

Etude et expérimentation de petits quadricoptères destinés à des missions d'observation

Travail de Bachelor

Alain Vuille
10/06/2014

Professeur Responsable
Lorenzo Zago

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Hes·SO

University of Applied Sciences
Western Switzerland

Travail de bachelor 2014*Etude et expérimentation de petits quadricoptères destinés à des missions d'observation*

*Proposé par : heig-vd***Résumé**

Etude et expérimentation de petits robots volants destinés à des missions d'observation. Ce projet sera la suite d'un travail précédent qui a produit un premier prototype qui, toutefois, n'est pas encore satisfaisant. Il s'agira de comprendre comment l'améliorer et ensuite construire un deuxième prototype qui tiendrait compte de la rapide et récente évolution de cette technologie.

Cahier des charges

Complément de formation sur les quadricoptères : aérodynamique, électromécanique, régulation, programmation sur microcontrôleur. La commande est basée sur des microcontrôleurs MSP430 et Arduino.

- Analyse et essais du prototype existant, identification des problèmes et améliorations possibles.
- Etude de l'état de l'art et des produits actuellement présents sur le marché :
 - Catégories de produits et de tailles,
 - Cartes CPU,
 - Types de capteurs : gyromètres, accéléromètres, centrales inertielles, analogiques, numériques, ...,
 - Communication sans fil,
 - Batteries.
- Conception et construction d'un système de taille optimale compatible avec une caméra sans fil.

J...

heig-vd

- Essais avec, en particulier, divers types de capteurs inertiels, en air calme et turbulent (vent).

Vers un quadricoptère autonome et « intelligent »

- Définition de l'objectif, par exemple :
 - Décollage et atterrissage autonome (avec, par exemple, altimètre plus capteur optique),
 - Stabilisation à point fixe,
 - Trajectoire programmée.
- Configuration des capteurs et du logiciel. Configuration d'un protocole de commande sans fil ad hoc.
- Essais et mesures.
- Rapport de l'étude.

Le candidat :


Alain Vuille

Le professeur responsable :


Lorenzo Zago

Le chef du département des
Technologies industrielles


Prof. Guido Frosio

Remerciements

En premier lieu, je tiens à remercier M. Lorenzo Zago pour m'avoir permis de réaliser ce projet ainsi que pour son suivi et son expertise tout au long du travail

Je remercie également toutes les personnes de mon entourage qui m'ont aidé et soutenu durant les quelques mois qu'a duré ce travail.

Un grand merci en particulier à mes collègues Sarah, Damian, Linard, Kevin et Nicolas pour leurs conseils et leurs soutiens.

Merci encore à toutes les personnes qui œuvrent au sein de l'école pour faciliter le travail des diplômants. En particulier les employés de l'atelier mécanique, rapides et efficaces.

Un grand remerciement finalement à mes parents, relecteurs de dernière minutes, pour leur disponibilité et leur efficacité.

Je dédie ce travail à mon grand-père, Daniel, qui me demandait régulièrement comment mon « hélicoptère » se portait.

TABLE DES MATIÈRES

Table des matières.....	i
Table des illustrations.....	iv
Liste des tableaux.....	v
1. Introduction.....	1
2. Introduction aux quadricoptères.....	2
2.1. Mécanique du quadricoptère.....	2
2.2. Dynamique de vol.....	3
Degrés de liberté.....	3
Gestion des déplacements à l'aide des moteurs	4
2.3. Navigation d'un quadricoptère	5
Télécommande	5
Aide à la stabilisation.....	6
3. Etude d'un premier prototype	7
3.1. Prise en main du prototype de Florian Nicolier.....	7
Structure	7
Composants	8
Câblage.....	13
Programmation	14
Type de régulation.....	15
3.2. Essais de vol.....	16
Premiers vols.....	16
Modifications code.....	16
Ajustements des trims et du gain.....	18
Résultats des modifications	18
3.3. Appréciation critique du premier prototype	19
Structure	19
Composants de base.....	20
Régulation et stabilisation	21
4. Etat de l'art du quadricoptère	22
4.1. Modèles prêt à voler	22
Drones de divertissements.....	22
Drones professionnels.....	23
Drones de loisirs	24

4.2.	Drones personnalisés.....	25
	Structures	25
	Régulateurs de vol.....	25
	Moteurs et Contrôleurs de vitesse électroniques	27
5.	Conception et étude d'un deuxième prototype.....	28
5.1.	Choix du deuxième prototype	28
	Critères de sélection du deuxième prototype.....	28
	Choix des composants	28
5.2.	FlameWheel 450	29
	Montage.....	30
5.3.	MultiWii Pro.....	31
5.4.	Logiciel	32
	Structure et principaux sous-programmes.....	33
	Initialisation.....	34
	GUI	35
5.5.	Actions contrôlées par les manches de commandes.....	39
5.6.	Modes de vols et PID	39
5.7.	Caractérisation du deuxième prototype	41
	Poids	41
	Poussée	42
	Durée de vol.....	45
6.	Application Finale	46
6.1.	Nouveau capteur	46
	Implémentation du capteur sur la carte de contrôle.....	47
	Caractérisation	48
6.2.	Décollage et atterrissage automatisés	50
	AltHold	50
	Décollage et atterrissage	52
7.	Conclusion	54
8.	Annexes	55
8.1.	Configuration de la carte MultiWii Pro	55
8.2.	Exporter le code sur la MultiWii Pro	56
8.3.	Annexe # MultiWii GUI.....	56
8.4.	Marche à suivre pour le décollage et l'atterrissage automatisé.....	57
8.5.	Charger les batteries	58

8.6.	Datasheet Capteur de forces ATI Mini40 SI-80-4.....	59
8.7.	Datasheet Sonar HC-SR04.....	61
8.8.	Stick Configuration & MultiWii Pro Pins.....	63

TABLE DES ILLUSTRATIONS

Figure 1 - Exemples de la diversité des quadricoptères.....	2
Figure 2 - Quad "+" et Quad "X"	2
Figure 3 - Système d'axes utilisés comme référentiels	3
Figure 4 - Sens de rotations des hélices pour chaque moteur.....	4
Figure 5 - Séquence d'actions amenant à un déplacement horizontal	5
Figure 6 - Structure du prototype de Mr. Nicolier.....	7
Figure 7 - Correspondance entre les pistes de la télécommande et du récepteur.....	8
Figure 8 - Carte contenant les gyromètres extraite d'une WiiMotion Plus.....	9
Figure 9 - Tension convertie de l'axe de lacet lorsque le gyromètre est immobile.....	10
Figure 10 - Tension convertie de l'axe de lacet lors de rotations dans les deux sens	10
Figure 11 - MSP430F2274 (à droite) et son connecteur USB.....	11
Figure 12 - ESC et moteur.....	12
Figure 13 - Batterie LiPo 2S 800 mAh	12
Figure 14 - Schéma de la distribution électrique du premier prototype.....	13
Figure 15 - Câblage des différents composants entre eux	13
Figure 16 - Ajustement de l'équilibre avant un vol.....	18
Figure 17 - Prototype de Mr. Nicolier en vol	19
Figure 18 - Fixation d'une hélice à l'aide d'un joint torique élastique.....	20
Figure 19 - Composants constituant le Kit DJI F450	29
Figure 20 - Les câbles d'alimentations sont soudés sur la "PowerBoard".....	30
Figure 21 - Schéma de la distribution électrique du deuxième prototype.....	30
Figure 22 - Schéma de branchements des câbles de signaux sur la MultiWii Pro	31
Figure 23 - Logiciel Arduino 1.0.5-r2.....	32
Figure 24 - Interface graphique MultiWii GUI 2.3	35
Figure 25 - Réglages des PID	35
Figure 26 - Paramètres de correction de la télécommande	36
Figure 27 - Modes de vol enclenchés en fonction de la position du potentiomètre.....	37
Figure 28 - Zone d'affichage des informations des capteurs	37
Figure 29 - Diverses informations sur l'état du quadricoptère	38
Figure 31 - Banc de mesure de la poussée.....	42
Figure 32 - Graphique de la poussée produite par les hélices 8"	43
Figure 33 - Graphique de la poussée produite par les hélices 10"	44
Figure 34 - Le deuxième prototype prêt à l'emploi	45
Figure 35 - Sonar HC-SR04.....	46
Figure 36 - Banc de mesure pour la réponse du Sonar	48
Figure 37 - Graphique de la réponse du Sonar en fonction de la distance.....	50

LISTE DES TABLEAUX

Tableau 1 - Appellation des mouvements pour différents degrés de liberté.....	4
Tableau 2 – Mesure des valeurs extrêmes des PWM de la télécommande.....	9
Tableau 3 - Branchements des câbles de signaux	13
Tableau 4 - Principaux modèles de quadricoptères de divertissement.....	23
Tableau 5 - Aperçu de quelques quadricoptères de prise de vue aérienne.....	24
Tableau 6 - Liste des quadricoptères de loisir les plus présent sur le marché	24
Tableau 7 - Principales structures disponibles sur le marché	25
Tableau 8 - Acteurs majeurs des régulateurs de vol	26
Tableau 9 - Caractéristiques du microprocesseur ATmega 2560.....	31
Tableau 10 - Liste des capteurs présents sur la carte MultiWii Pro	32
Tableau 11 - Capteurs sur lesquels les PID sont appliqués	36
Tableau 12 - Influence des paramètres sur la commande	36
Tableau 13 - Position des manches de commandes pour réaliser une action.....	39
Tableau 14 - Capteurs utilisés par les différents modes de vol	39
Tableau 15 - Explications de chaque mode de vol et réglages des PID	40
Tableau 16 - Détail de la masse de chaque composant	41
Tableau 17 - Caractéristiques du capteur de force ATI Mini40 SI-80-4.....	42
Tableau 18 - Donnée des 5 mesures de distance des deux sonars.....	49

1. INTRODUCTION

Depuis quelques années, le marché du drone civil est en pleine expansion. Le potentiel de ces systèmes aériens sans pilote a séduit plusieurs secteurs de l'industrie. Parmi ces drones, le quadricoptères se détache comme étant l'un des appareils les plus prometteurs de par la diversité des applications pour lesquels il peut être utilisé. Mission de surveillance et d'observation, prises de vue aériennes, contrôle de l'état d'un édifice difficilement accessible ou même transport de marchandise sont des filières à présents largement développées par les constructeurs.

Dans un premier temps, ce projet tente d'établir les caractéristiques indispensables au vol d'un tel appareil. En s'appuyant sur un premier prototype rudimentaire développé au cours d'un travail précédent ce rapport étudie la dynamique de vol particulière de ces drones mais aussi l'électronique embarquée et la régulation nécessaires au bon fonctionnement d'un quadricoptère.

Ensuite, ce travail cherche à concevoir une deuxième plateforme à l'aide des conclusions tirées du premier prototype mais également en examinant l'état de l'art actuel du marché du quadricoptère. Le développement de cette plateforme doit s'inscrire dans un objectif d'automatisation du quadricoptère.

Finalement, un premier pas vers un robot volant automatisé a été réalisé en assurant le décollage et l'atterrissage du modèle à l'aide d'un capteur de distance.

2. INTRODUCTION AUX QUADRICOPTÈRES

2.1. MÉCANIQUE DU QUADRICOPTÈRE

Un quadricoptère (ou quadcoptère) est un drone multirotor à quatre moteurs. Sa structure est constituée d'une partie centrale reliant entre eux quatre bras disposés en forme de croix. Les moteurs entraînant les hélices se situent aux extrémités de chaque branche. La plateforme centrale permet également d'accueillir les divers composants nécessaires au vol du drone. Les quadricoptères peuvent être constitués de toutes sortes de matériaux : des structures rudimentaires en bois ou en aluminium aux modèles plus élaborés en fibre de carbone ou en plastique. Elle doit être en même temps rigide et légère. La plupart des modèles comportent des pieds pour atterrir voir des protections pour les hélices. L'envergure des quadricoptères varie énormément. Certains modèles mesurent moins de 50 mm alors que d'autre dépassent 1 m.



Figure 1 - Exemples de la diversité des quadricoptères

On distingue deux types de quadricoptères, les modèles dit "+" ou "plus" et ceux appelés "X". La structure en elle-même ne change pas, c'est l'orientation de l'avant de l'appareil qui varie. Sur un Quad+, l'avant de l'appareil pointe dans la direction d'une des branches. Alors que pour un QuadX la structure est pivotée de 45°. L'avant se situe donc entre deux bras.



Figure 2 - Quad "+" et Quad "X"

Une même structure peut potentiellement voler dans les deux dispositions. Cependant la gestion des mouvements ne se fait pas de la même manière. Il faut donc que l'attribution des moteurs puisse être modifiée. Le mode "X" implique une réaction de chaque moteur pour tous les déplacements alors que pour certaines rotations le Quad+ n'en modifie que deux. Les performances de vol des deux modes ne présentent pas de grosses différences.

Le mode "+" facilite le pilotage du quadricoptère. Le bras avant peut être facilement mis en évidence par marquage distinctif (couleur différente ou LED par exemple) et permet une identification rapide de la position du quadricoptère.

Le mode "X" est presque toujours utilisé pour les modèles possédant une caméra fixe. Il est possible de filmer à l'avant du quadricoptère sans qu'un des bras ou qu'une hélice n'obstrue le champ de vision. Cela améliore la visibilité pour un vol en FPV¹ (*First Person View*).

2.2. DYNAMIQUE DE VOL

Le quadricoptère répond à une dynamique de vol bien particulière. Si sa mobilité dans l'espace est équivalente à celle d'un hélicoptère, la réalisation des déplacements est fondamentalement différente. En effet, c'est l'effet combiné des quatre moteurs qui permet au quadricoptère de s'orienter et de se mouvoir. Il est donc primordial de comprendre le fonctionnement de ce type de drone afin d'espérer le contrôler.

DEGRÉS DE LIBERTÉ

Le quadricoptère possède quatre degrés de liberté, une translation et trois rotations. On peut constituer un système d'axes orthogonaux ayant leur origine au centre de gravité de l'appareil. L'axe X pointe vers l'avant du quadricoptère (axe longitudinal), l'axe Y est perpendiculaire à X sur un plan horizontal (axe transversal) tandis que l'axe Z représente l'axe vertical.

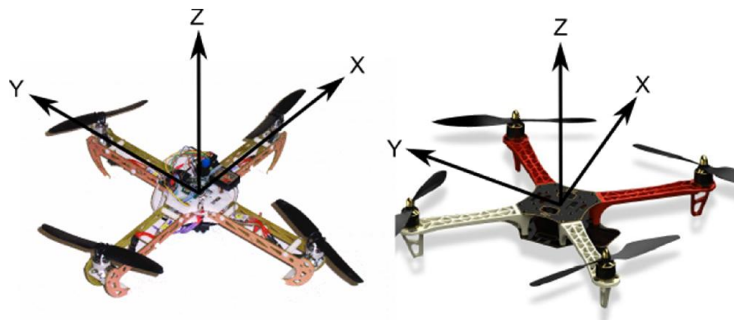


Figure 3 - Système d'axes utilisés comme référentiels

En aéronautique, on donne un nom à chacun de ces degrés de libertés. On retrouve fréquemment ces termes en anglais (en particulier dans le code de stabilisation), c'est pourquoi ils sont également présents dans le tableau suivant. Par la suite nous utiliseront les termes en français.

¹ Le vol FPV (*First Person View*) est le fait de diriger l'appareil à l'aide d'une caméra fixée sur celui-ci. Le pilote regarde un écran qui retransmet l'image prise par la caméra et n'a donc pas besoin d'avoir un contact visuel direct avec l'engin en vol.

Tableau 1 - Appellation des mouvements pour différents degrés de liberté

Degré de liberté	Mouvement en français	Mouvement en anglais
<i>Translation Z</i>	Poussée/Gaz	Throttle ²
<i>Rotation selon X</i>	Roulis	Roll
<i>Rotation selon Y</i>	Tangage	Pitch
<i>Rotation selon Z</i>	Lacet	Yaw

GESTION DES DÉPLACEMENTS À L'AIDE DES MOTEURS

Le quadricoptère possède quatre moteurs entraînant des hélices pour maintenir l'appareil en l'air. La poussée d'une hélice est directement liée à sa vitesse de rotation, une vitesse plus élevée produira plus de portance et inversement. Les rotors du quadcoptère sont fixes et pointent tous dans la même direction. A vitesse égale, ils ne produisent qu'une poussée selon l'axe Z. Cependant, en modifiant la vitesse d'un ou de plusieurs moteurs, il est possible de créer un déséquilibre temporaire contrôlé afin de modifier l'orientation de l'appareil. Le contrôle des différents degrés de liberté du quadricoptère se fait en réglant la vitesse de rotation de chaque moteur séparément.

Le déplacement vertical (translation selon Z) est le plus intuitif, tous les moteurs produisent la même poussée. En partant d'une poussée égale à la gravitation (l'appareil est stabilisé en altitude), une augmentation de la vitesse de rotation de tous les moteurs permet une ascension du modèle alors qu'une diminution induit une descente.

La rotation de lacet (selon l'axe Z) utilise le principe de contre-réaction d'une hélice en rotation. Chaque hélice en rotation induit un couple qui tend à faire tourner l'appareil dans le sens opposé. C'est un phénomène bien connu des hélicoptères, qui possèdent un rotor anticouple qui annule l'effet provoqué par rotor principal. Le quadricoptère adopte une configuration qui contre les effets des couples de réaction sans ajout d'une hélice supplémentaire. Il possède deux paires d'hélices au sens de rotation inversé. Chaque paire d'hélices tournant dans le même sens se situe sur deux bras opposés.

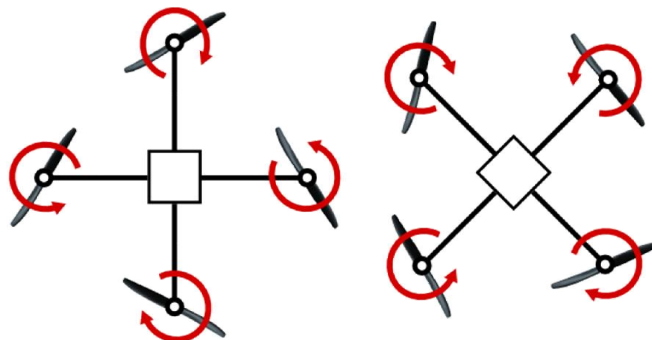


Figure 4 - Sens de rotations des hélices pour chaque moteur

² Littéralement accélération. Utilisé pour « gaz » ou « poussée » en aéronautique

En accélérant volontairement une des paires et en ralentissant l'autre, la différence de vitesse de rotation provoque un couple de contre-réaction entraînant une rotation de l'appareil selon l'axe de lacet.

Pour obtenir une rotation de roulis ou de tangage, il suffit d'augmenter la poussée du/des moteur(s)³ situés du côté inverse à la rotation souhaitée. Afin de ne pas influencer la poussée globale, l'accélération d'un moteur est toujours accompagnée de la diminution équivalente de son opposé. Par exemple, pour faire pivoter l'appareil sur la droite, il s'agit d'augmenter la vitesse de rotation du moteur gauche en diminuant celle du rotor droit. De même pour incliner le modèle vers l'avant, on augmente la poussée du moteur arrière alors que le moteur avant est ralenti. La différence de poussée induit un moment de force autour de l'axe de rotation. Dans une configuration en "X", chaque moteur est impliqué à la fois dans les rotations de roulis et de tangage.

Le quadricoptère ne possède pas intrinsèquement les degrés de libertés de translation dans le plan horizontal, mais ils peuvent être obtenus par l'addition d'une rotation et de la composante verticale. Lorsque le quadricoptère est incliné grâce à une rotation selon X ou Y, la poussée selon Z comprend une composante horizontale. En augmentant la poussée pour que la composante verticale soit égale à la force de gravité, le déplacement est purement horizontal.

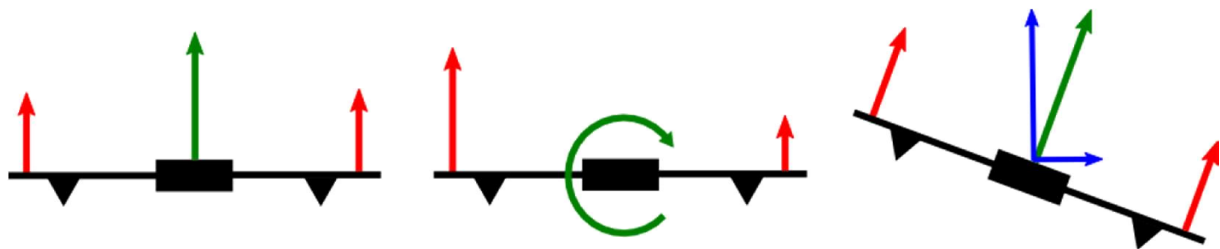


Figure 5 - Séquence d'actions amenant à un déplacement horizontal

2.3. NAVIGATION D'UN QUADRICOPTÈRE

Le guidage d'un quadricoptère nécessite l'utilisation d'un dispositif de commande afin que l'utilisateur puisse diriger l'appareil mais requiert également un système de stabilisation. La dynamique particulière du quadcoptère implique d'une part un système permettant de gérer les informations reçues par la commande et d'attribuer séparément à chaque moteur sa vitesse spécifique afin que le mouvement soit effectué correctement. D'autre part, la sensibilité d'un tel système aux perturbations rend le contrôle d'un tel appareil impossible sans l'aide d'une stabilisation indépendante.

