différents composants choisis a été détaillée précédemment.

Voici la liste des composants, classés par fournisseur :

Nom composant	Prix (€)	Fournisseur	Descriptif
Kit drone	103	HobbyKing	HobbyKing Spec FPV250 V2 Quad Copter ARF Combo Kit Mini Sized FPV Multi-Rotor
Contrôleur Multiwii	25,62	HobbyKing	MultiWii 328P Flight Controller w/FTDI DSM2 Comp. Port
Arduino Nano	36	Atlantique Composants	Code produit : ODDA000005 ARDUINO NANO ATmega328 V3.0
Accéléromètre MPU-6050	1,84	eBay	GY-521 6 DOF MPU-6050 Module 3 Axis Accelerometer Gyroscope for Arduino IE
Module transmission RF	3,44	eBay	New 4pcs NRF24L01+ 2.4GHz Antenna Wireless Transceiver Module STGG for Arduino

TABLE 4 – Liste des composants

Nous pouvons préciser que certains composants ne sont pas affichés. Par exemple, les résistances qui seront nécessaires à l'adaptation d'alimentation du module RF. Il s'agit de composants déjà existants au département SRC et qui nous sont prêtés dans le cadre du projet pluridisciplinaire.

D'autre part, le gant est un élément déjà en notre possession. Nous ne l'incluons pas dans notre budget.

Nous prévoyons d'acheter deux composants sur eBay. Nous savons que ce fournisseur n'apparaît pas dans la liste imposée par le département. Cependant, les fournisseurs conseillés disposent de ces composants qu'en CMS. Cela aurait entraîné des complications puisque l'INSA ne dispose pas du matériel adéquat pour souder ce type de composant.

Remarquons que le budget total dépasse les 120€ initiaux. Cependant, l'équipe pédagogique a accepté d'acheter le drone ainsi que le contrôleur MultiWii. C'est un investissement pour le département puisque ce drone n'a pas de code propriétaire et pourra donc être utilisé par d'autres étudiants ou chercheurs pour d'autres projets.

Finalement, notre budget total est de : 37,84 €. Cela respecte les règles de départ et nous laisse une marge raisonnable pour les différents frais de port ainsi qu'imprévus.



## 7 Démonstration

Notre démonstration se déroulera dans un environnement clos mais haut de plafond. Nous pensons l'effectuer dans la halle de l'INSA. Le drone sera posé sur le sol et un volontaire se munira du dispositif permettant de contrôler le drone (accéléromètres, Arduino et émetteur). Le pilote se placera à quelques mètres du drone et effectuera quelques gestes pour montrer le bon fonctionnement du gant. Le drone sera mis sous tension par un autre membre du groupe.

## 8 Conclusion

Ce premier semestre nous a permis d'établir les différentes technologies possibles pour l'élaboration de notre projet et de sélectionner les plus efficaces. Nous avons également mis en évidence les difficultés qui pourraient survenir lors de la conception. Nous avons alors détaillé les solutions et les aides qui pourraient être utilisées. Malgré les difficultés que nous pourrions rencontrer, l'innovation qu'apporte notre projet est motivante.

Nous avons également pu compter sur l'aide précieuse des encadrants du département, qui nous ont poussé à une amélioration continue de notre projet grâce à des questions soulevées au cours des différentes présentations intermédiaires.

L'ensemble des composants respectent le budget, les tâches sont réparties, la planification jusqu'à la démonstration finale est effectuée. Maintenant, nous pouvons nous atteler à la mise en pratique de ce qui a été établi dans ce cahier des charges : la réalisation du prototype.