

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A.MIRA Bejaïa
Faculté de Technologie
Département de Génie Electrique



Mémoire de fin d'étude

En vue de l'obtention du diplôme de Master en Automatique
Spécialité : Automatique et Informatique Industrielle

Thème

Réalisation et contrôle d'un Quadri Rotor

Présenté par :

M^r : HACHEMI Youcef

Encadré par :

P^r : MENDIL Boubekour

M^r : BELAID Amine

Examineurs :

M^r : HADDAR Hocine

M^{me} : BELLAHSENE Nora

Ce projet a été réalisé au sein du laboratoire de recherche des technologies industrielle et de l'information LT2I

Année universitaire 2017/2018

Remerciements

Je tiens à remercier tout d'abord le Dieu tout puissant pour la volonté, la santé et la patience qu'il nous a données durant toutes ces longues années.

Je tiens à remercier vivement Mr MENDIL Boubkeur, pour nous avoir honorés par son encadrement, ses orientations, ses précieux conseils et ses encouragements qui nous ont permis de mener à bien ce travail. Je remercie aussi Mr BELAID Amine mon co-encadreur pour son bon encadrement et le suivi du projet.

Je tiens à remercier Mr. MEZZAH Ibrahim et Mme. MEZZAH Samia qui m'ont encadré aussi pendant ma période de stage et m'ont orienté vers le monde professionnel.

Mes remerciements vont aussi à mes amis, ALLOU Hichem et METHIA Mounir qui m'ont aidé à réaliser ce projet.

Un énorme merci à ma famille et amis pour leur soutien et la confiance qu'ils ont en nos capacités.

S'il y a des personnes à ne pas oublier, c'est bien toute l'équipe du laboratoire de recherche LT2II.

Je tiens à exprimer ma gratitude aux membres du jury pour avoir accepté de juger ce travail.

Dédicaces

Je dédie ce modeste travail :

A mes chers parents que je remercie énormément pour leurs aides et leurs patiences ;

A ma très chère grande mère IMA AZZIZOU ;

A mes sœurs et frères (LISA, KATIA, FERHAT, YASSMINE, Mnd AREZKI) ;

A ma famille sans exception ;

A mes enseignants ;

A toutes les personnes qui nous ont apporté de l'aide.

Youcef

Table de matières

Introduction générale	1
1 Généralités sur les drones	2
1.1 Introduction	2
1.2 Brève histoire des drones	2
1.3 Classification des drones	3
1.4 Domaines d'application des drones	5
1.4.1 Utilisations dans le domaine civil	5
1.4.1.1 Agriculture	5
1.4.1.2 Sécurité Civile	6
1.4.1.3 Inspection	6
1.4.1.4 Recherche scientifique	6
1.4.1.5 Média	6
1.4.1.6 Humanitaire	7
1.5 Configurations des drones	7
1.5.1 Voilure Fixe	7
1.5.2 Hélicoptères	8
1.5.3 Multi rotors	8
1.5.4 Hybride	9
1.5.5 Ailes battantes	9
1.6 Conclusion	10

2	Modélisation et commande du Quadri-rotor	11
2.1	Introduction	11
2.2	Description générale du Quadri-rotor	11
2.3	Mouvements du Quadri-rotor	12
2.3.1	Mouvement vertical	13
2.3.2	Mouvement de lacet	13
2.3.3	Mouvement de Translations horizontales	14
2.3.3.1	Mouvement de translation selon l'axe x	14
2.3.3.2	Mouvement de translation selon l'axe y	15
2.4	Modélisation du quadri rotor	15
2.4.1	Le modèle dynamique non linéaire du quadri-rotor	15
2.4.2	Modèle simplifié du quadri-rotor	17
2.5	Commande et simulation du quadri-rotor	17
2.6	Conclusion	18
3	Description matérielle du Quadri-rotor	20
3.1	Introduction	20
3.2	Le Châssis	20
3.3	Les moteurs Brushless	22
3.4	Electronic Speed Control	23
3.5	Les helices	24
3.6	Carte Arduino	25
3.7	L'unité de mesure inertielle	26
3.8	Télécommande radio	26
3.9	La batterie	27
3.9.1	Tension / nombre de cellules	27
3.9.2	Capacité	28
3.9.3	Taux de décharge «C»	28