TÉLÉCOMMANDE

Les quadricoptères sont des drones civils télécommandés. Ils nécessitent un pilote et donc un moyen de transmettre les commandes de ce dernier au modèle en vol. Une télécommande d'aéromodélisme est le plus souvent utilisée, même si certains modèles utilisent d'autres systèmes tels que des smartphones ou des ordinateurs. Les directives transmises par la radiocommande correspondent directement aux degrés de libertés du quadricoptère. Deux

³ Par la suite, seul le singulier sera utilisé. Le nombre de moteurs impliqué dépend de la configuration en "+" ou "X".

manches de commande⁴ à deux axes permettent de diriger l'appareil. Un des manches permet de contrôler la poussée (ou gaz) et le lacet. Le deuxième, commande l'avance et la dérive du quadricoptère. Une étude plus poussée de la télécommande utilisée pour ce projet est faite plus loin.

AIDE À LA STABILISATION

Un appareil en vol est soumis à des perturbations multiples, comme les turbulences créées par les hélices, le vent et les courants d'air ou les effets de sol par exemple. Le système est continuellement déstabilisé et un utilisateur, même confirmé, n'est pas assez réactif pour gérer toutes ces perturbations. De plus, les informations transmises par la télécommande ne peuvent être directement interprétées par les moteurs. Comme vu précédemment, un changement de direction nécessite une modification spécifique de poussée pour chaque rotor et donc une seule et même consigne ne peut être envoyée à tous les moteurs. Pour qu'un quadricoptère puisse voler, il doit comprendre un régulateur de vol (*Flight controller*). Ce système se compose d'un microprocesseur et de capteurs.

Le microcontrôleur reçoit en entrée les informations de l'utilisateur (commande) et les informations des capteurs embarqués (mesure). Il compare ces informations et applique les corrections sur chaque moteur pour que la mesure corresponde à la commande.

Les capteurs présents dans le régulateur de vol peuvent être nombreux. La configuration minimale comprend un système de gyromètres mesurant la vitesse angulaire sur les trois axes de rotations. Ce système détecte les petits mouvements angulaires et permet de corriger rapidement ces erreurs.

⁴ Les manches de commande (*joysticks* ou *sticks*) sont les manches à balais présents sur la télécommande. Ce sont des potentiomètres à deux degrés de liberté.

3. ETUDE D'UN PREMIER PROTOTYPE

3.1. PRISE EN MAIN DU PROTOTYPE DE FLORIAN NICOLIER

Lors d'un précédent travail de Bachelor, un étudiant avait conçu un prototype de quadricoptère. A la fin de son travail, Florian Nicolier avait été en mesure de présenter un modèle fonctionnel. Cependant, ce premier modèle, basique, ne permettait pas d'effectuer de vols équilibrés. La première partie de mon travail a donc consisté en l'étude du prototype existant afin d'améliorer la durabilité des vols, puis de déterminer les paramètres permettant une meilleure stabilisation en vue de la conception d'un deuxième prototype.

STRUCTURE

La structure proposée par Mr. Nicolier, de type "+", mesure 290 mm d'envergure entre deux moteurs opposés. Elle est constituée de pieds à double parois en résine pour circuit imprimé reliées entre elles par des entretoises de 22 mm en POM⁵. Les deux plaques centrales, en POM, maintiennent les pieds ensemble et permettent le support des divers composants du quadricoptère. Toutes les vis d'assemblage sont également en matière plastique. La structure complète sans les composants extérieurs pèse un peu plus de 100 g.

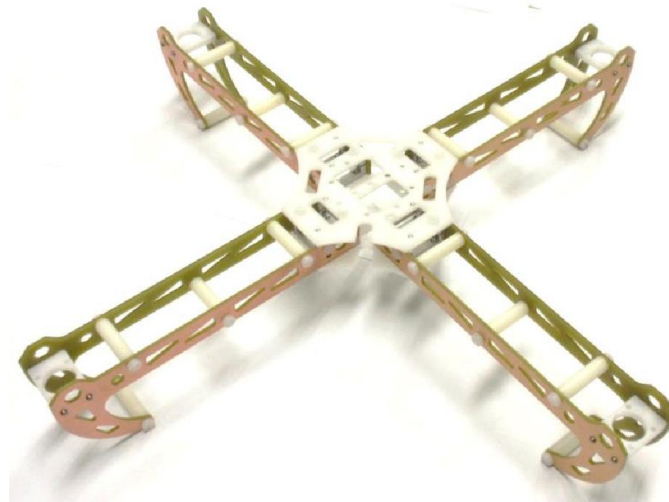


Figure 6 - Structure du prototype de Mr. Nicolier

⁵ Polyoxyméthylène, matière plastique de couleur blanche.

COMPOSANTS

Les éléments présents sur ce premier prototype sont les composants indispensables au vol d'un quadrirotor. Afin de comprendre le fonctionnement de ce quadricoptère chaque composant a été étudié. Les résultats sont présentés dans la partie suivante, en suivant le cheminement des signaux de la commande jusqu'à la source de propulsion.

TÉLÉCOMMANDE ET RÉCEPTEUR

La télécommande⁶ possède 6 potentiomètres, deux manches de commande à deux voies et deux potentiomètres rotatifs, qui permettent de transmettre 6 canaux d'information, par onde radio, à un récepteur placé sur le quadricoptère. Le récepteur⁷ en lui-même est muni d'une antenne de réception, d'une entrée « BAT » pour l'alimentation et des 6 sorties (*channel*). Chaque sortie est composée respectivement d'une pin pour la terre, d'une pin 5V et d'une pin pour le signal. Les pins d'alimentation sont reliées entre elles, il n'est donc pas nécessaire d'alimenter chaque piste. Dans notre cas, le récepteur est alimenté à l'aide de la sortie 5V de la Wii motion plus. Seules les pins « signaux » des sorties sont branchées au microprocesseur.



Figure 7 - Correspondance entre les pistes de la télécommande et du récepteur

Les canaux 1, 2 et 4 modifient les consignes de rotations (roulis, tangage et lacet). Les manches de ces trois degrés de liberté reviennent au centre lorsqu'ils sont relâchés. C'est leur position de repos. Le canal 3 (les gaz) quant à lui est cranté et n'a pas de position préférentielle. Les divers crans permettent de maintenir les gaz constants si l'utilisateur ne modifie pas sa position.

Les signaux de sortie du récepteur sont des PWM (En français : modulation de largeur d'impulsion), variant selon la position du potentiomètre. La période du cycle est de 22.2 milliseconde. Afin d'établir les valeurs limites de ces PWM une mesure à l'aide d'un oscilloscope a été effectuée sur chacun des canaux.

⁶ FlySky, FS-CT6B.

⁷ FlySky, FS-R6B.

Tableau 2 – Mesure des valeurs extrêmes des PWM de la télécommande

Canal	Période [ms]	Durée impulsion [ms]			Trim [ms]	Tension [V]
		Min	Milieu	Max		
1	22.15	1.20	1.55	3.33	±0.08	3.33
2	22.15	1.19	1.52 ±0.015	3.33	±0.07	3.33
3	22.15	1.14	1.52	3.33	±0.09	3.33
4	22.2	1.16	1.55	3.33	±0.07	3.33
5	22.15	1.02	/	3.33	/	3.33
6	22.15	1.02	/	3.33	/	3.33

Les quatre premiers canaux présentent des caractéristiques sensiblement similaires. Les impulsions varient entre 1,14 et 1,93 ms. Ces valeurs sont des valeurs typiques qu'on retrouve dans la plupart des radiocommandes. Cette normalisation permet d'utiliser une télécommande sans restrictions sur le modèle de quadricoptère utilisé.

La télécommande est un modèle bon marché. Cela se remarque par des différences de valeurs neutres des canaux de direction. La convention en modélisme propose des impulsions de 1.5 ms pour les positions de repos. Or, dépendamment de la position d'où le manche de commande est relâché, la valeur neutre peut varier de plusieurs microsecondes. Le canal 2 est particulièrement sujet à ce défaut. Des variations de 30 µs ont été mesurées.

A chaque degré de liberté des manches de direction est associé un compensateur (trim). Ce curseur permet d'incrémenter ou de décrémenter légèrement les valeurs du signal afin de donner un offset à la valeur du canal. Par exemple pour compenser une dérive récurrente de l'appareil.

GYROMÈTRES



Figure 8 - Carte contenant les gyromètres extraite d'une WiiMotion Plus

La puce, extraite de la Wii MotionPlus⁸, utilisée comme capteur de stabilisation, contient trois gyromètres. Chacun d'eux transmet la vitesse angulaire selon l'un des axes de rotation. Ce signal transmis est une tension continue qui augmente proportionnellement à la vitesse de rotation dans un sens, respectivement diminue proportionnellement à la vitesse de rotation dans l'autre

⁸ Accessoire Nintendo Wii, contenant un capteur gyroscopique 3 axes.

sens. La caractérisation d'un gyromètre, pour connaître sa résolution, sa linéarité ou la vitesse associée à une tension mesurée par exemple, nécessite un système rotatif permettant de contrôler précisément la vitesse de rotation du gyromètre. En l'absence d'un tel dispositif, quelques tests préliminaires ont été effectués afin d'appréhender les valeurs typiques transmises par les gyromètres. La réponse du gyromètre de l'axe des lacets à une rotation rapide⁹ fait varier la tension de sortie entre 1.1 et 1.6 V. La tension produite par le gyromètre est convertie par un convertisseur A/D du microprocesseur. Le convertisseur quantifie une valeur de 0 à 2.5 V sur 10 bits. En fonction des tensions mesurées, on constate que les valeurs extrêmes produites par le gyromètre n'utilisent qu'un intervalle allant de 44 à 64 % de la plage de mesure du convertisseur. Après la conversion A/D, le programme somme 16 valeurs consécutives du convertisseur et les divise par quatre. C'est ces valeurs qui sont utilisées par le programme, il est donc intéressant d'observer leur comportement. En extrayant ces valeurs, deux mesures ont été effectuées. La première sans mouvement du gyromètre et la seconde en effectuant des rotations lentes (de l'ordre de 90 [°/s]) dans les deux sens avec une position de repos avant chaque changement de sens.

Figure 9 - Tension convertie de l'axe de lacet lorsque le gyromètre est immobile

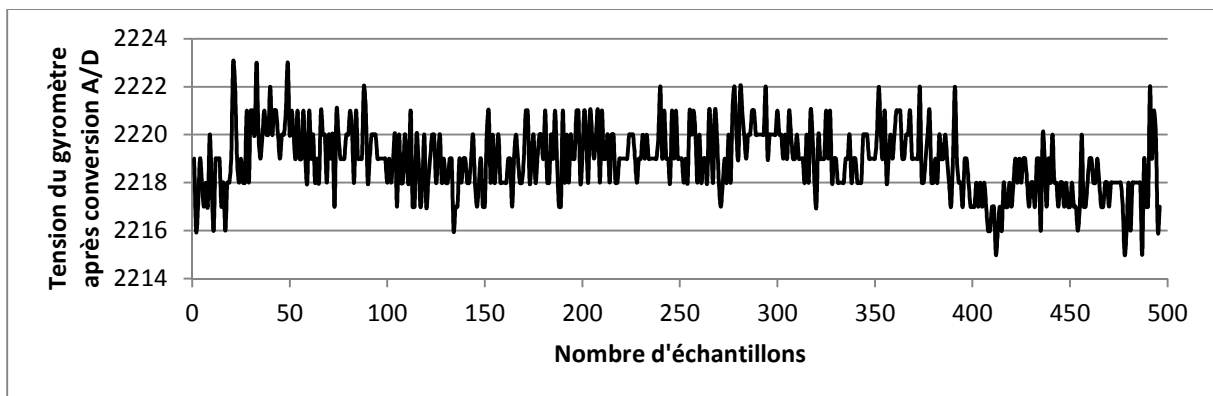
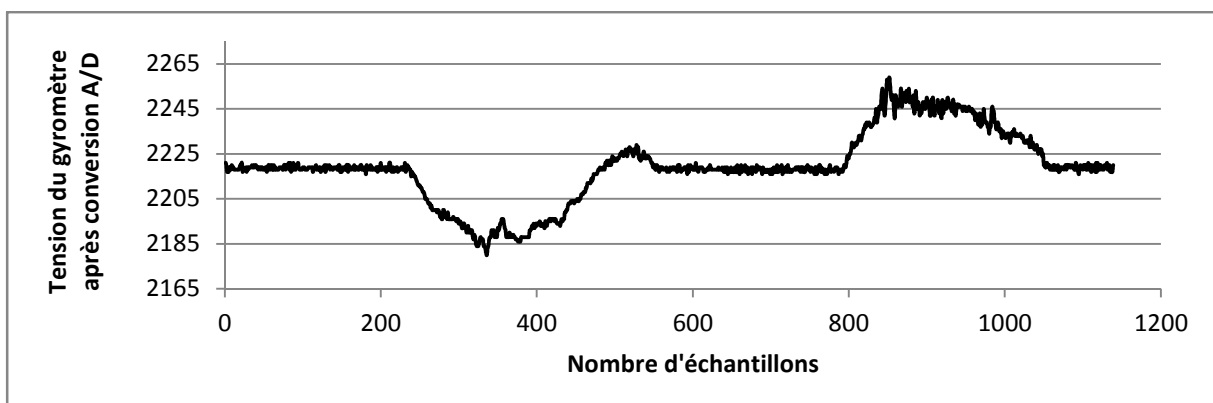


Figure 10 - Tension convertie de l'axe de lacet lors de rotations dans les deux sens



On constate qu'au repos, les valeurs utilisées par le microprocesseur ne sont pas totalement stables. Une variation de ± 4 a été mesurée. Lors d'une rotation qui demanderait une forte correction de la part du processeur, les variations atteignent ± 40 . Cela montre que malgré un lissage de la mesure, le bruit du capteur est toujours de 10 %.

⁹ La notion de « rapide » décrit une rotation qui dépasse le comportement usuel d'un quadricoptère. Si l'appareil était soumis à une telle rotation, le vol serait sans doute compromis.

MICROPROCESSEUR

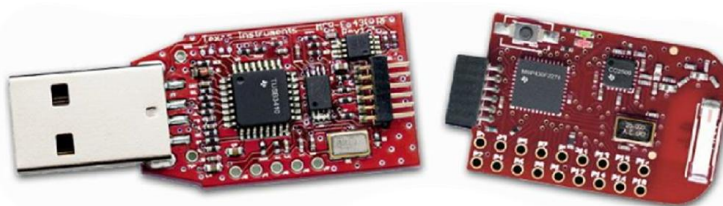


Figure 11 - MSP430F2274 (à droite) et son connecteur USB

Le microprocesseur utilisé dans ce prototype est un MSP430¹⁰. Il est utilisé pour la régulation du quadricoptère. Il reçoit en entrée les différentes commandes de mouvement provenant du récepteur ainsi que les informations envoyées par les gyromètres. Il effectue la conversion de ces différentes données en valeurs digitales puis les compare afin d'obtenir l'erreur selon chaque degré de liberté. Il génère finalement les PWM spécifiques à chaque moteur et les envoie aux ESC¹¹ par l'intermédiaire de sorties digitales. Ces PWM ont une fréquence de 200 Hz et des impulsions variant entre 1 et 2.2 ms. Les valeurs d'entrées (du récepteur) et de sorties sont traitées en microsecondes. Elles varient donc entre 1000 et 2200 μ s. Dans la suite du travail, la durée de l'impulsion d'un PWM sera traitée en tant que valeur digitale et ne comprendra pas d'unité.

CONTRÔLEURS BRUSHLESS ET MOTEURS

Les contrôleurs de moteur brushless (ESC) sont l'intermédiaire entre les signaux provenant du microcontrôleur et la vitesse de rotation effective du moteur. Les moteurs sont des moteurs brushless alimentés par trois phases. La rotation provient de l'alternance de ces phases. Plus la cadence d'alternance est élevée, plus le moteur tourne vite. Les ESC effectuent le basculement des phases pour atteindre la vitesse demandée. Elles utilisent en outre le principe de la régulation par retour de tension induite (*back EMF*), qui mesure la tension induite par la rotation du moteur et en déduit sa vitesse réelle, afin de corriger le basculement des phases. Pour assurer le bon fonctionnement des contrôleurs électroniques, il est important de les calibrer. En leur envoyant successivement la valeur de PWM correspondant à la vitesse maximale de rotation, puis le PWM le plus faible qui désigne l'arrêt du moteur, les ESC constituent un rapport proportionnel entre la durée du PWM et la vitesse de rotation. Le couple ESC-moteur sera considéré comme une « boîte noire » avec en entrée un PWM et en sortie une vitesse de rotation proportionnelle à la consigne.

¹⁰ Texas Instrument, MSP430F2274.

¹¹ *Electronic Speed Control*. Circuit électronique permettant de réguler la vitesse d'un moteur brushless en utilisant la tension induite du moteur pour interpréter sa vitesse réelle et la corriger.

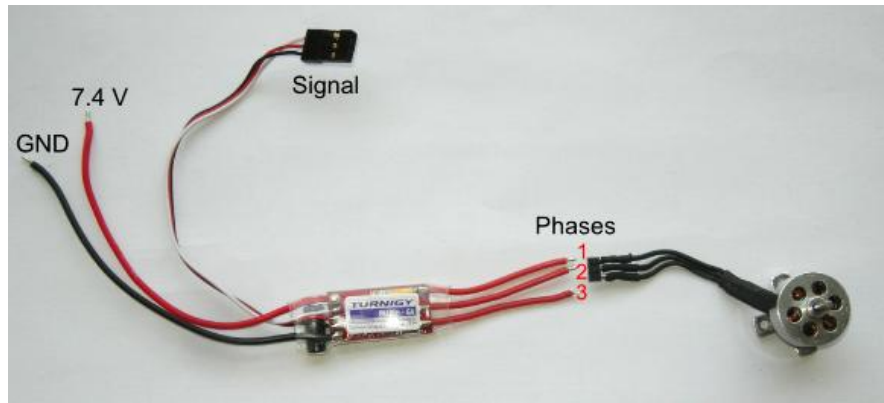


Figure 12 - ESC et moteur

Les ESC comprennent également un « BEC » qui permet de générer une tension de 5 V à partir d'une alimentation plus élevée. Les modèles réduits utilisent souvent des moteurs fonctionnant à des tensions plus élevées que les 5 V utilisés par la plupart des autres composants électroniques. Le BEC permet d'éviter de devoir embarquer plusieurs batteries de tensions différentes.

BATTERIE



Figure 13 - Batterie LiPo 2S 800 mAh

L'alimentation des divers composants est assurée par une batterie LiPo¹² d'une capacité de 800 mAh. Elle est constituée de deux cellules produisant une tension de 7.4 V. Ces batteries sont régulièrement utilisées en modélisme car elles sont plus légères que d'autres technologies. Elles doivent cependant être manipulées avec beaucoup de précautions, puisque elles présentent le risque de s'enflammer notamment lors du chargement ou en cas de court-circuit.

La batterie est reliée aux quatre ESC qui régulent les moteurs. Le BEC d'une des ESC est utilisé pour alimenter le récepteur et la carte Wii MotionPlus en 5 V. Cette dernière possède une sortie 3 V qui est utilisée pour fournir le courant nécessaire au microcontrôleur.

¹² Accumulateur Lithium Polymère.

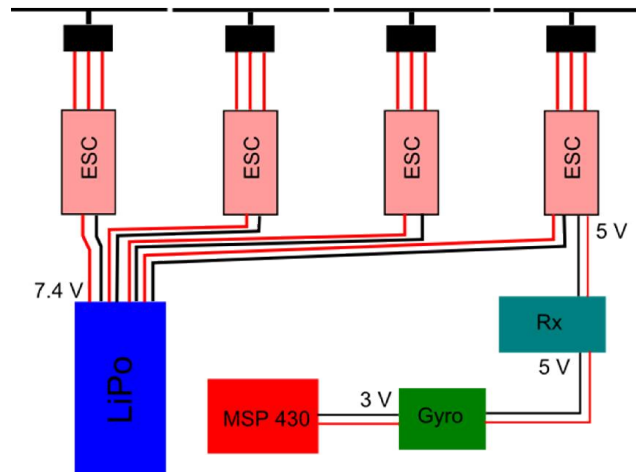


Figure 14 - Schéma de la distribution électrique du premier prototype

CÂBLAGE

Le microcontrôleur MSP430 est le centre névralgique des signaux. Chaque composant est relié à cette carte. Les codes couleur et les branchements des câbles provenant du récepteur et de gyromètre ainsi que ceux partant vers les moteurs sont détaillés si dessous.

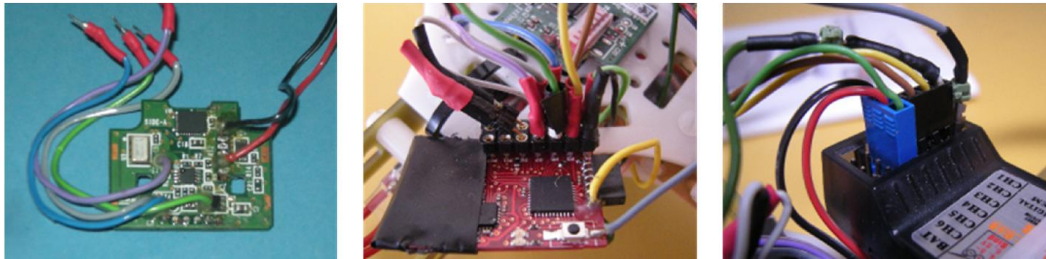


Figure 15 - Câblage des différents composants entre eux

Tableau 3 - Branchements des câbles de signaux

Signaux Récepteur	N° pin MSP430	Couleur du câble
Gaz	P3	Jaune
Tangage	Soudé sur la carte	Vert foncé
Roulis	Soudé sur la carte	Gris
Lacet	P4	Brun
AUX	P5	Vert clair
Signaux Gyromètres		
Tangage	P6	Bleu
Roulis	P8	Gris
Lacet	P7	Violet

Signaux Moteurs	N° pin MSP430	Couleur du câble
Avant	Soudé sur la carte	Jaune
Arrière	Soudé sur la carte	Rose
Gauche	P10	Blanc (connecteur noir)
Droite	P9	Blanc (connecteur rouge)

PROGRAMMATION

Le logiciel implanté dans le microcontrôleur est programmé en langage C. Il est constitué d'une boucle dans le programme principal et de plusieurs fonctions permettant l'initialisation et le traitement des informations relatives à un composant. Il ne s'agit pas ici de rentrer dans les détails du code mais de donner un résumé des fonctions importantes.

Le projet est composé de plusieurs fichiers *.c* contenant le code à proprement parler qui s'accompagne chacun d'un fichier *.h* contenant les définitions de variables globales et de fonctions.

Quad_main

Le fichier main contient la boucle principale qui, à l'aide d'une variable *woken_up_by*, permet la mise à jour régulière des différents paramètres par l'appel de diverses fonctions. La boucle se préoccupe notamment de la lecture des demandes tapées par l'utilisateur, de la conversion A/D, de la régulation ou de l'envoi des puissances de sortie pour chaque moteur. Il contient également les fonctions de moyennage de l'ADC (*process_adc_results*), de mise à zéro des gyromètres (*find_gyro_neutral_values*), de traitement des ordres utilisateur (*process_stick_input_commande*), de lecture de la mémoire flash (*read_config_from_flash*) et de contrôle des LED (*strobe_light*).

Cmd

L'utilisateur a la possibilité d'effectuer une série de demandes par l'intermédiaire d'un programme interagissant avec le MSP. La gestion de lecture de ce qui a été inscrit et les actions en découlant se fait dans le fichier *cmd.c* par la fonction *process_cmd*. Cela permet par exemple la calibration des ESC ou de faire des réglages fins sur des paramètres (correction d'un déséquilibre récurrent de l'appareil) mais aussi d'afficher des informations sur les variables. Le fichier contient également la fonction *show_info* qui retourne les valeurs actuelles et extrêmes des inputs avant la régulation, ainsi que la durée des impulsions des PWM transmis aux moteurs. Cette fonction est très utile pour la résolution de problèmes.

Captures

Cette procédure n'est constituée que d'une fonction qui permet de quantifier la longueur de la pulsation des PWM provenant de la télécommande. Elle est appelée par une routine d'interruption qui se déclenche à chaque flanc montant ou descendant d'un des 5 canaux de la télécommande et transcrit la durée de cette impulsion en nombre de « coup d'horloge ». Cette information est utilisée pour déterminer les déplacements souhaités par l'utilisateur.

Mixer

Ici, s'effectue la régulation à proprement parler. Il s'agit en fait de la compilation des diverses commandes et du calcul de l'action à attribuer à chaque moteur pour obtenir le mouvement désiré. La fonction *mix_mixing_quad* récolte les informations provenant de la télécommande et des gyromètres, puis constitue la puissance respective de chaque moteur. La vitesse de chaque moteur est ainsi modifiée lorsque les gyromètres indiquent une erreur par rapport à la position désirée ou lorsqu'un des manches de commande a été déplacé.

Hal

Ce sous-programme s'occupe de la gestion des « entrées/sorties » du microprocesseur. On retrouve les fonctions d'initialisation classique pour un MSP430 tels que l'attribution des ports en entrées ou sorties ainsi que le réglage de l'unité arithmétique et des horloges. Sont présentes également toutes les routines d'interruptions des entrées, la gestion du convertisseur A/D et l'envoi du PWM de chaque moteur en sortie.

Utils

Ce module est le centre linguistique du programme. Il englobe les fonctions permettant de lire ce que l'utilisateur écrit (détection des lettres et des sigles) et d'afficher des informations dans un format normalisé et compatible avec l'interface utilisée.

Types et Wireless

Types.h définit seulement les types de variables utilisées par le programme. Il est constitué que d'un fichier d'en-tête. Wireless quant à lui permet d'utiliser la communication radio incluse dans le microcontrôleur. Cette fonction n'a pas été utilisée dans ce prototype.

TYPE DE RÉGULATION

Le but de la régulation du quadricoptère est de contrer les effets des perturbations en permettant au pilote de diriger le modèle de façon stable. On distingue trois types d'informations présentes dans la phase de contrôle. La poussée, les rotations souhaitées et les mesures de rotation effective.

La poussée est uniquement contrôlée par le pilote à l'aide du manche des gaz. Elle s'applique à tous les moteurs de façon équivalente et définit si l'appareil monte, descend ou vole de façon stationnaire.

Les rotations souhaitées sont les instructions de direction données par le pilote. Lorsque l'utilisateur ne touche pas aux manches de direction, le quadricoptère devrait rester dans la même position. On définit alors, pour chaque rotation, une valeur neutre correspondant au signal de la commande lorsque les manches de direction sont centrés. La différence entre le signal provenant de la télécommande et cette valeur neutre est proportionnelle au déplacement du manche de commande et donc à l'inclinaison souhaitée.

La mesure provenant des gyromètres est également comparée à une valeur neutre, obtenue lors de la calibration du capteur, correspondant à la tension retournée par le capteur lorsqu'aucune rotation ne s'applique au quadricoptère. Cette résultante est multipliée par un gain défini par l'utilisateur pour lui donner plus ou moins d'importance.

Les valeurs des gyromètres et de la commande sont comparées respectivement pour chaque rotation. Cela donne un facteur de correction à appliquer à la consigne des gaz des moteurs concernés par la rotation. Si le gain est trop faible, la mesure des gyromètres n'aura aucune influence sur le comportement des moteurs. Si au contraire il est trop élevé, les valeurs de la commande seront surpassées par la mesure et le pilote ne pourra plus diriger l'appareil.

3.2. ESSAIS DE VOL

Le seul moyen de tester un quadricoptère est de le faire voler. L'observation de son comportement en vol ainsi que l'aisance du guidage donnent des informations qualitatives sur la bonne marche du prototype. L'appréciation du vol est parfois difficile à exprimer puisqu'elle repose uniquement sur la vision et le ressenti du pilote.

PREMIERS VOLS

Les premiers essais ont montré un comportement très instable rendant tout vol impossible. Une inégalité dans la poussée des quatre moteurs voire parfois l'arrêt soudain de l'un d'entre eux. Des problèmes de connexions ont tout d'abord été mis en évidence, certains branchements avaient tendance à se désolidariser durant le vol dû aux vibrations ou à certains à-coups. Une fois les branchements renforcés, l'entrave au vol la plus significative semblait être un mauvais équilibrage de la poussée entre les quatre moteurs. Trois paramètres peuvent expliquer ce déséquilibre. Le premier vient de la valeur neutre de chaque signal de direction. De légères fluctuations des signaux transmis par la télécommande ont été mises en évidence, de ce fait la comparaison d'un tel signal avec une valeur fixe peut entraîner une erreur dans l'appréciation de la commande. Le deuxième facteur découle des caractéristiques de chaque moteur. Il n'est pas possible d'avoir quatre moteurs exactement similaires, une erreur systématique pouvant être induite par un moteur plus performant. Finalement les hélices également peuvent être un facteur de déséquilibre. A vitesses de rotation égales, certaines hélices produisent plus de poussée que d'autre. L'utilisation des trims de la télécommande ont permis d'ajuster ces disparités de fonctionnement.

MODIFICATIONS CODE

L'étude des informations transmises par la télécommande a mis en évidence des fluctuations des valeurs centrales des manches de direction. En vol le logiciel compare les valeurs de la télécommande avec une valeur centrale prédéfinie dans une constante. Si les valeurs effectives des positions centrales des manches diffèrent de la valeur prédéfinie, le logiciel interprète à tort une demande de rotation de l'utilisateur. Après plusieurs modifications des valeurs neutres, les constantes des valeurs centrales ont été remplacées par des variables qui sont mises à jour à chaque calibration des gyromètres. Ainsi les valeurs de comparaisons sont celles effectivement envoyées par la télécommande lors de la calibration. Des dérives répétées ont été observées même après cette modification. Elles proviennent vraisemblablement d'une différence de poussée entre les moteurs. Les rotations indésirables étant toujours orientées dans le même sens, une valeur de compensation a été ajoutée sur les valeurs neutres de l'axe de tangage et de lacet.

Dans le fichier mixer.h

```

58 //define RUD_CENTER_PULSE_VAL      1564
59 //define PIT_CENTER_PULSE_VAL      1508
60 //define AIL_CENTER_PULSE_VAL      1567
61
62 extern UINT16 ail_center_pulse_val;
63 extern UINT16 pit_center_pulse_val;
64 extern UINT16 rud_center_pulse_val;

```

Dans le fichier mixer.c

```

90 //INT32 ail_cmd = ail - AIL_CENTER_PULSE_VAL ;
91 //INT32 pit_cmd = pit - PIT_CENTER_PULSE_VAL ;
92 //INT32 rud_cmd = rud - RUD_CENTER_PULSE_VAL ;
93
94 INT32 ail_cmd = ail - ail_center_pulse_val ;
95 INT32 pit_cmd = pit - pit_center_pulse_val ;
96 INT32 rud_cmd = rud - rud_center_pulse_val ;

```

Dans le fichier quad_main.c

```

282
283     ail_center_pulse_val = ail_pulse;
284     pit_center_pulse_val = pit_pulse + 50;
285     rud_center_pulse_val = rud_pulse + 100;

```

Une inversion de la commande de l'axe de lacet a également été constatée. Lorsque le manche est incliné vers la droite, le quadricoptère tourne autour de l'axe vertical dans le sens antihoraire. Le signe des valeurs de la commande de lacet a donc été inversé.

Dans le fichier mixer.c

```

166     if (thr < 1200)
167     {
168
169         *fr = thr - (pit_cmd) - yaw + (INT16)(pitch_gyro_rate);
170         *bk = thr + (pit_cmd) - yaw - (INT16)(pitch_gyro_rate);
171         *al = thr - (ail_cmd) + yaw - (INT16)(roll_gyro_rate);
172         *ar = thr + (ail_cmd) + yaw + (INT16)(roll_gyro_rate);
173     }
174     else
175     {
176         thru = palier + (((UINT32)(thr-thrmin)*factor)>>8);
177
178         *fr = thru - (pit_cmd) - yaw + (INT16)(pitch_gyro_rate);
179         *bk = thru + (pit_cmd) - yaw - (INT16)(pitch_gyro_rate);
180         *al = thru - (ail_cmd) + yaw - (INT16)(roll_gyro_rate);
181         *ar = thru + (ail_cmd) + yaw + (INT16)(roll_gyro_rate);
182     }
183     /*
184     if (thr < 1200)
185     {
186
187         *fr = thr - (pit_cmd) + yaw + (INT16)(pitch_gyro_rate);
188         *bk = thr + (pit_cmd) + yaw - (INT16)(pitch_gyro_rate);
189         *al = thr - (ail_cmd) - yaw - (INT16)(roll_gyro_rate);
190         *ar = thr + (ail_cmd) - yaw + (INT16)(roll_gyro_rate);
191     }
192     else
193     {
194         thru = palier + (((UINT32)(thr-thrmin)*factor)>>8);
195
196         *fr = thru - (pit_cmd) + yaw + (INT16)(pitch_gyro_rate);
197         *bk = thru + (pit_cmd) + yaw - (INT16)(pitch_gyro_rate);
198         *al = thru - (ail_cmd) - yaw - (INT16)(roll_gyro_rate);
199         *ar = thru + (ail_cmd) - yaw + (INT16)(roll_gyro_rate);
200     }

```

Le gyromètre de l'axe de tangage a également été inversé. Il avait en effet tendance à amplifier une perturbation selon cet axe au lieu de la contrebalancer. Une variable dans le code permet simplement de changer le signe du signal produit par le gyromètre.

Dans le fichier quad_main.c

```
378   copter_config_data.gyro_dir ^= PITCH_GYRO_DIR;
```

AJUSTEMENTS DES TRIMS ET DU GAIN

Afin de pouvoir équilibrer la poussée de chaque moteur sans que le modèle ne tombe, la méthode suivante a été adoptée. Le quadricoptère est maintenu en l'air à la main, puis la poussée des moteurs est augmentée jusqu'à ce qu'elle soit suffisante pour soutenir l'appareil. Les valeurs des trims de directions sont ensuite modifiées afin que la main n'ait plus besoin de retenir le quadricoptère. Enfin le modèle est relâché temporairement afin de constater une éventuelle dérive résiduelle puis ramené manuellement. Les trims sont le cas échéant à nouveau légèrement modifiés pour que la dérive du quadricoptère soit presque nulle. Cette méthode est utilisée avant chaque vol, car il a été constaté que les réglages pouvaient différer en fonction de la charge de la batterie.



Figure 16 - Ajustement de l'équilibre avant un vol

Lors du décollage depuis le sol, il est important d'augmenter rapidement les gaz afin de subir les effets de sol le moins longtemps possible. Lorsque le modèle est sur le point de décoller, un léger décentrage du centre de masse peut provoquer l'inclinaison du modèle. A faible altitude et avec encore une partie des pieds du quadricoptère au sol, les gyromètres ne sont pas capables de corriger ce déséquilibre.

Quelques vols, de quelques dizaines de secondes, ont ensuite pu être effectués. Une valeur pour le gain des gyromètres a été trouvée empiriquement. Un gain de 488 représente le meilleur compromis trouvé pour un vol stable qui ne handicape pas la navigation du prototype.

RÉSULTATS DES MODIFICATIONS

Les différentes modifications ont permis des vols de près d'une minute. Le prototype peut être maintenu relativement aisément en position stationnaire et des déplacements à faible vitesse sont possibles. Le pilote reste tout de même le facteur principal d'un vol réussi. Il doit corriger en permanence la dérive du modèle et son maintien à l'horizontale. Un vol plus long semble tout à fait envisageable pour un utilisateur plus expérimenté. L'exiguïté des locaux de test a également été un facteur très limitant. Il était très difficile de déplacer le quadricoptère sans risque de collision. Ce type de prototype est destiné à un usage intérieur, sa petite taille et la régulation sommaire rendant un vol extérieur périlleux. Dehors l'appareil est soumis à beaucoup plus de perturbations et un modèle aussi léger est particulièrement vulnérable.



Figure 17 - Prototype de Mr. Nicolier en vol

La phase de réglage des trims est relativement contraignante puisqu'elle diminue grandement l'autonomie de l'appareil. Cependant elle permet de corriger plusieurs facteurs d'erreurs systématiques. Une radiocommande de meilleure qualité ainsi qu'une caractérisation des couples moteurs-hélices permettrait de faciliter grandement la mise en service de l'appareil.

3.3. APPRÉCIATION CRITIQUE DU PREMIER PROTOTYPE

Le bon fonctionnement d'un quadricoptère repose sur plusieurs facteurs d'origine mécanique mais également sur une régulation performante. L'étude de ce premier prototype a permis de déterminer quelques-uns de ces paramètres, mais l'absence d'un système de test fiable pour le système en vol ne permet parfois pas de connaître l'origine d'un comportement défaillant. Pour une analyse quantitative et non qualitative, il s'agirait de constituer un banc de test qui n'entraverait aucun des degrés de liberté du quadricoptère.

STRUCTURE

Le principal défaut de la structure de ce premier prototype est son manque de rigidité. Afin de réduire au maximum les vibrations de l'appareil, une liaison plus rigide entre les pieds et les plaques centrales devrait être conçue. Une structure trop souple peut provoquer des oscillations internes qui pénalisent grandement tant la mesure des gyromètres que la capacité de poussée du quadricoptère.

Un bilan de masse du prototype a également été réalisé. Les pieds ont un poids élevé par rapport au poids total de l'appareil. Une diminution de la masse des pieds associée à une meilleure fixation sur les plaques centrales contribuerait à améliorer les performances dynamiques du modèle.

La carte Wii MotionPlus, contenant les gyromètres, est fixée de façon rigide sur la plaque supérieure. Les vibrations de la structure sont donc directement transmises au capteur. Les gyromètres sont très sensibles aux vibrations qui représentent des rotations parasites. Une fixation amortissante serait plus appropriée et améliorerait la mesure des gyromètres.

La taille du quadricoptère est adaptée à un vol d'intérieur mais un modèle plus petit permettrait un déplacement plus agile dans un milieu encombré. Si le prototype veut être utilisé à l'air libre, une envergure supérieure permettrait une meilleure stabilité et la possibilité d'utiliser des rotors plus puissants.

COMPOSANTS DE BASE

Le prototype a été conçu avec des composants d'entrée de gamme. Ils permettent de réduire le coût de fabrication d'un tel modèle mais altèrent également la capacité du drone à voler de façon stable.

Les gyromètres à sortie analogique produisent inévitablement du bruit qu'il est difficile de filtrer. S'ajoute à ça la conversion A/D du microprocesseur. L'utilisation de gyromètres digitaux permettrait d'augmenter le rapport signal sur bruit afin d'obtenir une mesure plus précise.

Les moteurs eux aussi sont un facteur important d'un vol stable. Des moteurs aux caractéristiques les plus proches possible sont souhaitables pour une diminution des écarts de poussées. Un moteur dont l'inertie du rotor est légèrement déséquilibré par rapport à son axe de rotation produit des vibrations suffisantes pour perturber les gyromètres. Des moteurs spécialement dédiés aux quadricoptères réduisent au maximum ces deux facteurs de perturbations.

Les hélices sont fixées sur les moteurs à l'aide de joints toriques élastiques. Si cette fixation suffit à éviter que l'hélice ne se désolidarise du moteur, elle procure une certaine flexibilité à la liaison qui n'est pas souhaitable.



Figure 18 - Fixation d'une hélice à l'aide d'un joint torique élastique

Une télécommande de qualité supérieure contient des potentiomètres plus performants et les manches de directions ont un système de retour à la position neutre plus précis. Ce qui permet d'éviter des différences de signal lors d'un retour à la position initiales.

RÉGULATION ET STABILISATION

La régulation du premier prototype est très sommaire. Une rotation mesurée par le capteur engendre une modification de la vitesse des moteurs dont l'intensité dépend du gain. Plus le gain est grand, plus la correction sera importante. Ce gain étant le même pour les trois rotations, il ne tient pas compte des différentes inerties en jeu. Typiquement, la rotation de lacet, qui utilise un principe de mouvement différent des deux autres rotations, demande un gain plus élevé que les deux autres. Ainsi un gain adéquat aux rotations de roulis et tangage, sera insuffisant pour le contrôle de l'orientation du quadricoptère dans le plan. De plus la modification du gain des gyromètres influence directement la capacité de commande de l'appareil. Un grand gain donne de grandes valeurs de compensation, la valeur de la commande devient alors négligeable par rapport à cette modification. Une régulation plus poussée incluant une composante intégrale et une composante dérivée améliorerait également l'équilibre du modèle.

La stabilisation pourrait également être améliorée. Aucune notion de position n'est utilisée dans ce prototype. Seule la vitesse des rotations est mesurée. Si le quadricoptère se stabilise, après une perturbation, dans une position inclinée, les gyromètres détecteront une vitesse nulle et donc n'appliqueront pas de correction. Cependant l'appareil étant penché, il se déplacera. C'est à l'utilisateur de ramener le modèle à l'horizontale afin de le stabiliser. Pour obtenir une position, il suffirait théoriquement d'intégrer la vitesse de rotation. Cependant un gyromètre présente une dérive relativement importante¹³, rendant son utilisation pour une régulation de position impossible. La stabilisation de position d'un drone nécessite l'ajout d'accéléromètres et de filtres combinant les deux informations.