3.10	Conclusion	28
4	Réalisation et programmation du Quadri-rotor	29
4.1	Introduction	29
4.2	Schéma synoptique et principe de fonctionnement	29
4.3	Description électrique du système	30
4.3.1	Caractéristiques du microcontrôleur ATmega 328P	30
4.3.2	Réalisation du circuit de commande autour de la carte Arduino	31
4.3.3	Réalisation du circuit final	31
4.3.4	Mise en marche du circuit	34
4.4	Programme de contrôle du quadri-rotor	35
4.4.1	Conversion de signaux de commande en PWM	35
4.4.2	Compensation de décharge de la batterie	36
4.4.3	Calcul de la commande	36
4.4.4	Principe de fonctionnement du PID	37
4.4.4.1	Terme Proportionnel	37
4.4.4.2	Terme Intégral	37
4.4.4.3	Terme Dérivé	38
4.4.5	Technique d'Ajustement des paramètres du PID	38
4.5	Conclusion	39
	Conclusion générale	41

Table des figures

1.1	Le premier vrai drone fabriquer par les Américains Lawrence et Sperry en 1916	3
1.2	Quelques quadri-rotor développés récemment	4
1.3	L'aile volante eBee et Techpod	7
1.4	Hélicoptères	8
1.5	Le Falcon 8	8
1.6	Le Quadshot	9
1.7	Nano Air Vehicle d' Aerovironment	10
1.8	Le Robobee	10
2.1	Structure générale d'un Quadri-rotor	12
2.2	Mouvement vertical	13
2.3	Mouvement lacet	14
2.4	Translation selon l'axe x	14
2.5	Translation selon l'axe y	15
2.6	Le modèle Simulink du quadri-rotor	18
2.7	La réponse du quadri-rotor pour $k_p = 3.5$, $k_d = 0.5$ et $k_I = 0$	18
3.1	Châssis dji Q450 V3	21
3.2	Dessin technique du modèle Q450	21
3.3	Model 3D des pieds sur logiciel Repitier-Host	22
3.4	Structure d'un moteur Brushless	23

3.5	Electronic Speed Control	24
3.6	Hélices 10"4.5" calibré	25
3.7	Carte Arduino Uno	25
3.8	MPU 6050	26
3.9	Télécommande radio et récepteur	27
3.10	Batterie «LiPo»	27
4.1	Schéma synoptique du système	30
4.2	Circuit prototype du Quadri-rotor	31
4.3	Schéma du circuit sur Eagle	32
4.4	PCB final sur Eagle	33
4.5	PCB réalisé en Industrie	33
4.6	Circuit final	34
4.7	Schéma brochage ICSP	34
4.8	Utilisation de la carte Arduino comme Convertisseur UART - USB	35
4.9	Organigramme du programme principal	35
4.10	Principe de calcul de la PWM	36
4.11	Principe de fonctionnement du PID	37
4.12	Quadri-rotor réel en vol	39

Liste des acronymes

UAV	Unmanned Aerial Vehicle
NAV	Nano Air Vehicle d'Aerovironment
OVMII	Objet Volant Mimant l'Insecte
DDL	degrés de liberté
PID	Proportionnel Intégral Dérivé
BLDC	Brushless DC electric motor
RPM	rotation per minute
ESC	Electronic Speed Control
PWM	Pulse Width Modulation
CW	Clock Wise
CCW	Counter Clock Wise
ICSP	in-circuit serial programming
USB	Universal Serial Bus
IMU	Inertial Measurement Unit
I2C	Inter-Integrated Circuit
Li Po	lithium polymère
EEPROM	Electrically Erasable Programmable Read-Only Memory
SRAM	Static Random Access Memory
CAN	Convertisseur Analogique Numérique
SPI	Serial Peripheral Interface
USART	Universal Synchronous Asynchronous Receiver Transmitter
PCB	Printed Circuit Board
SMD	Surface Mount Components

Introduction générale

Introduction générale

Actuellement, la robotique est devenue une discipline extrêmement populaire. Les drones sont les robots qui ont attiré plus d'attention. Réservés au domaine militaire, récemment ils ont envahi le domaine grand public. Ils sont utilisés dans plusieurs domaines, tels que : l'agriculture, l'inspection des zones inaccessibles, les médias, etc.

Notre projet a pour but l'étude et la réalisation concertée d'un Quadri rotor. Pour cela, on a tout d'abord commencé à choisir le matériel nécessaire, afin d'avoir un ensemble homogène de composants. Ensuite, on a procédé à l'assemblage, la conception/réalisation d'un circuit électronique de commande et, enfin, la programmation et la mise en marche.

Le travail a été effectué en deux phases. Durant la phase de développement, le circuit de commande a été implémenté autour d'une carte Arduino. Après les tests de validation, on a procédé à la réalisation du circuit final. Le schéma d'implantation a été conçu avec le logiciel Eagle AutoDesk et le circuit imprimé PCB a été réalisé, sur commande, en industrie. L'implantation et la soudure des composants SMD utilisés a été faite avec le matériel nécessaire au niveau du centre de développement des technologies avancées d'Alger (CDTA). La dernière partie du travail concerne la mise en marche finale et les tests de vol du quadri rotor.