D'autres composants peuvent également faciliter la tâche du pilote. Il ne s'agit plus à proprement parler d'améliorer la stabilisation mais de permettre un vol mieux contrôlé. Pour maintenir une altitude et une position dans le plan constante par exemple.

¹³ Un gyromètre est utilisé en mode différentiel, il mesure une variation de la vitesse de rotation. Si la vitesse est constante sur une durée importante, le capteur tendra à considérer cette vitesse comme nulle.

4. ETAT DE L'ART DU QUADRICOPTÈRE

Depuis la conception du premier prototype, le drone domestique a connu un développement exponentiel. En particulier les appareils multi-rotors, des drones possédant entre trois et huit hélices. Autrefois réservé aux passionnés et aux experts, le multicoptère est à présent utilisé pour une infinité d'application. Du jouet radiocommandé aux drones de cinéma en passant par les quadricoptères autonomes, de prise de vue événementielle ou à application commerciale.

La diversité des multi-rotors, même en ne se concentrant que sur les modèles à quatre moteurs, rend une liste exhaustive impossible. Cette section tente de classer les différents drones afin de trouver le type d'appareil se prêtant le mieux à l'application désirée pour ce travail de bachelor.

4.1. MODÈLES PRÊT À VOLER

Les quadricoptères de cette catégorie sont des appareils livrés complètement fonctionnels. Les modèles d'entrée de gamme incluent également la télécommande. Tous les composants ont été appariés et montés. Il est donc possible de les utiliser sans connaissance particulière dans le domaine.

DRONES DE DIVERTISSEMENTS

Ces quadricoptères bas de gamme se rapprochent plus du jouet que du drone. Ils s'apparentent aux petits hélicoptères ou avions radiocommandés. Ils sont d'ailleurs la plupart du temps produits par des entreprises spécialisées dans les hélicoptères RC. Faciles à piloter, légers et capables de résister à plusieurs chutes, ils possèdent une stabilisation relativement simple composée de gyromètres 3 axes voir d'accéléromètres pour les modèles les plus chers et sont à utiliser de préférence à l'intérieur. Ils mesurent entre 5 et 30 cm et sont disponibles pour un prix compris entre une dizaine et une centaine de francs. Certains modèles comprennent même une caméra. Bien que peu coûteux, ces drones volent tout à fait correctement. Ils ont néanmoins une autonomie très limitée et ne permettent aucune modification.

Le tableau ci-dessous présente une sélection des acteurs principaux du marché. Certaines marques proposent plusieurs modèles relativement semblables, les modèles les plus populaires sont donc cités à titre indicatif.

Tableau 4 - Principaux modèles de quadricoptères de divertissement

Marque	Modèles	Prix [\$]
Blade	Nano QX, 180 QX	90-180
Syma	X5, X6	60-80
AirHogs	Helix X4 Stunt	80
Walkera	W100, LadyBird, HotenX	100-300
Hubsan	Q4, X4	50-70
Traxxas	QR-1	60
Estes	Dart, ProtoX	40-80
Heli-Max	1SQ	100
Udi R/C	U816A, U818A	40-90

DRONES PROFESSIONNELLS

Dans cette catégorie se trouvent les quadrirotors qui sont optimisés pour la prise de vue aérienne. Largement dominé par des drones constitués de 6 ou 8 rotors, pour pouvoir supporter de grandes charges, on retrouve cependant quelques quadricoptères dans ce secteur. Assez imposants, ils possèdent une très bonne stabilisation et sont capables de supporter une charge importante, allant jusqu'à quelques kilos, pour embarquer une caméra de bonne résolution qui est indépendamment stabilisée et orientée. Les quadricoptères contribuant à la recherche, par exemples les drones de l'ETHZ¹⁴, les drones qui construisent un mur¹⁵ ou capable de voler en formation¹⁶, peuvent également être rangés dans cette catégorie. Cependant ces modèles sont dirigés de manière automatisée par des programmes s'aidant de caméras 3D pour repérer le quadricoptère dans une pièce. Ces drones sont très spécifiques et ne seront pas étudiés plus en détail.

Les modèles professionnels sont proposés sur des sites spécialisés dans la prise de vue aérienne. Ces sites proposent des modèles prêts à l'emploi, avec une structure personnalisée suivant l'application et des composants standard provenant souvent d'autres entreprises. Le prix est très élevé (de 1'500 à 15'000 \$) et varie fortement suivant la taille et les capacités du modèle (support de caméra orientable, gestion indépendante de la prise de vue, charge admissible). D'envergures imposantes, ils se distinguent des quadcoptères de loisirs en favorisant la stabilité par rapport à l'agilité.

Le tableau suivant offre un aperçu des quadricoptères spécialisés dans la prise de vue. La marque indiquée est celle qui commercialise le modèle, les composants provenant d'autres entreprises ne sont pas détaillés, sauf pour le stabilisateur de vol.

¹⁴ <http://www.flyingmachinearena.org/>

¹⁵ <http://www.gramazikohler.com/web/d/projekte/209.html>

¹⁶ https://www.grasp.upenn.edu/success_story/swarm_nano_quadrotors

Tableau 5 - Aperçu de quelques quadricoptères de prise de vue aérienne

Marque	Modèle	Stabilisation	Prix [\$]
Hoverfly	VENU	HoverflyPRO	-
OnlyFlyingMachine	Hunter 700, Jumper 600	Xaircraft, Dji	1'500-5'500
Droidworx	CX-600, BOT	Dji, Hoverfly, APM	5'000-10'000
SteadyDrone	QU4D	APM, Dji	2'000-15'000
FlyingEye	QuadCopter	-	10'000

DRONES DE LOISIRS

La dernière catégorie est la plus étendue, elle est souvent nommée « sport ». Elle se distingue par des drones de taille moyenne (entre 30 et 60 cm d'envergure) qui optent pour une certaine maniabilité par rapport au drone professionnel mais offrent plus de possibilités que les drones de divertissement notamment du point de vue de l'autonomie et de la diversité des applications.

Comme pour les quadcoptères de loisir, on trouve des modèles vendus avec une télécommande, prêt à voler. Ils possèdent néanmoins une stabilisation plus avancée et une caméra intégrée ou un support pour en ajouter une. Les quadricoptères présents dans le tableau qui suit sont les modèles les plus populaires.

Tableau 6 - Liste des quadricoptères de loisir les plus présent sur le marché

Marque	Modèle	Spécificités	Temps de vol [min]	Prix [\$]
Dji	Phantom 2	LED, caméra Phantom Vision, GPS	25	680
Parrot	AR.Drone 2.0	Caméra, commande par smartphone, GPS, capteur ultrason	12	300
Blade	350QX	Support caméra, GPS	7-10	470
Walkera	QR X350	Wifi, FPV, GPS	10-15	300
Hubsan	X4 FPV	FPV sur la télécommande	7	200
Heli-Max	1Si	LED, Caméra	-	100

4.2. DRONES PERSONNALISÉS

Il est également possible de construire un drone à partir de pièces détachées. Les composants indispensables pour le vol d'un quadcoptère sont facilement accessibles. Chaque composant doit être choisi de manière soignée et comporte un certain nombre de défis. La liste qui suit présente les composants indispensables ainsi que les principaux acteurs du marché.

STRUCTURES

Les structures disponibles sur le marché résultent d'une série de mutations des premiers quadricoptères. Elles sont donc plus performantes que les structures maison notamment quant aux problèmes de vibrations, de résistance et de dynamique de vol. D'abord en bois ou en aluminium, les modèles sont à présents fabriqués en plastique moulé, permettant des prix très avantageux, ou en fibres de carbone, matériaux léger et résistant. Il existe plusieurs copies des modèles de structures les plus connus. Ces génériques sont sensiblement identiques aux modèles d'origine mais sont constitués de matériaux moins résistants afin de diminuer le coût.

Tableau 7 - Principales structures disponibles sur le marché

Marque	Modèle	Spécificités	Dimensions [mm]	Prix [\$]
Dji	Flame Wheel F330, F450	Structure robuste, plastique moulé	330, 450	30
Hoverthings	Flip FPV, Flip sport	Pliable, Compartiment pour GoPro, résistant	375, 385	90-120
Parallax	ELEV-8	Leger, grande envergure	660	180
Tarot	Iron Man 650	Leger, grande envergure, carbone	650	150
Action Drone	AD1, AD1 FG	Pliable, Fibre de carbon, Fibre de verre	-	400-460
GAUI	350X, 500X	Résistant, capsule de protection	350, 500	30

RÉGULATEURS DE VOL

On trouve sur le marché deux types de régulateur de vol. Les kits « fermé » qui sont composés de plusieurs composants connectés entre eux, produit par la même marque et se comportant comme une « boîte noire ». Seul l'entrée et la sortie sont accessibles, les composants ne sont pas visibles et la régulation est faite par un logiciel propriétaire. La deuxième catégorie englobe les cartes open source. Le code est libre, même s'il n'est souvent disponible qu'après l'achat de la carte correspondante. Ce type de régulateur est particulièrement intéressant car il est possible de modifier le code, bien que ce dernier soit tout à fait fonctionnel d'origine, pour améliorer les performances de stabilisation. Ces cartes open source se basent majoritairement sur un Arduino mais également sur MSP430, STM32 ou RaspberryPI par exemple.

Tableau 8 - Acteurs majeurs des régulateurs de vol

Marque	Type	Spécificité	Prix [\$]
<i>DJI</i>	Fermé	Leader du marché	150 - 1300
<i>Hoverfly</i>	Fermé	Destiné aux hobbyistes et aux professionnels	225 - 450
<i>Xaircraft</i>	Fermé	Destiné aux professionnels	400 - 600
<i>Feyu Tech</i>	Fermé	Milieu de gamme	100 - 200
<i>Optimus</i>	Fermé	Bas de gamme	45
<i>MultiWii</i>	Open source	Précurseurs du quadricoptère open source	20 - 65
<i>ArduPilot</i>	Open source	Principal concurrent de MultiWii	150 - 250
<i>OpenPilot</i>	Open source	Processeur 32 bits	60 - 200
<i>AeroQuad</i>	Open source	Orienté vers DIY	100 - 150

Dans ces deux classes de régulateur on retrouve un certain nombre de capteurs permettant la stabilisation du quadrirotor. Chaque capteur permet une meilleure stabilisation ou une fonctionnalité en plus dans le contrôle du drone. Voici la liste des composants qu'on retrouve fréquemment dans les régulateurs actuels. D'autres capteurs peuvent être ajoutés mais ne figurent pas par défaut dans les contrôleurs de vol.

GYROMÈTRE

Le gyromètre est un capteur qui mesure la vitesse angulaire. Sur les quadcoptères il mesure la vitesse de rotation autour d'un des axes de direction qui peut être ensuite comparée à la consigne souhaitée. C'est le seul capteur indispensable pour la stabilisation.

ACCÉLÉROMÈTRE

L'accéléromètre mesure l'accélération de l'objet sur lequel il est placé. Les quadrirotors sont équipés d'un accéléromètre 3 axes qui permet de déterminer l'accélération du modèle selon les 3 degrés de liberté de translation. L'accéléromètre est également capable de mesurer l'inclinaison d'un objet car il est influencé par la force de gravité. Combiné aux gyromètres et après application d'un filtre, ces capteurs permettent d'obtenir la position du quadricoptère dans l'espace à l'exception de la dimension verticale.

BAROMÈTRE

Le baromètre mesure la pression atmosphérique. En mesurant la différence de pression par rapport à une référence, il permet au régulateur de vol de connaître la composante verticale du quadcoptère. Ce capteur est utilisé afin de maintenir la plateforme à une hauteur définie (mode *Altitude Hold*). Pour les déplacements proches du sol, un capteur à ultrasons peut également être utilisé afin de maintenir une altitude fixe.

MAGNÉTOMÈTRE

Le magnétomètre mesure le champ magnétique. Il est utilisé dans des modes de navigation un peu plus avancés. Il permet par exemple de définir une direction de pilotage fixe quelle que soit la position du quadcoptère. L'avant de l'appareil n'est plus défini par rapport à la structure mais par rapport à une direction géographique. Cela facilite le pilotage puisque la rotation du quadricoptère sur l'axe vertical n'a plus d'influence. Le magnétomètre est également utilisé dans les modes de pilotages utilisant le GPS.

GPS

Le GPS n'est pas utilisé pour stabiliser l'appareil à proprement parler. Il est utilisé pour programmer des déplacements à l'avance (*Waypoint*) et pour revenir automatiquement au point de lancement (*Return to Home*). Le GPS permet également de maintenir l'appareil à un endroit précis (*Position Hold*). Certains modèles utilisent également un capteur à flux optique (même principe que sur les souris d'ordinateur optique) afin de garantir un positionnement fixe de l'appareil, mais il ne marche qu'à des hauteurs relativement faibles.

AUTRES CAPTEURS

En plus de ces capteurs servant directement à stabiliser ou à positionner le quadricoptère, il est possible d'ajouter un grand nombre de capteurs en fonction de l'utilisation. On retrouve notamment des capteurs de détection d'objet ou de collision, de mesure de vitesse de l'air, mais également des capteurs externes à l'appareil comme des caméras 3D pour le positionnement du drone dans l'espace.

MOTEURS ET CONTRÔLEURS DE VITESSE ÉLECTRONIQUES

Il est possible d'utiliser n'importe quel moteur de modélisme sur un quadricoptère. Ce sont généralement des moteurs *brushless* à rotors externes. Comme ces moteurs n'ont pas de sens de rotations prédéfinis, il suffit d'interchanger deux des trois phases pour inverser le sens de rotation. Les ESC sont également facilement disponibles, elles sont utilisées depuis longtemps dans les avions et hélicoptères radiocommandés.

Depuis l'explosion du multirotor, on trouve des moteurs spécifiquement destinés à ces drones. Ils se caractérisent par un nombre d'électroaimants plus élevé que sur les moteurs « traditionnels ». Cela résulte en une vitesse de rotation plus lente qui permet d'éviter une partie des vibrations néfastes au vol d'un quadricoptère mais aussi un meilleur couple en limitant la consommation électrique. On utilise également des hélices plus longues qui augmentent la portée du modèle. Certaines compagnies proposent également des contrôleurs de vitesse dédiés à ces moteurs « multi-rotors » ils sont développés pour assurer une très bonne réactivité aux multiples changements de vitesse. Il existe même des boîtiers contenant les quatre contrôleurs afin de faciliter les branchements des moteurs et de l'alimentation.

5. CONCEPTION ET ÉTUDE D'UN DEUXIÈME PROTOTYPE

5.1. CHOIX DU DEUXIÈME PROTOTYPE

Pour poursuivre ce projet, il est nécessaire de réaliser un deuxième prototype. La conception de celui-ci doit permettre d'apporter plusieurs améliorations par rapport au premier modèle en tenant compte des constations faites durant son étude. Il s'agit avant tout de suivre l'évolution de la technologie de stabilisation à la fois par l'utilisation de plusieurs capteurs précis et d'un logiciel plus performant.

CRITÈRES DE SÉLECTION DU DEUXIÈME PROTOTYPE

Le but de la suite du projet est de mener des recherches en vue d'un quadricoptère autonome et intelligent. Cela implique que le modèle de base ait une bonne capacité de vol, une stabilisation efficace étant primordiale afin de pouvoir automatiser le contrôle de l'appareil.

Le régulateur de vol devrait donc être équipé d'un système de stabilisation avancé, utilisant des capteurs précis et si possible digitaux afin de réduire le bruit de ceux-ci. De plus, pour pouvoir effectuer des modifications, le logiciel du contrôleur de vol devrait être accessible. Le premier prototype a été conçu dans un esprit de miniaturisation du quadricoptère. Toutefois, un modèle d'une envergure plus importante permettrait d'obtenir un meilleur équilibre en gardant une certaine maniabilité et une bonne dynamique de vol. Il serait également plus puissant et conviendrait à une utilisation à l'extérieur. Finalement, le projet étant réalisé dans le cadre d'un travail de bachelor, le coût de l'appareil ne devrait pas être trop important.

CHOIX DES COMPOSANTS

La solution la moins coûteuse consiste à concevoir et de construire un second prototype artisanal en achetant seulement les composants qu'il n'est pas possible de fabriquer. Cependant, plusieurs éléments notamment l'aérodynamique de la structure et l'implémentation des capteurs nécessitent une certaine expérience dans le domaine ou une étude approfondie. Ces paramètres rendent la conception d'un modèle de cette complexité fastidieux, chronophage voire plus coûteux qu'un modèle du commerce, sans garantir de meilleures performances. Le but n'étant pas de reproduire ce qui est déjà disponible sur le marché à des prix abordables, cette solution n'a pas été retenue.

Les modèles prêts à voler ne correspondent pas non plus aux critères établis. Ils offrent effectivement une aptitude de vol convenable pour notre application mais ne permettent ni d'accéder au code, ni d'effectuer des modifications sur le prototype.

La solution qui répond le mieux aux objectifs fixés consiste en l'achat d'un kit et d'une carte de contrôle open source. Les kits comprennent en général une structure, les ESC, les moteurs et les hélices. Ces composants ne sont pas livrés assemblés, ce qui diminue le prix par rapport à des modèles prêts à voler. Leurs compatibilités ainsi que les performances de vol de l'ensemble ont néanmoins été testées ce qui assure une base matériel fiable. Comme pour les structures, il existe des copies des modèles les plus populaires à un prix légèrement inférieur. Cette

alternative est cependant peu judicieuse puisque l'économie est réalisée en utilisant des matériaux moins résistants ou des moteurs moins performants.

Le kit retenu est le FlameWheel 450 produit par DJI. D'un prix abordable, il est constitué d'une structure robuste ayant fait ses preuves. Les ESC et les moteurs sont des composants de qualité, balancés et conçus pour les multirotors. La marque DJI possède une très bonne expérience dans le domaine et offre en outre la possibilité de racheter n'importe quel composant du kit en pièce détachée.

Le régulateur de vol choisi est la carte open source Multiwii Pro achetée chez HobbyKing. Multiwii est un logiciel originellement basé sur les capteurs de la manette Wii. Le logiciel s'est développé et inclut à présent une vingtaine de cartes de contrôle¹⁷ conçues spécialement pour ce projet open source. Il est basé sur la plateforme Arduino et ses microprocesseurs ATmega. MultiWii bénéficie également d'une large communauté de développement, d'un wiki et de plusieurs forums qui seront très utiles lors des réglages du prototype.

5.2. FLAMEWHEEL 450

La structure DJI a une envergure de 450 mm. Elle est constituée de quatre pieds en plastique injecté et de deux plaques centrales métalliques. La plaque inférieure fait également office de distribution de puissance pour réduire la longueur des fils d'alimentation. Les pieds sont fixés à l'aide de 24 vis ce qui assure une très bonne rigidité. La structure est relativement massive, elle pèse 630 g, mais est très résistante aux chocs. Le kit comprend aussi quatre ESC Opto 30A¹⁸, quatre moteurs DJI ainsi que deux sets d'hélices de 8" et 10".

Le principal avantage de ce kit est de réunir des composants dimensionnés pour fonctionner ensemble, balancés et avec des moteurs équilibrés. Cela assure une portance suffisante pour un bon contrôle du quadricoptère (même avec une petite charge telle qu'une caméra embarquée) et réduit au maximum les vibrations.



Figure 19 - Composants constituant le Kit DJI F450

¹⁷ Les cartes sont fabriquées par des entreprises indépendantes de MultiWii mais sont constituées de composants compatibles avec le code.

¹⁸ Ces ESC n'ont pas besoin d'être calibrée par l'utilisateur car elles sont pré-balancées d'usine

MONTAGE

Le montage de la structure est très aisé. Il suffit de visser les moteurs sur les bras, puis les quatre bras aux deux plaques centrales. Comme la structure est destinée à être utilisée selon le mode "X", les deux bras de couleur rouge ont été disposés à l'avant. Lors du montage du kit, la principale difficulté est de souder les câbles d'alimentations des ESC sur la plaque inférieure.



Figure 20 - Les câbles d'alimentations sont soudés sur la "PowerBoard"

Le quadricoptère est entièrement alimenté par la batterie LiPo 3 cellules de 2650 mAh, qui est reliée à un connecteur lui aussi soudé sur la plaque de distribution. Le branchement des moteurs s'effectue à l'aide de trois câbles, un pour chaque phase. Pour alimenter la carte de régulation et le récepteur, un SBEC séparé, relié à la batterie, a été utilisé. Le SBEC est un transformateur de tension qui convertit le 11.1 V de la batterie en 5 V.