Pour une meilleure présentation, on a organisé le manuscrit en quatre chapitres. Le chapitre 1 présente des généralités sur les drones (description des UAV, historique, domaines d'application et les configurations existantes). Le chapitre 2 est consacré à la modélisation du Quadri-rotor et l'explication des différents mouvements, puis l'étude de simulation et résultats. Dans le chapitre 3, on a présenté tous le matériel utilisé et le choix des différents composants parmi une très grande gamme disponible sur le marché. Le chapitre 4 expose les étapes suivies pour la réalisation et la mise en marche du circuit de commande et sa programmation. Enfin, on termine avec une conclusion générale et les perspectives envisagées.

Chapitre 1

Généralités sur les drones

Chapitre 1

Généralités sur les drones

1.1 Introduction

Un drone ou Unmanned Aerial Vehicle (UAV) est un aéronef sans passager ni pilote qui peut voler de façon autonome ou être contrôlé à distance depuis le sol. Le mot « drone » est une extrapolation d'un terme anglais qui signifie « faux-bourdon ». En français, le terme est employé pour désigner des véhicules aériens, terrestres, de surface ou sous-marins, alors que la classification anglo-saxonne distingue chaque type d'appareil. La taille d'un drone aérien peut aller de quelques centimètres pour les modèles miniatures à plusieurs mètres pour les drones spécialisés. Dans ce premier chapitre, nous allons donner une brève histoire sur les drones ainsi que ces différents types et classifications et leurs domaines d'application.

1.2 Brève histoire des drones

En 1907 un hélicoptère à quatre rotors conçu par Louis Breguet, ce fut le premier avion à voilure tournante à se soulever du sol, mais seulement en vol attaché à une altitude de quelques pieds. Le premier vrai drone a été fabriqué par les Américains Lawrence et Sperry en 1916. Ils ont développé un gyroscope pour stabiliser le corps afin d'automatiser le pilotage. Ceci est connu comme le début du "contrôle d'attitude", qui est venu pour être utilisé pour le pilotage automatique. Puis, en 1924, Etienne Oehmichen a développé un hélicoptère avait quatre rotors et huit hélices, tous entraînés par un seul moteur. L'avion a présenté un degré considérable de stabilité et d'augmentation de la précision

de contrôle pour son temps. Il a établi le premier record de distance pour les hélicoptères de 360 m. Le modèle Quadri-rotor Convertawings conçu en 1956 était l'hélicoptère le seul qui devait être le prototype le plus adopté pour les deux secteurs civil et militaire. La conception comportait deux moteurs entraînant quatre rotors à travers un système de courroies trapézoïdales. Aucun mouvement de tête des hélices n'était nécessaire et le contrôle a été obtenu en faisant varier la poussée entre les rotors. Ce modèle aurait une masse maximale de 19 t avec une charge utile de 4,9 t sur 300 milles et jusqu'à 278 km/h [1] [2].



FIGURE 1.1 – Le premier vrai drone fabriqué par les Américains Lawrence et Sperry en 1916

Au cours des dernières décennies, des véhicules aériens sans pilote à petite échelle ont été utilisés pour de nombreuses applications. Le besoin d'avions avec une plus grande manœuvrabilité et une capacité de vol stationnaire a conduit à une augmentation de la recherche sur les quadcopters. La conception à quatre rotors permet aux quadcoptères d'être de conception relativement simple tout en étant très fiable et maniable. La recherche continue d'augmenter les capacités des quadcoptères en faisant des progrès dans la communication multimédias, l'exploration de l'environnement et la maniabilité. Si ces qualités de développement peuvent être combinées, les quadcoptères seraient capables de missions autonomes avancées qui ne sont actuellement pas possibles avec d'autres véhicules.

1.3 Classification des drones

Des recherches importantes ont été consacrées à l'augmentation de l'endurance au vol et de la charge utile des UAV, ce qui a donné lieu à diverses configurations d'UAV de

tailles, de niveaux d'endurance et de capacités différents. Ici, nous tentons de classer les UAV en fonction de leurs caractéristiques (configuration aérodynamique, taille, etc) [1].