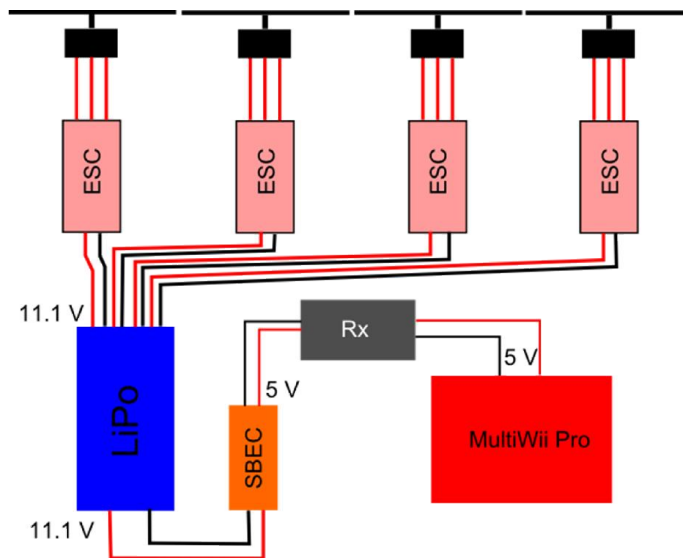


Figure 21 - Schéma de la distribution électrique du deuxième prototype

Les quatre canaux des signaux de commande du récepteur sont branchés sur les pins d'entrée de la MultiWii Pro. Les fils de signaux des ESC se branchent sur les pins de sortie et le GPS sur le port « Serial 2 ». Chaque entrée et sortie (aussi bien sur le récepteur que sur la carte) comportent trois pins. Deux pour l'alimentation (GND et +5V) et la troisième pour le signal en PWM. Il n'est pas nécessaire de connecter les trois pins à chaque fois car les pistes d'alimentation sont reliées entre elles. Une fois que le module est alimenté, seuls les câbles de signaux nécessitent d'être branchés.

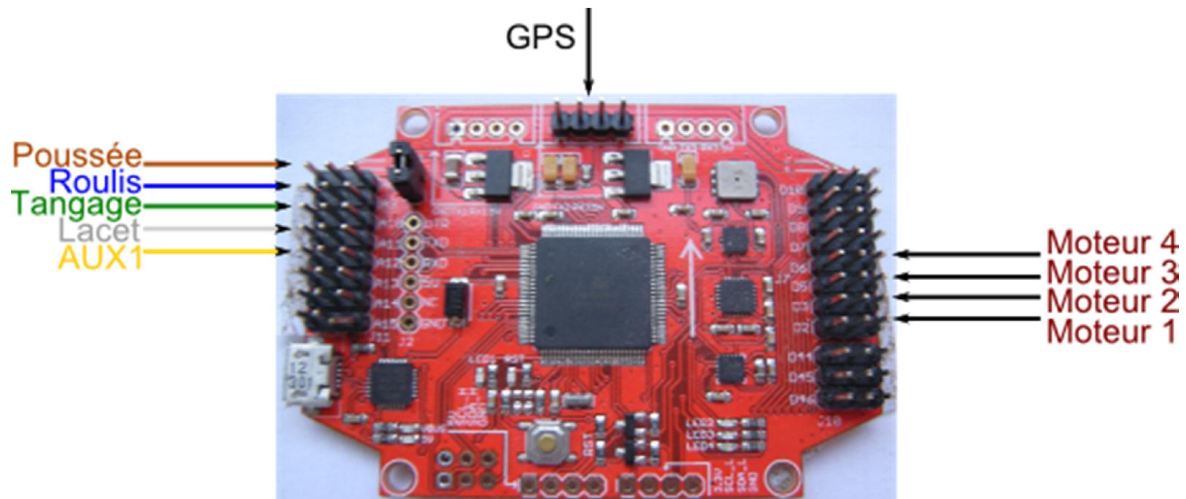


Figure 22 - Schéma de branchements des câbles de signaux sur la MultiWii Pro

5.3. MULTIWII PRO

La carte MultiWii Pro est un PCB mesurant 70x50x12 mm et pesant 16 g. Elle comprend le microprocesseur et les capteurs montés en surface. Elle inclut 8 entrées et 11 sorties digitales ainsi que trois « Serial port » et deux interface I²C.

Tableau 9 - Caractéristiques du microprocesseur ATmega 2560

<i>Fréquence maximale opérationnelle</i>	16 MHz
<i>Processeur</i>	8 bits
<i>Timer/counters</i>	6
<i>Convertisseur A/D 10-bits</i>	16
<i>Tension supportée</i>	4.5-5.5 V

Tableau 10 - Liste des capteurs présents sur la carte MultiWii Pro

Type de capteur	Mesure	Résolution	Plage de mesure	Utilisation
Gyromètre trois axes¹⁹	Vitesses angulaires	-	± 2000 [°/s]	Mesure les rotations de l'appareil
Accéléromètre trois axes²⁰	Accélérations	-	± 1 [G]	Détermine l'orientation de l'appareil
Baromètre²¹	Pression atmosphérique	0.25 [m]	-500...+9000 [m]	Permet le maintien à une hauteur fixe
Magnétomètre²²	Champ magnétique	1-2 [°]	360 [°]	Détermine l'orientation du champ magnétique terrestre
GPS²³	Position à l'aide de satellites	3 [m]	-	Détermine la position de l'appareil dans le plan

5.4. LOGICIEL

La dernière version²⁴ du logiciel open source MultiWii est un programme en C++. Il est compatible avec une trentaine de capteurs et cinq microprocesseurs différents. C'est pourquoi il doit être configuré afin de déterminer quels capteurs sont utilisés par la carte et comment gérer les différentes entrées/sorties. Si cette première initialisation est relativement simple à effectuer, le programme est en revanche relativement imposant et demande quelques connaissances en langage C pour d'être modifié. Le logiciel Arduino²⁵ permet d'ouvrir les fichiers du programme, de modifier et compiler le code et de le téléverser dans la carte à l'aide d'un câble USB.

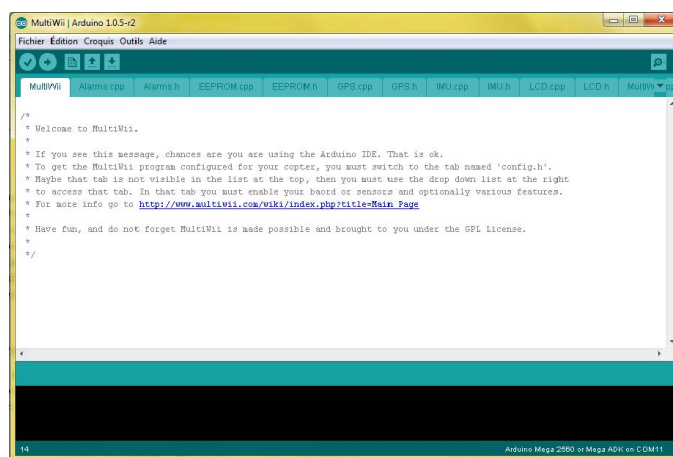


Figure 23 - Logiciel Arduino 1.0.5-r2

¹⁹ ITG3205 de InvenSense

<http://www.invensense.com/>

²⁰ BMA180 de Bosch

<http://www.bosch-sensortec.com/en/>

²¹ BMP085 de Bosch

<http://www.bosch-sensortec.com/en/>

²² HMC5883L de Honeywell

<http://honeywell.com/Pages/Home.aspx>

²³ MTK 3329 de Mediatek

<http://www.mediatek.com/en/>

²⁴ MultiWii 2.3 téléchargé sur <https://code.google.com/p/multiwii/downloads/list>

²⁵ Cf. Annexe 8.2 « Exporter le code sur la MultiWii Pro »

STRUCTURE ET PRINCIPAUX SOUS-PROGRAMMES

Le logiciel est composé de 11 fichiers **.cpp** et de leurs fichiers d'en-tête et de trois fichiers **.h** seuls. La boucle principale se trouve dans le fichier `MultiWii`. Les principaux fichiers modifiés lors de ce travail sont rapidement présentés ici.

MultiWii.cpp

C'est le fichier principal. Il comprend la boucle principale, une fonction d'initialisation qui appelle plusieurs fonctions d'autres fichiers pour configurer les interactions entre les différents composants, et les fonctions d'activation ou de désactivation des moteurs. La boucle principale exécute une série d'instructions de façon cyclique. Dans un premier temps elle récupère les données provenant du récepteur et les compare à des positions spécifiques des manches de commande impliquant une action particulière (par exemple activer les moteurs). Puis elle détermine le mode de vol qui a été sélectionné. Ensuite, elle appelle des fonctions relatives au magnétomètre, au baromètre et au GPS afin d'obtenir des informations sur la direction, la position ou la hauteur de l'appareil. Ces données, ainsi que celles des gyromètres et des accéléromètres sont envoyés et traités par le fichier `IMU.cpp` qui les fusionne pour obtenir le comportement du quadricoptère. Ces valeurs sont finalement récupérées par la boucle principale et viennent s'ajouter aux commandes de l'utilisateur avant d'être envoyées aux moteurs.

IMU.cpp

L'IMU (Inertial Measurement Unit ou centrale inertielle en français) est un système électronique constitué de gyromètres et d'accéléromètres permettant d'estimer l'orientation d'un objet. Le fichier `IMU.cpp` combine les renseignements des gyromètres et des accéléromètres pour en extraire des informations sur la position du modèle. Il utilise la méthode des filtres complémentaires, qui effectue un compromis entre la précision à court terme des gyromètres et celle à long terme des accéléromètres. Les informations du magnétomètre sont ensuite combinées afin d'obtenir une orientation du quadricoptère dans l'espace. Le fichier inclut aussi une fonction permettant d'estimer l'altitude de l'appareil à l'aide du baromètre. La gestion du GPS ainsi que les informations sur la position du quadricoptère sur le plan horizontal sont traitées dans un fichier séparé, `GPS.cpp`.

Output.cpp

Ce fichier s'occupe, comme son nom l'indique, de gérer les sorties digitales. Il génère des PWM transmis aux moteurs utilisés par le multirotor. Divers *timers* sont utilisés pour permettre des pulsations de durée indépendante pour chaque moteur. Les pins de sortie sont assignées à un registre de comparaison qui produit le PWM. La fonction *mixTable* reçoit, du fichier `MultiWii.cpp`, les modifications de positions souhaitées et applique les corrections à apporter à chaque moteur. La fonction *writeMotors* change le registre de comparaison de chacun des moteurs pour modifier la longueur de la pulsation du PWM. Les valeurs de ces pulsations sont comprises entre 1000 et 1950 μ s.

RX.cpp

Les informations provenant du récepteur sont traitées dans ce fichier. La longueur d'une impulsion reçue par une pin d'entrée est convertie en valeur utilisable par le programme. Pour ce faire, une routine d'interruption détecte les changements d'état du signal d'entrée. Le temps écoulé entre une montée et une descente correspond à la durée de l'impulsion du PWM. La fonction *computeRC* permet de transmettre les valeurs de toutes les pistes au fichier principal.

Sensors.cpp

Ce fichier regroupe les initialisations et les mesures des signaux des capteurs présents sur la carte. Les signaux provenant de chacun des capteurs sont convertis en valeurs numériques de même ordre de grandeur et stockées pour être fournies au programme lorsqu'il en a besoin. Les capteurs et le microprocesseur utilisent un bus de données I²C pour s'échanger des informations.

Config.h, Def.h et types.h

Ces trois fichiers sont des fichiers d'en-tête uniquement. Ils ne contiennent pas de code source, ils servent uniquement au compilateur en lui indiquant des actions à effectuer. Les deux premiers fichiers servent à définir des constantes de pré compilations. Le langage C++ permet de définir des constantes utilisées seulement par le pré-compilateur afin de compiler certaines parties du code uniquement lorsque cette constante est activée. Ces constantes sont précédées du signe « # ».

Config.h permet à l'utilisateur de configurer les différents composants qu'il désire utiliser²⁶ en activant ou désactivant des « *#define constante* ».

Def.h définit une série de constantes supplémentaires en fonctions des mots-clefs activés dans le fichier *config.h*. Typiquement, si l'utilisateur indique l'utilisation d'un quadricoptère dans le fichier de configuration, le fichier *def.h* définira un nombre de moteur égal à 4.

Le dernier fichier déclare certaines structures utilisées dans le programme. Ces structures sont créées afin de simplifier la visibilité de plusieurs variables en les regroupant sous un seul nom.

INITIALISATION

La première action à réaliser afin de pouvoir utiliser le logiciel MultiWii est de configurer le type d'appareil qu'il doit réguler et les composants à sa disposition. Dans le fichier *config.h*, il est possible d'activer des constantes prédéfinies. Ces constantes sont incluses dans le programme pour faciliter la tâche de l'utilisateur. Elles permettent notamment de choisir le type de multirobot employé, le type de carte ou les capteurs utilisés, les valeurs maximale et minimale à envoyer aux moteurs, d'appliquer un filtre aux gyromètres et de configurer le GPS.

Les actions réalisées pour l'utilisation de notre prototype sont détaillées dans l'annexe 8.1 « Configuration de la carte MultiWii Pro ».

²⁶ Cf. Section 8.1 « Configuration de la carte MultiWii Pro »

GUI

Pour faciliter l'utilisation du programme aux personnes peu habituées à la programmation informatique, le réglage des paramètres principaux peut être effectué par l'intermédiaire d'une fenêtre graphique. Ce programme permet d'interagir avec le quadricoptère par l'intermédiaire d'un câble USB. Une fois connectée²⁷, la fenêtre affiche les paramètres pouvant être modifiés ainsi que diverses informations fournies par la carte. Cette section explique les différentes zones de la fenêtre et tente d'expliquer leurs significations.

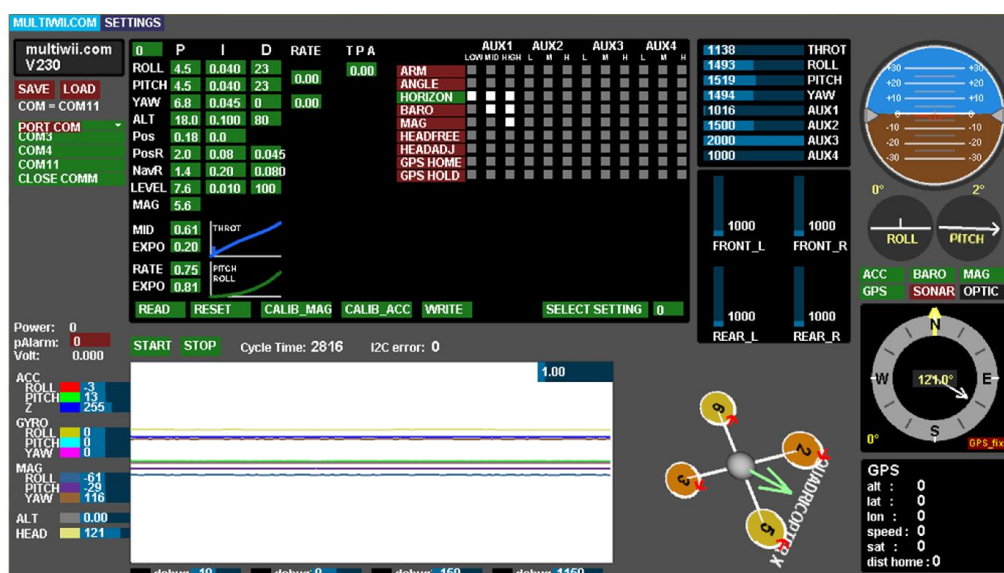


Figure 24 - Interface graphique MultiWii GUI 2.3

Une première zone de la fenêtre permet de visualiser et de changer le coefficient des PID de chaque composant. Les valeurs de chaque ligne sont appliquées à un des capteurs ou à un mode de vol particulier. L'ajustement des PID du prototype actuel est détaillé dans la section 5.6 « Mode de vol et PID ».

0	P	I	D
ROLL	4.5	0.040	23
PITCH	4.5	0.040	23
YAW	6.8	0.045	0
ALT	18.0	0.100	80
Pos	0.18	0.0	
PosR	2.0	0.08	0.045
NavR	1.4	0.20	0.080
LEVEL	7.6	0.010	100
MAG	5.6		

Figure 25 - Réglages des PID

²⁷ Cf. Annexe 8.3 « MultiWii GUI »

Tableau 11 - Capteurs sur lesquels les PID sont appliqués

Abréviation	Capteur concerné
Roll / Pitch / Yaw	Respectivement chaque axe du gyromètre
Alt	Baromètre pour le mode « AltHold »
Pos	GPS pour le mode « GPS Hold »
PosR / NavR	Respectivement la position du point de départ et le comportement du déplacement pour le mode « Return to Home »
LEVEL	Accéléromètre
MAG	Magnétomètre

En dessous se trouve une zone comprenant quatre paramètres. Ils permettent de modifier l'interprétation des valeurs reçues par la télécommande. Les deux premiers concernent la commande de la poussée. Il est parfois difficile d'obtenir la poussée qui maintient l'appareil en vol stationnaire. Le manche de poussée étant cranté, le passage d'un seul cran peut faire passer le modèle d'une légère montée à une légère descente et inversement. Si le pilote veut garder une altitude fixe, il doit donc continuellement modifier la commande des gazes. Les paramètres « MID » et « EXPO » définissent la pente de la commande des gazes autour d'une valeur définie. La différence de gaz entre deux crans peut donc être diminuée sur une certaine plage, permettant un réglage plus fin de la poussée. Les paramètres « RATE » et « EXPO » quant à eux les manches de direction de roulis et tangage. Ils permettent de donner plus ou moins d'importance au mouvement du manche et d'adoucir la zone neutre de ces deux rotations. Ce dernier paramètre est très utile dans notre cas car il permet de diminuer l'influence de la fluctuation des valeurs neutres.



Figure 26 - Paramètres de correction de la télécommande

Tableau 12 - Influence des paramètres sur la commande

Abréviation	Effet du paramètre
MID	Définit la valeur des gazes considérée comme la valeur qui permet un vol stationnaire.
EXPO	Permet de diminuer la pente des gazes afin d'avoir une plus grande précision aux alentours de la valeur définie dans MID.
RATE	Modifie la sensibilité des commandes de tangage et roulis.
EXPO	Définit la dimension de la zone adoucie autour du point milieu.

La zone à coté permet d'assigner les modes de vols en fonction de la valeur d'une des quatre entrées auxiliaires. Chaque AUX est divisé en trois états, lorsque l'utilisateur modifie l'état d'un des AUX, le capteur présélectionné est activé. Plusieurs capteurs peuvent être activés en même temps. Le mode de vol résultant dépend des capteurs activés²⁸. Dans notre cas, l'un des potentiomètres de la télécommande a été relié à l'entrée AUX1. Cela permet de sélectionner jusqu'à trois modes de vols différents et d'effectuer des changements de mode lorsque le quadricoptère est en l'air.



Figure 27 - Modes de vol enclenchés en fonction de la position du potentiomètre

En dessous, une zone présente les informations retournées par les capteurs. Les valeurs retournées par chacun des capteurs sont présentées numériquement et sous forme de graphe afin de voir son évolution. Cette section comprend également la durée de l'exécution de la boucle principale. Tout en bas, se trouvent quatre valeurs « debug » qui permettent de visualiser des paramètres du programme que l'utilisateur choisit. Une variable du code peut être assignée à ces visualisations grâce à l'instruction « debug[i] = variable ; »²⁹.

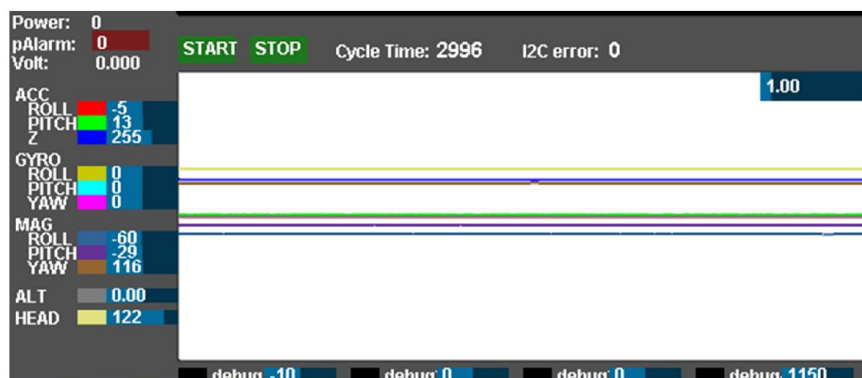


Figure 28 - Zone d'affichage des informations des capteurs

²⁸ Cf. Tableau 14 « Capteurs utilisés par les différents modes de vol »

²⁹ Avec « i » compris entre 0 et 3

Tout à droite se trouvent plusieurs informations provenant également de la carte. On trouve les valeurs des entrées (1), les valeurs de sortie des quatre moteurs (2), l'inclinaison des axes de roulis et de tangage (3), les capteurs détectés par le microprocesseur (4), l'orientation magnétique du prototype (5), les données GPS (6) et une visualisation 3D de l'orientation du modèle dans l'espace (7).

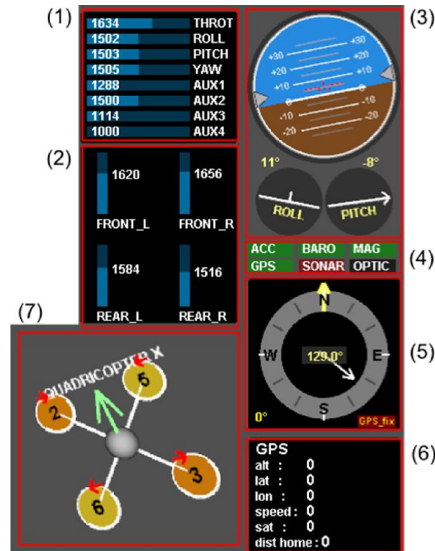


Figure 29 - Diverses informations sur l'état du quadricoptère

La visualisation des entrées permet en particulier de contrôler les valeurs reçues de la télécommande. Il est donc possible de calibrer celle-ci à l'aide des trims pour que les valeurs centrales des manches de directions correspondent aux valeurs attendues par le logiciel. Ces valeurs, identiques pour les trois pistes, sont fixées à 1500.

5.5. ACTIONS CONTRÔLÉES PAR LES MANCHES DE COMMANDES

Un certain nombre d'actions peuvent être effectuées avec la télécommande en plaçant les manches de direction dans des positions spécifiques. Le logiciel détecte ces positions en comparant les valeurs de la télécommande avec des valeurs prédéfinies correspondant aux positions extrêmes de la commande.

Tableau 13 - Position des manches de commandes pour réaliser une action

<i>Activer les moteurs</i>	
<i>Désactiver les moteurs</i>	
<i>Calibrer les gyromètres</i>	
<i>Calibrer l'accéléromètre</i>	
<i>Calibrer le magnétomètre</i>	
<i>Ajuster « l'horizon » de l'accéléromètre</i>	
<i>Calibration en vol</i>	

La plupart des radiocommandes ont des PWM qui varient entre 1000 et 2000 μ s. Ce n'est pas le cas de celle utilisée dans ce travail. Afin de pouvoir accéder aux actions contrôlées par les manches (*Command Stick Handler*) il a fallu changer les valeurs par défaut dans le code. Ces valeurs se trouvent dans le fichier MultiWii.h.

```

// #define MINCHECK 1100
// #define MAXCHECK 1900

#define MINCHECK 1200
#define MAXCHECK 1850

```

5.6. MODES DE VOLS ET PID

Le régulateur de vol MultiWii Pro permet de voler de façon plus ou moins assistée. L'utilisateur définit dans l'interface graphique les capteurs qu'il souhaite utiliser. Chaque capteur procure un certain type d'information qui peut être utilisé pour stabiliser, orienter ou positionner le modèle. Les différents modes de vols résultent des capteurs auquel l'utilisateur a décidé de recourir.

Tableau 14 - Capteurs utilisés par les différents modes de vol

Mode	Gyroscope	Accéléromètre	Baromètre	Magnétomètre	GPS
<i>Acro</i>	X				
<i>Horizon</i>	X	X			
<i>Mag</i>	X	X		X	
<i>Altitude Hold</i>	X	X	X		
<i>GPS Position Hold</i>	X	X		X	X

A chaque capteur sont associés des coefficients PID. Ils permettent de modifier la régulation des modes de vols. La valeur à attribuer aux coefficients dépend du type de vol désiré (réactif ou calme) mais aussi du poids, de l'envergure et de la puissance de l'appareil. Pour la plupart des capteurs, la composante P est la plus importante. Pour déterminer quelle est la valeur idéale, il faut l'augmenter progressivement en testant le comportement du quadricoptère à chaque fois. Si des oscillations apparaissent, le coefficient P est trop élevé. Ensuite le paramètre I est également augmenté jusqu'à ce que les dérives importantes aient disparu. Finalement la composante D détermine la réactivité de la correction. Une valeur élevée rend le modèle plus réactif alors qu'une valeur faible produit une correction en douceur. Lorsque plusieurs capteurs sont utilisés en même temps, il peut être difficile de déterminer quel facteur doit être corrigé. Il s'agit donc de régler ces paramètres successivement en ajoutant un capteur à la fois. Le gyromètre étant utilisé dans tous les modes de vol, c'est le premier capteur à étudier. Le tableau suivant expose les différents modes de vol testés, ainsi que les réglages des PID jugés adéquats pour ce projet.

Tableau 15 - Explications de chaque mode de vol et réglages des PID

Mode de vol	Description	Dénomination	P	I	D
Accro	N'utilise que les gyromètres, nécessite une bonne capacité de pilotage mais permet un vol très réactif	Roll & Pitch	4.5	0.040	23
		Yaw	6.8	0.045	0
Horizon	Maintien l'appareil à l'horizontale lors que les commandes de direction sont en position centrale	LEVEL	7.5	0.010	80
Mag	Maintien l'avant de l'appareil dans une direction fixe en l'absence de commande de lacet	MAG	4.0	-	-
GPS Hold	Garde une position GPS définie	Pos	0.20	0.1	0

Les modes de vol « Accro » et « Horizon » sont très stables et permettent un contrôle aisé de l'appareil. Les déplacements du modèle sont fluides et la dynamique de vol, bien que moins réactive que celle du premier prototype, permet des changements de directions décents. Le mode « Mag » a été combiné avec le mode Horizon afin de faciliter le pilotage en gardant une direction fixe. Pour un pilote inexpérimenté, il est difficile de reconnaître la position du quadricoptère en vol et de le diriger lorsque le modèle est pointé contre l'utilisateur. Le mode GPS maintient le quadricoptère dans un cercle d'environ 5 m de diamètre. Un réglage des PID plus important n'améliore pas la précision du positionnement, le modèle oscille de plus en plus jusqu'à s'écarter dramatiquement du point de départ.

5.7. CARACTÉRISATION DU DEUXIÈME PROTOTYPE

L'appareil a montré de bonnes capacités de vol. Il semble être une bonne plateforme pour le développement d'applications supplémentaires. Il est cependant nécessaire de caractériser certaines propriétés du modèle comme sa masse ou la durée de vol. En vue de l'ajout de capteurs supplémentaires ou d'une caméra par exemple, il est également utile de déterminer la poussée dont le modèle dispose.

POIDS

Chaque composant a été pesé afin de connaître son influence sur la masse totale du quadricoptère.

Tableau 16 - Détail de la masse de chaque composant

Nombre de Pièces	Composant	Poids total [g]
4	Hélices 8"	25
4	Hélices 10"	50
4	Moteurs	220
4	Bras	216
4	ESC	96
1	Plaque haute	28
1	Plaque basse	44
1	Multiwii pro	15
1	GPS	7
1	Récepteur	13
1	Bec	21
1	Batterie	212
4	Pied	12

On constate que près de 46 % de la masse totale du quadricoptère provient de trois composants. Les moteurs, les quatre bras et la batterie. Les moteurs et les bras du quadricoptère sont répartis symétriquement et ne posent donc pas de problème pour l'équilibre de l'appareil. Une batterie d'un tel poids est indispensable pour fournir suffisamment d'énergie au quadcoptère et pour permettre un vol d'une durée acceptable. Il faut cependant veiller à la placer au centre de la plateforme centrale afin de ne pas déséquilibrer l'appareil. Deux éléments sont positionnés de façon asymétrique. Il s'agit du récepteur et du BEC qui sont positionnés chacun à une extrémité de la plateforme inférieure. La différence de poids qui en résulte est cependant négligeable (de l'ordre de 5 g) par rapport à la masse totale du prototype.

Les câbles ainsi que divers connecteurs n'ont pas pu être pesés. Ils représentent toutefois une fraction négligeable du poids total. Lors du calcul de la poussée, une mesure de la force de pesantur du modèle complet a été réalisée.

POUSSÉE

Pour connaître la poussée que l'appareil est capable de délivrer ainsi que son évolution en fonction de la consigne, un petit banc de test a été développé. L'objectif est de mesurer le comportement de l'appareil dans son ensemble, en restant le plus proche possible d'une situation en vol, et de comparer les performances des différentes hélices.

La mesure est effectuée par un capteur de forces. Il permet de mesurer les forces et les couples auxquels il est soumis selon 3 axes. Dans un premier temps, nous n'utiliserons que l'axe vertical mais il est envisageable de caractériser le modèle plus précisément en utilisant les deux autres axes.

Tableau 17 - Caractéristiques du capteur de force ATI Mini40 SI-80-4

Plage de mesure³⁰		
F _x , F _y	± 240	[N]
F _z	± 240	[N]
T _x , T _y , T _z	± 4	[Nm]
Résolution³⁰		
F _x , F _y	0.005	[N]
F _z	0.01	[N]
T _x , T _y , T _z	0.125 * 10 ⁻³	[Nm]
Précision mesurée		
F _z	0.1 ± 0.05	[N]

Les valeurs physiques mesurées par le capteur sont transmises à un logiciel par l'intermédiaire d'un DAQ (*Data acquisition systems*). Ces valeurs peuvent alors être exportées. Le logiciel permet d'acquérir une série de données et de les enregistrer dans un fichier **.csv**. Il est ensuite possible d'effectuer une moyenne de ces valeurs afin d'améliorer la précision de la mesure. Le logiciel comprend également une fonction de tare qui ramène toutes les forces à zéro.



Figure 30 - Banc de mesure de la poussée

³⁰ Selon le Datasheet du fabricant

Un support spécialement conçu pour le prototype est assemblé sur le capteur. Il permet d'accueillir le quadricoptère et de le fixer par les quatre bras. Une tare du capteur est effectuée avant l'ajout du prototype, permettant de mesurer la masse du quadricoptère. Puis une consigne identique est envoyée à tous les moteurs après une nouvelle mise à zéro. En augmentant linéairement la consigne, il est possible de mesurer l'évolution de la poussée. Dans un premier temps, la consigne a été contrôlée directement avec le manche des gaz. En utilisant le logiciel, il a été possible de déterminer la valeur de commande qui produit une portance égale à la force de pesanteur du modèle. Cette poussée permet donc le maintien de l'appareil en vol stationnaire, on l'appellera poussée d'équilibre. En dessous de cette portance, l'étude du comportement des moteurs ne présente pas beaucoup d'intérêt, les mesures suivantes ont donc été effectuées à partir d'une valeur légèrement inférieure à la consigne d'équilibre. Deux séries de mesures ont ensuite été effectuées. Une pour chaque set d'hélice. Afin d'obtenir un contrôle précis de la consigne, les valeurs ont été introduites directement dans le programme de la carte de contrôle. Elles ont été augmentées par incrément de 50 jusqu'à une valeur de 1900, qui correspond à la limite supérieure de la télécommande.

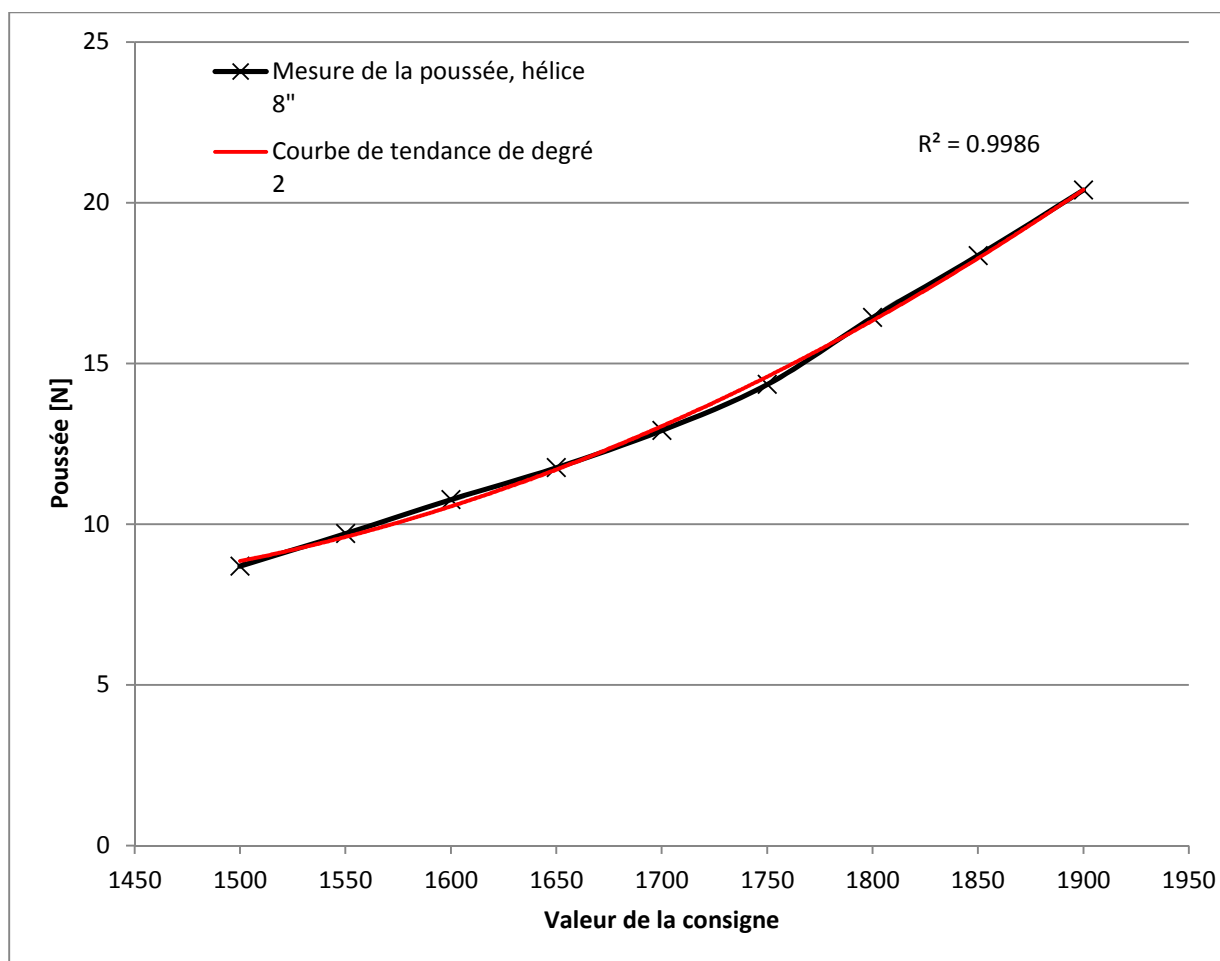


Figure 31 - Graphique de la poussée produite par les hélices 8"

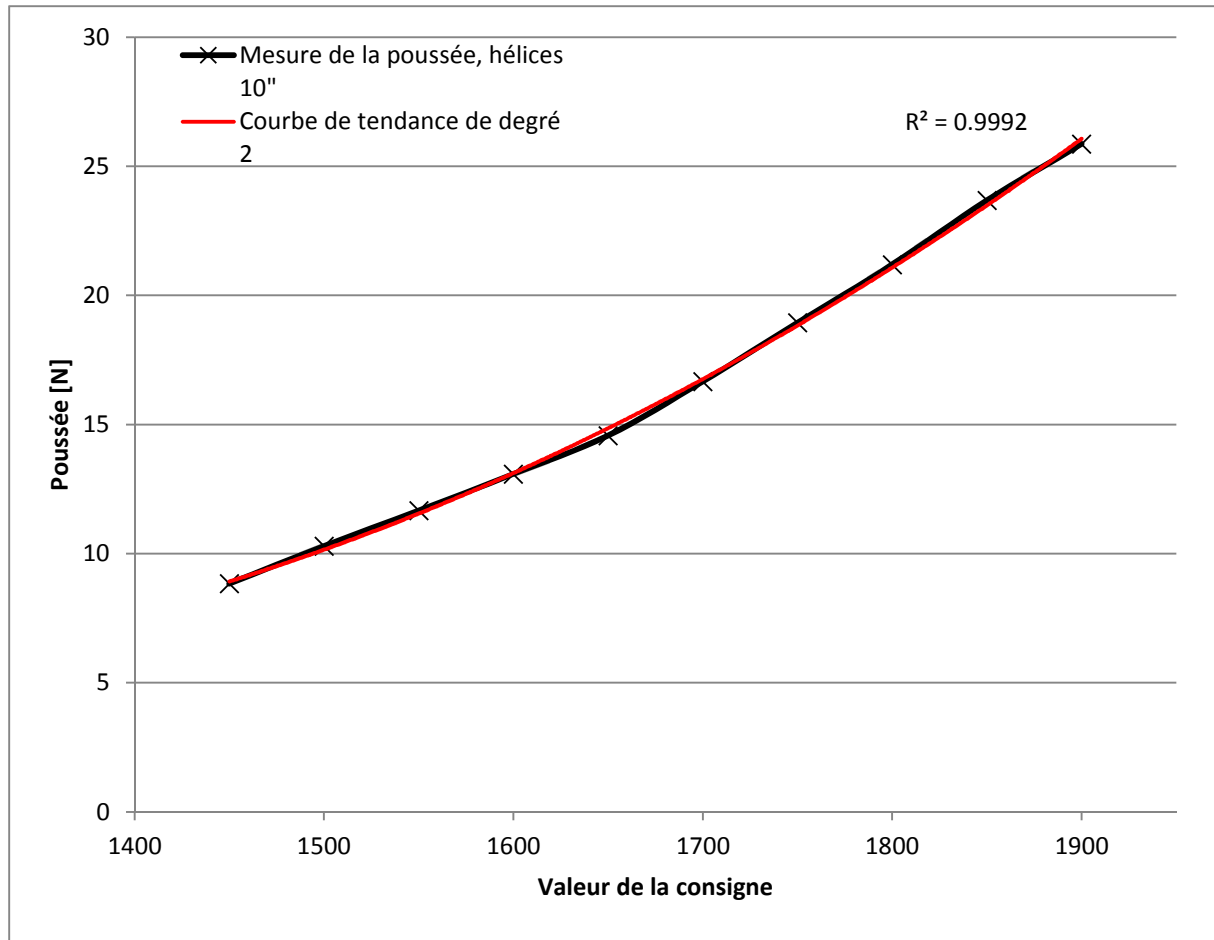


Figure 32 - Graphique de la poussée produite par les hélices 10"

Les graphes montrent que l'évolution de la poussée est quadratique par rapport à la consigne pour les deux types d'hélice. On obtient une poussée maximale de 20,4 N (2,08 Kg) avec les hélices 8" et de 25,9 N soit 2,64 Kg avec les hélices 10".

Les mesures de poussées ci-dessus ne tiennent pas compte de la masse du quadricoptère. Afin de déterminer la capacité de charge de ce dernier, il faut soustraire le poids du quadricoptère à la valeur de la poussée. D'après les mesures de la force de pesanteur de l'appareil, le quadricoptère pèse 956 g lorsqu'il est équipé des hélices 10" contre 931 g avec les hélices 8". Un autre facteur diminue la poussée utile du modèle. Il est important de tenir compte de la régulation du quadricoptère en vol. La stabilisation du modèle s'effectue en augmentant certains des moteurs. En faisant tourner tous les moteurs à plein régime, on supprime la capacité d'ajustement du régulateur. Il faut donc garder une marge de régulation pour obtenir un vol stable. La valeur maximale (gaz + corrections) que le programme transmet aux moteurs est de 1950. Augmenter cette valeur peut provoquer la surchauffe des ESC ou des moteurs. En admettant qu'une marge de 150 permet une correction suffisante, la portance maximale utile de ce quadricoptère peut être calculée.

$$10'' : Poussée_{1800} - Masse_{10} = Poussée_{utile\ 10}$$

$$2.16 - 0.956 = \mathbf{1.20\ Kg}$$

$$8'' : Poussée_{1800} - Masse_8 = Poussée_{utile\ 8}$$

$$1.67 - 0.931 = \mathbf{0.74\ Kg}$$

On constate que la poussée du quadricoptère avec les hélices 8" représente seulement 61 % de la portance à disposition avec les hélices 10". C'est ces dernières qui ont donc été privilégiées pour la suite du travail.

DURÉE DE VOL

La durée d'un vol dépend fortement de la nature du pilotage. Des accélérations répétées et des déplacements agressifs utilisent bien plus de puissance qu'un vol stationnaire. De même les perturbations extérieures telles que le vent peuvent diminuer le temps de vol. La capacité de la batterie permet un vol allant de 7 à 9 minutes environ. Un vol test a été effectué. Il s'agissait de se contenter de maintenir le modèle dans un cercle de 3 m de diamètre à une hauteur constante. Ces conditions ont été maintenues durant un peu plus de 10 minutes, ce qui peut être considérée comme la capacité de vol maximale du quadricoptère.



Figure 33 - Le deuxième prototype prêt à l'emploi

6. APPLICATION FINALE

L'application finale de ce travail de bachelor consiste à automatiser le décollage et l'atterrissage du quadricoptère. Le modèle doit être capable de décoller, se stabiliser à une hauteur définie puis d'atterrir sans intervention de l'utilisateur. Pour ce faire, le quadricoptère doit être équipé d'un capteur permettant de mesurer la hauteur du modèle de façon précise. Le baromètre inclus dans la carte de contrôle a une précision de l'ordre de 0.25 m, ce qui est largement insuffisant pour du contrôle à basse altitude. Un nouveau capteur a donc été ajouté au prototype.

6.1. NOUVEAU CAPTEUR

Le capteur sélectionné est un capteur de distance à ultrasons. Bien plus sensible que le capteur de pression atmosphérique, il offre une plage de mesure supérieure à un capteur infrarouge. Un capteur à ultrasons mesure la distance entre le capteur et un objet en émettant une onde sonore de fréquence élevée et en calculant le temps que le signal met à revenir au capteur après avoir été réfléchi par un obstacle. L'onde se propageant dans l'air à une vitesse constante (vitesse du son), il est possible de calculer la distance séparant les deux objets.

Le capteur choisi doit être contrôlé par un système électronique extérieur. Lorsque le capteur reçoit une impulsion de 10 μ s il émet huit impulsions d'ultrasons à 40 kHz, puis calcule le temps mis par les impulsions pour revenir. Il retourne alors un signal de la durée de l'aller-retour. Il est recommandé d'attendre au minimum 60 ms entre chaque mesure pour éviter la superposition de plusieurs signaux. Le capteur comprend quatre pins accessibles. Les pins « VCC » et « GND » sont les pins d'alimentation (5 V). La pin « Trig » permet d'envoyer l'impulsion de déclenchement et la pin « Echo » de lire la réponse du capteur.

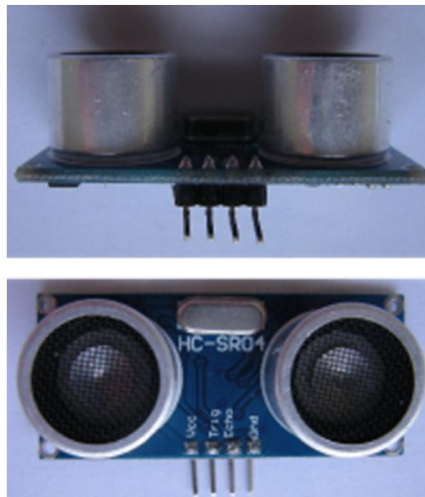


Figure 34 - Sonar HC-SR04

Le principal défaut du capteur à ultrasons est d'être sensible aux échos. Si une impulsion est réfléchi plusieurs fois (par exemple sur les murs d'une salle), le capteur risque d'interpréter le signal provenant de ces échos comme la réponse directe d'une impulsion. Il peut également être perturbé par des ondes émises par un autre appareil. Lors des premiers essais, le capteur a montré des signes d'instabilité en retournant parfois des valeurs totalement erronées sur une

courte période. Afin de s'affranchir de ces erreurs aléatoires, deux capteurs ont été montés sur le prototype. Ainsi une valeur est considérée comme correcte lorsque les deux capteurs retournent la même valeur.

IMPLÉMENTATION DU CAPTEUR SUR LA CARTE DE CONTRÔLE

Pour contrôler le déclenchement des mesures, des canaux de sorties de la carte MultiWii Pro encore disponibles ont été utilisés. Ils permettent de générer des PWM ayant une durée d'impulsion de 10 μ s et une période de 60 ms. Les sorties sélectionnées, « D45 » et « D46 », sont normalement utilisées pour le contrôle d'une caméra. Le *timer* 5 du microcontrôleur a été utilisé pour définir deux PWM identiques sur ces sorties.

Dans le fichier *Output.cpp*

```
void initOutput() {
    /***** mark all PWM pins as Output
    for(uint8_t i=0;i<NUMBER_MOTOR;i++) {
        pinMode(PWM_PIN[i],OUTPUT);

        pinMode(46,OUTPUT); // Assign pins 45 and 46 as Output
        pinMode(45,OUTPUT); // for Sonar
    }

    TCCR5A |= (1<<WGM51); // phase correct mode & prescaler to 8 = 1us resolution
    TCCR5A &= ~(1<<WGM50);
    TCCR5B &= ~(1<<WGM52) & ~(1<<CS50) & ~(1<<CS52);
    TCCR5B |= (1<<WGM53) | (1<<CS51);
    ICR5 = 0xEA60; //top to 60000 (60ms)

    TCCR5A |= (1<<COM5A1); // pin 46
    TCCR5A |= (1<<COM5B1); // pin 45
    OCR5B = 10;
    OCR5A = 10; // 10us pulse
```

Les valeurs retournées par le capteur sont transmises aux entrées « AUX3 » et « AUX4 » de la carte de contrôle. Ces canaux sont déjà programmés pour mesurer la durée d'une impulsion reçue. Seule une limitation sensée contrôler les valeurs provenant d'une télécommande a dû être supprimée. Les valeurs du temps de trajet sont accessibles dans le code à travers les variables « *rcData[AUX3/4]* ».

Dans le fichier RX.cpp

```

#define RX_PIN_CHECK_SONAR(pin_pos, rc_value_pos)
if (mask & PCInt_RX_Pins[pin_pos]) {
    if (!(pin & PCInt_RX_Pins[pin_pos])) {
        dTime = cTime-edgeTime[pin_pos];
        rcValue[rc_value_pos] = dTime;
    } else edgeTime[pin_pos] = cTime;
}

#if (PCINT_PIN_COUNT > 2)
    RX_PIN_CHECK(2,5);
#endif
#if (PCINT_PIN_COUNT > 3)
    RX_PIN_CHECK_SONAR(3,6);
#endif
#if (PCINT_PIN_COUNT > 4)
    RX_PIN_CHECK_SONAR(4,7);
#endif

```

CARACTÉRISATION

En théorie, la valeur retournée par le capteur dépend uniquement de la vitesse du son dans l'air. Le temps de l'aller-retour multiplié par la vitesse du son est égal à deux fois la distance entre le capteur et l'obstacle. Cependant la vitesse du son varie en fonction des conditions climatiques de l'environnement dans lequel évolue l'onde, en particulier la température. Les capteurs peuvent également présenter des défauts ou des erreurs d'offsets. Pour quantifier la valeur liant réellement la mesure du temps de réponse à la distance, un test de distance a été effectué.

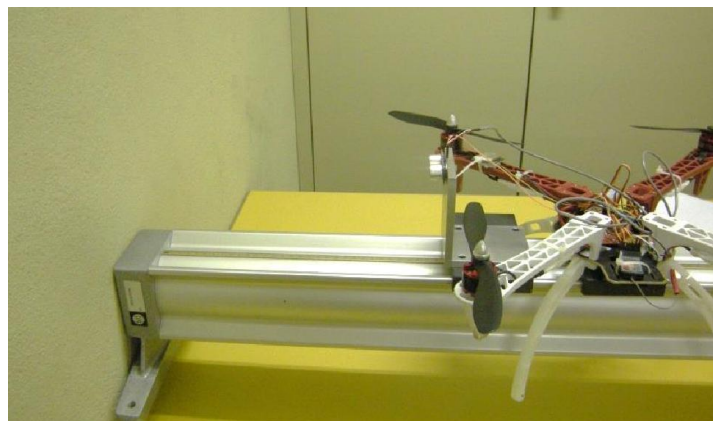


Figure 35 - Banc de mesure pour la réponse du Sonar

Le capteur a été placé sur un rail gradué et dirigé contre un mur. Des mesures du temps de réponse ont été réalisées pour plusieurs distances différentes. Les valeurs retournées par le capteur ont été affichées par l'intermédiaire de la fenêtre graphique MultiWii. Cela présente l'avantage d'afficher la valeur qui est réellement utilisée par le logiciel, après conversion digitale. En revanche les valeurs provenant du capteur présentent une fluctuation de plus en plus importante en fonction de la distance. Les valeurs présente dans le tableau # ont été sélectionnées lors d'une stabilisation temporaire de l'oscillation. De plus, pour chaque distance, 5 valeurs ont été moyennées. Les valeurs proviennent de deux « allers » et un « retour » du

premier capteur et d'un « aller-retour » du deuxième. Par « aller » on entend une série de mesures effectuées en augmentant la distance au mur. Le « retour » signifie le rapprochement du capteur contre le mur entre chaque mesure. Les distances varient entre 50 et 1000 mm. Au-delà, des phénomènes d'écho, dus à l'espace confiné de la salle d'essai, ont perturbé les mesures. Lors des tests en plein air, les capteurs ont permis de maintenir une altitude constante jusqu'à 2 m. Selon le fabricant, le capteur permet de mesurer des distances allant de 2 à 400 cm cependant au-dessus de 2 m le comportement du quadricoptère devient instable.

Tableau 18 - Donnée des 5 mesures de distance des deux sonars

Distance [mm]	Mesure					Moyenne [μ s]
	1	2	3	4	5	
50	320	320	314	315	291	312
55	347	349	342	319	318	335
60	376	370	346	346	375	362.6
70	429	426	428	425	428	427.2
80	514	510	482	483	481	494
100	602	596	595	598	598	597.8
200	1201	1200	1200	1198	1197	1199.2
300	1778	1798	1798	1750	1774	1779.6
500	2934	2959	2953	2958	2950	2950.8
750	4206	4180	4182	4200	5202	4394
1000	5638	5662	5640	5630	5629	5639.8

On constate de petites variations entre les différentes mesures de la même distance. Pour les faibles distances, le deuxième capteur retourne une valeur légèrement inférieure à celle du premier. Ces différences rendent des mesures de distance précises impossible. Cependant l'application pour laquelle les capteurs sont utilisés ne demande pas une précision excédant 10 [mm]. Dès lors, une variation de quelques dizaines de millisecondes est inférieure à la précision souhaitée.

Une courbe de tendance linéaire appliquée à la moyenne des valeurs permet de trouver un facteur de conversion exprimant au mieux la distance en fonction du temps de réponse. Ce facteur est de 5.76 [ms/mm]. Dans le code, la valeur reçue par le capteur est divisée par 58, afin de représenter la hauteur en centimètres.

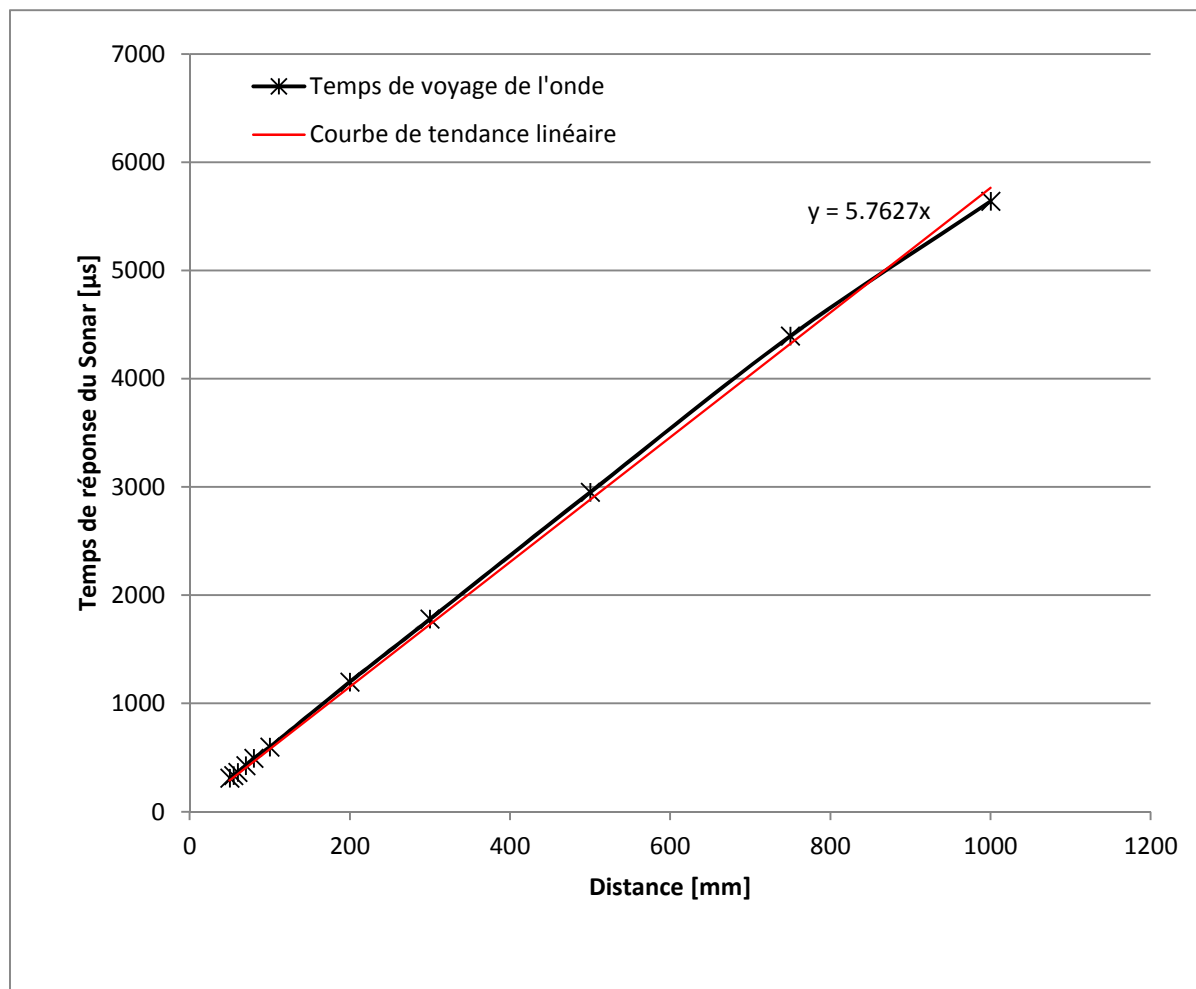


Figure 36 - Graphique de la réponse du Sonar en fonction de la distance

6.2. DÉCOLLAGE ET ATERRISSAGE AUTOMATISÉS

Avant d'implémenter le décollage et l'atterrissage automatique du quadricoptère, les capteurs à ultrasons ont tout d'abord été utilisés afin de stabiliser le modèle à basse altitude. Le logiciel MultiWii prévoit déjà une régulation de l'altitude en utilisant un baromètre. Les variables utilisées pour cette régulation ont été réutilisées pour les sonars.

ALTHOLD

Les valeurs provenant des deux sonars sont tout d'abord divisées par le facteur trouvé lors du test de distance et récupérées dans deux variables « *BaroAlt* » et « *BaroAlt2* ». Elles sont ensuite comparées pour détecter une éventuelle erreur d'un des capteurs. Si les deux valeurs sont proches, l'altitude utilisée est la moyenne des deux capteurs. Lorsque les valeurs sont trop différentes, le capteur retournant la valeur la plus proche de la distance précédente est alors considéré comme exact. Une moyenne flottante est ensuite appliquée afin de filtrer les éventuelles aberrations résiduelles. La hauteur (en centimètre) est alors utilisée pour réguler la position verticale du prototype.

L'accès à cette régulation de hauteur est obtenu par le mode de vol dit « AltHold » et peut être activé par le potentiomètre AUX de la télécommande. Pour obtenir de bons résultats, ce mode de

vol doit être combiné au mode « Horizon » qui maintien l'appareil à l'horizontale. Pratiquement le pilote amène le quadricoptère à la hauteur de maintien désirée et active le mode « AltHold ». Le quadricoptère se maintient à cette altitude et l'utilisateur peut alors se contenter de contrôler les déplacements du modèle dans le plan horizontal.

Comme les autres modes de vols, le mode « AltHold » est réglé par un correcteur PID. Les coefficients peuvent être modifiés à l'aide de l'interface graphique. Cependant l'ordre de grandeur de ces valeurs est bien plus élevé pour l'utilisation des sonars que du baromètre. Les coefficients *proportionnel* et *dérivé* ont donc été multipliés par quatre dans le code et le coefficient *intégral* par deux.

Dans le fichier IMU.cpp

```
BaroAlt = rcData[AUX4]/58;
BaroAlt2 = rcData[AUX3]/58;
//debug[0] = BaroAlt2;
if ({BaroAlt < 5 || BaroAlt > 300} && {BaroAlt2 < 5 || BaroAlt2 > 300}) {
    BaroAlt = lastBaroAlt;
}

if (abs(BaroAlt - BaroAlt2) < 2)
    BaroAlt = (BaroAlt + BaroAlt2)/2;
else if ({abs(BaroAlt - lastBaroAlt)} > {abs(BaroAlt2 - lastBaroAlt)})
    BaroAlt = BaroAlt2;

//BaroPID = constrain((conf.pid[PIDALT].P8 * error16 >>7), -150, +150);
BaroPID = constrain((conf.pid[PIDALT].P8 * error16 >>5), -150, +150);

//errorAltitudeI += conf.pid[PIDALT].I8 * error16 >>6;
errorAltitudeI += conf.pid[PIDALT].I8 * error16 >>5;

//BaroPID -= constrain(conf.pid[PIDALT].D8 * alt.vario >>6, -150, 150);
BaroPID -= constrain(conf.pid[PIDALT].D8 * alt.vario >>4, -150, 150);
```

Les valeurs du PID ont été ajustées pour atténuer au maximum les oscillations du quadricoptère. Les coefficients présentés ci-dessous permettent de garder le modèle dans une zone inférieure à 10 cm de hauteur.

ALT	18.0	0.100	80
-----	------	-------	----

DÉCOLLAGE ET ATTERRISSAGE

Le décollage et l'atterrissage automatisés sont réalisés par l'exécution d'une série d'instructions dépendant de la position du quadricoptère. Lorsque l'appareil a atteint une position désirée, le programme exécute l'instruction suivante à l'aide d'une « variable d'état » *Sonarflag*. Après la montée, le quadricoptère se stabilise à hauteur constante en attendant une consigne du pilote pour amorcer la descente. La séquence peut être résumée de la manière suivante³¹.

1. Le modèle est au sol, les moteurs sont activés mais ne tournent pas.
2. Augmentation de la poussée jusqu'au décollage de l'appareil.
3. Poussée maintenue jusqu'à ce que le quadricoptère ait atteint une hauteur prédéfinie.
4. Stabilisation de l'appareil à cette hauteur.
5. Légère réduction de la poussée pour que le modèle amorce la descente.
6. Descente jusqu'à ce que le quadricoptère touche le sol.
7. Arrêt des moteurs.

Le pilote se contente de déclencher le décollage du quadricoptère, en activant le mode de vol « AltHold » et en plaçant le manche de poussée à mi-course. Il provoque ensuite l'atterrissage en ramenant le manche de commande en position basse. Durant le vol, notamment durant la stabilisation à hauteur constante, il contrôle uniquement les déplacements de l'appareil dans le plan.

Dans le fichier *IMU.cpp*

```
if ((BaroAlt < 3) && (Sonarflag < 2) ) {  
    if (rcData[THROTTLE] > 1300){  
        Sonarflag = 1;  
    }  
    else rcCommand[THROTTLE] = MINTHROTTLE;  
}  
else if ((BaroAlt < 180) && (Sonarflag < 3) && (Sonarflag > 0)) {  
    Sonarflag = 2;  
    AltHold = alt.EstAlt;  
}  
else if (Sonarflag == 2) {  
    Sonarflag = 3;  
    AltHold = alt.EstAlt;  
    BaroPID = 0;  
}  
else if ((Sonarflag == 3) && (rcData[THROTTLE] < 1300)) {  
    Sonarflag = 4;  
}  
  
else if ((Sonarflag == 4) && (alt.EstAlt <= int(logGroundAlt + 1))) {  
    Sonarflag = 5;  
}
```

1.1. ³¹ Cf. Annexe 8.4 « Marche à suivre pour le décollage et l'atterrissage automatisé »

Dans le fichier *MultiWii.cpp*

```
#if BARO
  #if (!defined(SUPPRESS_BARO_ALTHOLD))
    if (rcOptions[BOXBARO]) {
      if (!f.BARO_MODE) {
        f.BARO_MODE = 1;
        AltHold = alt.EstAlt;
        #if defined(ALT_HOLD_THROTTLE_MIDPOINT)
          initialThrottleHold = ALT_HOLD_THROTTLE_MIDPOINT;
        #else
          //initialThrottleHold = rcCommand[THROTTLE];
          initThrottle = MIN_THROTTLE;
        #endif
        errorAltitudeI = 0;
        BaroPID=0;
        Sonarflag = 0;
      }
    } else {
      f.BARO_MODE = 0;
    }
  #endif

  #if BARO && (!defined(SUPPRESS_BARO_ALTHOLD))
    if (f.BARO_MODE) {

      if (Sonarflag == 1) {
        static int16_t AltCorr = 0;
        if (AltCorr > 3) {
          initThrottle += 1;
          AltCorr = 0;
        }
        else AltCorr++;

        rcCommand[THROTTLE] = initThrottle;
      }
      else if (Sonarflag == 2) {
        rcCommand[THROTTLE] = initThrottle;
        initialThrottleHold = initThrottle - 5;
      }
      else if (Sonarflag == 3) {
        rcCommand[THROTTLE] = initialThrottleHold + BaroPID;
        //debug[2] = rcCommand[THROTTLE];
      }
      else if (Sonarflag == 4) {
        if (alt.EstAlt > 10)
          rcCommand[THROTTLE] = initialThrottleHold - 10;
        else rcCommand[THROTTLE] = initialThrottleHold - 20;
      }
      else if (Sonarflag == 5) {
        rcCommand[THROTTLE] = MIN_THROTTLE;
      }
    }
  #endif
}
```

Les résultats sont relativement concluants. Il est en effet possible d'effectuer un vol à hauteur constante sans modifier la commande de poussée. Le quadricoptère est également capable d'adapter sa hauteur lorsqu'il évolue dans une pente. La montée est effectuée assez rapidement, mais le quadricoptère est légèrement déséquilibré lorsqu'il quitte le sol, nécessitant un ajustement manuel de la position du modèle. Le principal défaut de ce code est d'avoir des valeurs de poussée définies dans le logiciel, en particulier la vitesse de maintien et de descente. Le code ne peut donc pas s'adapter entre les différentes positions de référence.

7. CONCLUSION

Ce projet a permis de mettre en évidence l'importance du choix des différents constituants d'un quadricoptère. De par sa conception, un multirotor est très facilement déstabilisé. Que ce soit par un déséquilibre des composants, une mauvaise gestion de la régulation ou des vibrations faussant la lecture des capteurs. La réalisation d'un tel modèle demande dès lors une bonne connaissance à la fois en mécanique mais également dans l'électronique et la régulation des capteurs.

La diversité des quadricoptères présents sur le marché montre effectivement un développement rapide de ce secteur. Que ce soit dans le développement des appareils ou de leur utilisation à des fins lucratives. Ce projet a également permis de constater l'amélioration significative des divers composants d'un quadricoptère au cours de ces dernières années

Le développement du décollage et de l'atterrissage automatisé a permis une première approche d'un drone « intelligent ». Le code de contrôle est cependant très limité et ne fonctionne que pour une configuration très précise. L'ajustement en continu de la consigne des gaz ainsi qu'une gestion de la vitesse de montée et de descente devrait permettre un comportement mieux contrôlé. La fusion entre les informations du sonar et du baromètre permettrait de stabiliser le modèle à des hauteurs plus importantes. Un capteur infrarouge pourrait également permettre un atterrissage en douceur en mesurant les distances proche du sol de façon plus précise.

La plateforme a pu être testée et montre une stabilisation suffisante au développement d'applications supplémentaire. Sa poussée utile permet largement de supporter l'ajout d'autre capteur. Notamment pour la reconnaissance de position dans le plan ou pour l'évitement d'obstacles. Une caméra de quelques centaines de grammes pourrait également être ajoutée afin de permettre des missions d'observations.

8. ANNEXES

8.1. CONFIGURATION DE LA CARTE MULTIWII PRO

Avant de pouvoir utiliser la carte de contrôle, une série de configurations sont nécessaires pour initialiser correctement la carte.

- Ouvrir le fichier Multiwii.ino à l'aide du logiciel Arduino
- Ouvrir l'onglet « Config.h »
- Décommenter les lignes suivantes :
 - #define QUADX
 - #define FFIMUv2
 - #define MAG_DECLINATION 1.32³²
- Vérifier le code (compilation)
- Contrôler le type de carte : Outils > Type de carte > Arduino Mega 2560 or Mega ADK
- Sélectionner le port COM correspondant : Outils > Port série > COM #
- Uploader la carte par l'intermédiaire du câble USB

Charger un nouveau code sur la carte

Si un changement majeur de code est apportée, vider la mémoire EEPROM :

- Ouvrir le logiciel Arduino
- Ouvrir le programme eeprom_clear : Fichier > Exemples > EEPROM > eeprom_clear
- Uploader le programme eeprom_clear sur la carte
- Uploader le nouveau code MultiWii

Configuration pour le GPS

- Effacer la mémoire EEPROM
- Décommenter les paramètres suivants dans « Config.h » en plus des paramètres standards :
 - #define GPS_SERIAL 2
 - #define INIT_MTK_GPS
- Vérifier et charger le programme
- Déconnecter puis reconnecter la carte
- Commenter la ligne :
 - #define INIT_MTK_GPS
- Décommenter la ligne :
 - #define NMEA
- Charger à nouveau le code

³² <http://magnetic-declination.com/> pour le lieu de vol. Changer les minutes secondes en fraction de minutes (Exemple : 1°19' = 1.32°)

8.2. EXPORTER LE CODE SUR LA MULTIWII PRO

Les modifications du logiciel MultiWii 2.3 et le chargement sur la carte se fait par l'intermédiaire de l'interface Arduino.

- Télécharger et installer Arduino³³ (version actuelle 1.0.5)
- Télécharger le logiciel MultiWii³⁴ (version actuelle 2.3)
- Ouvrir le programme MultiWii 2.3 à l'aide du fichier MultiWii.ino
 - « Répertoire d'installation »\MultiWii_2_3\MultiWii
- Configurer l'interface pour le microprocesseur et le port COM utilisés
 - Outils > Type de carte > Arduino Mega 2560 or Mega ADK
 - Outils > Port série > COM #
- Modifier le code si nécessaire
- Compiler le code
 - 
- Le téléverser sur la carte
 - 

8.3. ANNEXE # MULTIWII GUI

L'interface graphique MultiWii permet d'accéder aux réglages importants et de configurer les modes de vol.

- Démarrer l'application MultiWiiConf
 - « Répertoire d'installation »\MultiWii_2_3\MultiWiiConf\application.windows32
- Connecter le câble USB sur la carte et à l'ordinateur
- Sélectionner le port COM utilisé par la carte
 - 
- Appuyer sur **START**
- Changer les valeurs désirées
- Appuyer sur **WRITE** pour transférer les modifications
- Lors de la première utilisation ou après avoir effacé la mémoire EEPROM il peut être nécessaire de calibrer les accéléromètres (**CALIB ACC**) avant que les données des capteurs ne s'affichent.

³³ <http://arduino.cc/en/Main/Software>

³⁴ <https://code.google.com/p/multiwii/downloads/list>

8.4. MARCHE À SUIVRE POUR LE DÉCOLLAGE ET L'ATTERRISSAGE AUTOMATISÉ

- Charger le code adéquat sur la carte
- Paramétrer les modes de vol pour avoir accès au mode « Horizon » seul et aux modes « Horizon » et « Baro » combinés depuis la télécommande.
- Brancher la batterie
- Basculer en mode « Horizon » seul
- Activer les moteurs
- Basculer en mode « Horizon » et « Baro » combiné
- Pour décoller monter les gazes jusqu'à la moitié
- Contrôler la position dans le plan avec le manche de tangage/roulis
- Une fois en vol stationnaire, il est possible de déplacer le modèle à l'aide de ce manche
- Pour amorcer la descente, ramener la commande des gaz à zéro
- Une fois l'appareil posé, il est possible de redécoller en remontant le manche des gazes ou revenir en mode « Horizon » seul pour désactiver les moteurs

8.5. CHARGER LES BATTERIES

Les batteries LiPo doivent être manipulées avec soin. Un court-circuit ou une mauvaise utilisation du chargeur peuvent provoquer la combustion de la batterie. La batterie du premier prototype doit être chargée en mode normale, courant 0.8 A et tension 7.4 V. La deuxième peut être chargée en mode balance si le connecteur 4 fils est branché, courant 1.5 A (limité par l'alimentation de 2 A) et tension 11.1 V.

Il est préférable de charger la batterie avec la limite de courant inférieure ou égale à sa capacité³⁵ et au courant max de l'alimentation. Une batterie 2650 mAh doit être chargée de préférence à 2.65 [A] ou moins.

Chargement d'une batterie en mode normal

- Brancher l'alimentation 12 [V] au chargeur **G.T. Power A8**
- Brancher la batterie aux câbles d'alimentation du chargeur
- Sélectionner « LiPo BATT » à l'aide du bouton **Batt Type** puis appuyer sur **Enter**
- Sélectionner « LiPo CHARGE » à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Modifier le courant à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Modifier la tension³⁶ à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Appuyer longuement sur enter, contrôler les paramètres, appuyer à nouveau sur **Enter**
- Lorsque le chargement est terminé, appuyer sur **Stop**

Chargement d'une batterie en mode Balance (batterie 3S et plus)

- Brancher l'alimentation 12 [V] au chargeur **G.T. Power A8**
- Brancher la batterie aux câbles d'alimentation du chargeur
- Brancher le connecteur blanc de la batterie dans la prise de chargement balancé du chargeur
- Sélectionner « LiPo BATT » à l'aide du bouton **Batt Type** puis appuyer sur **Enter**
- Sélectionner « LiPo BALANCE » à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Modifier le courant à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Modifier la tension à l'aide des boutons **Dec** et **Inc**, appuyer sur **Enter**
- Appuyer longuement sur enter, contrôler les paramètres, appuyer à nouveau sur **Enter**
- Lorsque le chargement est terminé, appuyer sur **Stop**

³⁵ La capacité d'une batterie est le nombre d'ampères-heures disponible (usuellement noté en [mAh]).

³⁶ La tension de charge dépend du nombre de cellules de la batterie. Chaque cellule produit une tension de 3.7 V. Une batterie 2S doit être chargée à 7.4 V.

8.6. DATASHEET CAPTEUR DE FORCES ATI MINI40 SI-80-4

MINI40

Product Advantages

One of the Smallest 6-axis Sensors in the World:
The Mini40 has a compact, low-profile design.

Extremely High Strength:

- EDM wire-cut from high yield-strength stainless steel.
- Maximum allowable overload values are 4.3 to 18 times rated capacities.

High Signal-to-Noise Ratio: Silicon strain gauges provide a signal 75 times stronger than conventional foil gauges. This signal is amplified, resulting in near-zero noise distortion.

Typical Applications

- Telerobotics
- Robotic surgery
- Robotic hand research
- Finger-force research

The Mini40 F/T transducer
The transducer is made of hardened stainless steel with integral interface plates made from high-strength aircraft aluminum.

SENSING RANGES		Calibrations					
Axes		US-5-10		US-10-20		US-20-40	
ENGLISH CALIBRATIONS	Fx, Fy (±lbf)	5		10		20	
	Fz (±lbf)	15		30		60	
	Tx, Ty (±lbf-in)	10		20		40	
	Tz (±lbf-in)	10		20		40	
RESOLUTION		System Type*					
Axes		CON		DAQ		CON	
		DAQ		CON		DAQ	

SENSING RANGES		Calibrations					
Axes		SI-20-1		SI-40-2		SI-80-4	
METRIC CALIBRATIONS	Fx, Fy (±N)	20		40		80	
	Fz (±N)	60		120		240	
	Tx, Ty (±Nm)	1		2		4	
	Tz (±Nm)	1		2		4	
RESOLUTION		System Type*					
Axes		CON		DAQ		CON	
		DAQ		CON		DAQ	

Contact ATI for complex loading information. Resolutions are typical. All Sensors calibrated by ATI. *CON: Controller F/T System, DAQ: 16-bit DAQ F/T System

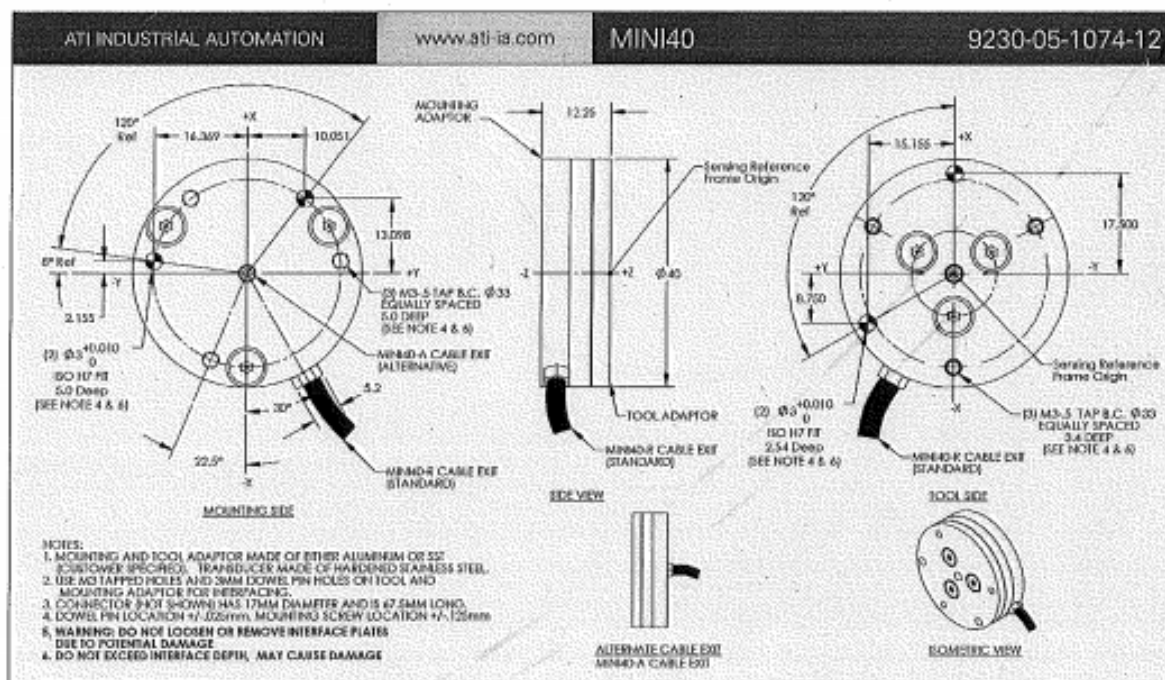
14 VISIT WWW.ATI-IA.COM FOR CURRENT PRODUCT SPECIFICATIONS, 2-D DRAWINGS, AND 3-D CAD MODELS

Single-Axis Overload	English	Metric
F _{xy}	±200 lbf	±870 N
F _z	±610 lbf	±2700 N
T _{xy}	±190 lbf-in	±22 Nm
T _z	±190 lbf-in	±21 Nm
Stiffness (Calculated)	English	Metric
X-axis & Y-axis force (K _x , K _y)	65x10 ³ lb/in	11x10 ⁶ N/m
Z-axis force (K _z)	130x10 ³ lb/in	23x10 ⁶ N/m
X-axis & Y-axis torque (K _{tx} , K _{ty})	29x10 ³ lbf-in/rad	3.3x10 ³ Nm/rad
Z-axis torque (K _{tz})	38x10 ³ lbf-in/rad	4.3x10 ³ Nm/rad
Resonant Frequency (Measured)		
F _x , F _y , T _z	3200 Hz	
F _z , T _x , T _y	4900 Hz	
Physical Specifications	English	Metric
Weight*	0.11 lb	50 g
Diameter*	1.57 in	40 mm
Height*	0.482 in	12.2 mm

*Specifications include standard interface plates.

"ATI's force transducers have given us high accuracy and easy to use tools that integrate perfectly with our research needs."

Dr. Douglas Weeks
Dept. of Physical Therapy
Regis University



8.7. DATASHEET SONAR HC-SR04



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

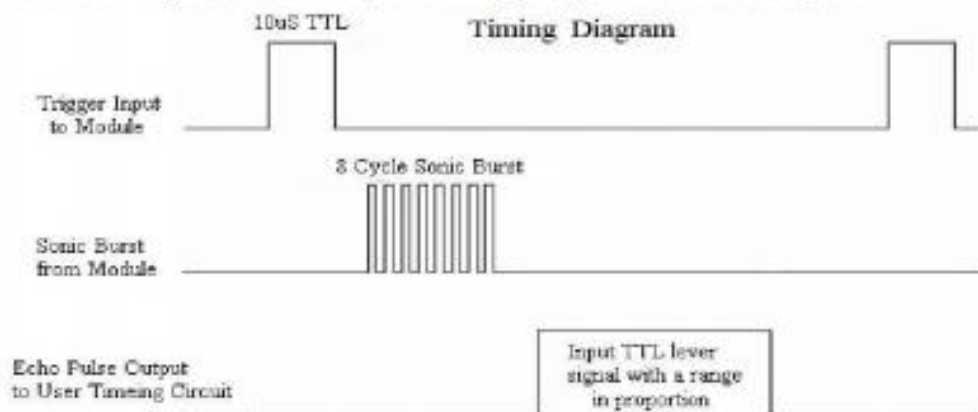
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

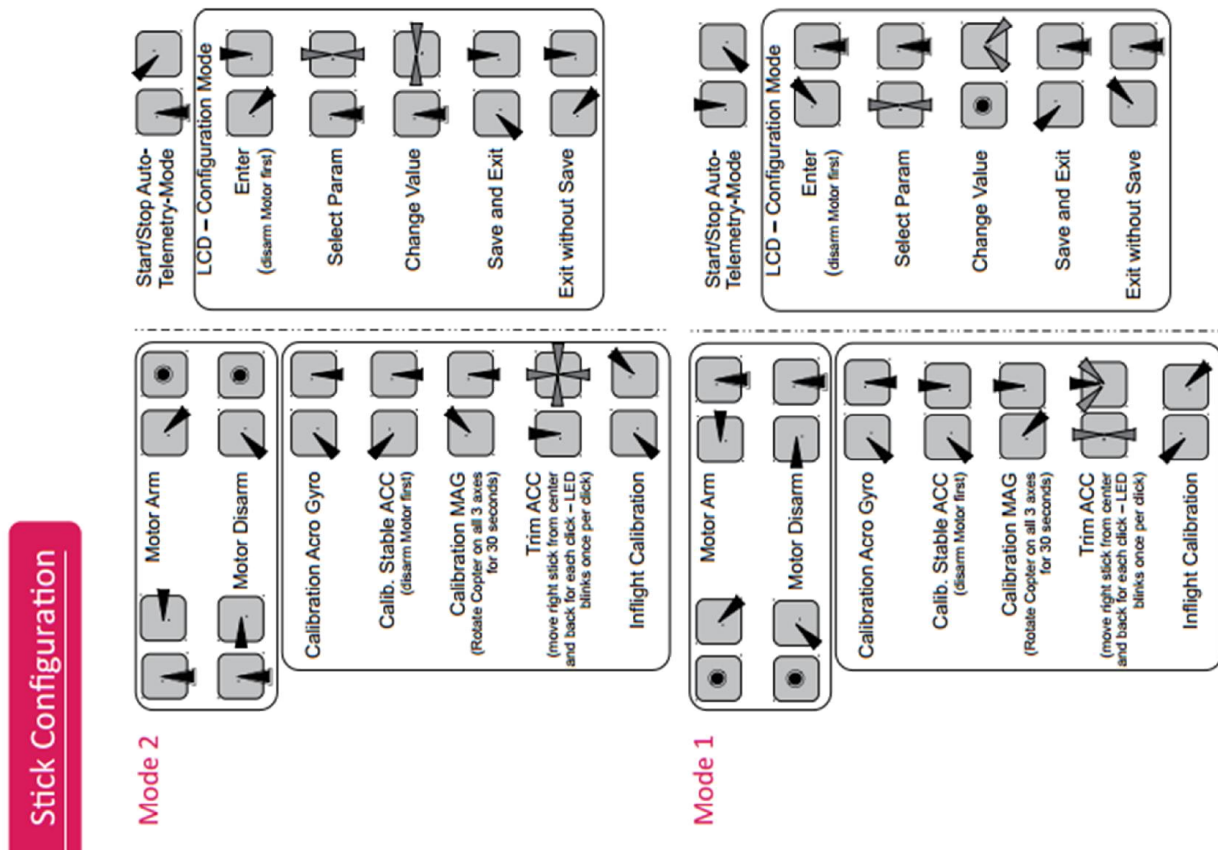


Timing diagram

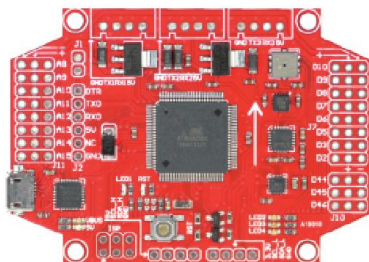
The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



8.8. STICK CONFIGURATION & MULTIWII PRO PINS



MWC MEGA v1.0



- 1x Atmega 2560, 256K Flash
- 9dof + Barometer (ITG 3205 / BMA 180 / HMC5883 / BMP085)
- 8x Motor output
- 3x Servo output for Camera
- 8CH PPM Input (incl. 2x AUX CH)
- USB Port on board
- 3 Status LED
- 4x Serial ports, supports GPS Function

Connection Plan