FIGURE 1.2 – Quelques quadri-rotor développés récemment

Un UAV appartient généralement à l'une des quatre catégories suivantes :

- Les drones Fixed-wing : qui se réfèrent à des avions sans pilote qui nécessitent une piste pour décoller et atterrir, ou le lancement de catapultes. Ceux-ci ont généralement une longue endurance et peuvent voler à des vitesses de croisière élevées.
- Les drones Rotary-wing : aussi appelés drones à voilure tournante ou drones à décollage et atterrissage verticaux (VTOL), qui ont les avantages de la capacité de vol stationnaire et d'une grande manœuvrabilité. Ces capacités sont utiles pour de nombreuses missions robotiques, en particulier dans les applications civiles.
- Les dirigeables, qui sont plus légers que l'air et qui ont une longue endurance, volent à basse vitesse et sont généralement de grandes dimensions.
- Les drones à ailes battantes, dotés de petites ailes flexibles et / ou morphings inspirés par les oiseaux et les insectes volants.

Un autre critère utilisé pour différencier et classer les avions est la taille et l'altitude [23], il y a 5 classes :

- Drones à haute altitude et haute endurance (HALE), environ 20 km d'altitude et 40 h de vol.
- Les drones de moyenne altitude à longue endurance (MALE), environ 8 km d'altitude et 30/40 h de temps de vol.
- UAV tactique, environ 4 km d'altitude et 5-6 heures de vol.

- Mini drones, moins de 300 m d'altitude et moins de 2 heures de vol.
- Micro drone, moins de 250 m d'altitude et moins d'une heure de vol.

1.4 Domaines d'application des drones

Actuellement, les principales applications d'UAV sont liées à la défense et les principaux investissements sont motivés par de futurs scénarios militaires. La plupart des systèmes d'avions sans pilote militaires sont principalement utilisés pour le renseignement, la surveillance, la reconnaissance (ISR) et les grèves. La prochaine génération de drones exécutera des missions plus complexes telles que le combat aérien ; détection, reconnaissance et destruction de cibles ; frappe / suppression de la défense aérienne d'un ennemi ; attaque électronique ; nœud de réseau / relais de communication ; livraison / réapprovisionnement aérien ; guerre des navires anti-surface ; guerre anti-sous-marine ; la guerre des mines ; expédier à des manœuvres objectives ; contre-air offensif et défensif ; et le transport aérien.

Aujourd'hui, les marchés civils pour les drones sont encore émergents. Cependant, les prévisions de croissance du marché des drones civils et commerciaux seront très élevées pour la prochaine décennie. Les applications civiles potentielles des UAV sont inspection du terrain, des pipelines, des services publics, des bâtiments, etc [1]

1.4.1 Utilisations dans le domaine civil

On recense de manière non exhaustive dans le paragraphe suivant l'usage de mini-drones dans divers contextes civils.

1.4.1.1 Agriculture

Les utilisations agricoles des mini-drones sont très diverses [3]. Dotés de capteurs optiques ou hyperspectraux ces engins permettent de collecter un grand nombre de données dont le traitement va ensuite faciliter l'analyse de l'évolution des cultures, le repérage des mauvaises herbes ou des nuisibles. Ainsi, un suivi régulier peut être garanti pour un coût modeste. En outre, la mise en œuvre des drones est facilitée par l'absence de population sur les zones survolées.

1.4.1.2 Sécurité Civile

Les drones offrent un précieux secours dans la réponse immédiate à une catastrophe naturelle ou à un accident. Encore une fois, ils permettent le déploiement rapide de capteurs pour obtenir des informations sur la zone touchée. En plus de ces avantages, ils offrent des points de vue différents dans la collecte d'images par rapport à ceux d'un satellite. Cela facilite le post-traitement des images [4].

Des robots de toutes tailles ont été utilisés dans les récents événements [5]. Un Global Hawk a permis de cartographier les zones sinistrées lors du séisme d'Haïti en 2010, des drones T-Hawk ont permis d'approcher les réacteurs défectueux de Fukushima. Des quadrirotors ont permis de suivre l'évolution d'une marée noire en Alaska. Les ouragans Katrina et Wilma furent également des lieux d'expérimentations sur les modes opératoires des véhicules [6].

1.4.1.3 Inspection

L'inspection de bâtiments, de ponts ou d'éléments d'une architecture industrielle ainsi que des oléoducs ou des lignes à haute tension sont autant de tâches à effectuer régulièrement pour détecter le plus rapidement possible une dégradation ou défaillance potentielle [7] [8].

1.4.1.4 Recherche scientifique

Les drones sont des plateformes idéales pour la collecte de données scientifiques [9] [10]. L'archéologie a également bénéficié de cette technologie pour le relevé de sites. La surveillance de populations animales et de l'écosystème d'une zone a également été envisagée [11] [12].

1.4.1.5 Média

Une société professionnelle de journalistes utilisant des drones a été créée en novembre 2011 [13] pour tenter à la fois de proposer un cadre éthique à cette pratique et la promouvoir. Les micro-drones sont appréciés pour le tournage de séquences vidéo (Films, publicités ou événements sportifs). Ils permettent de réaliser des plans avec un minimum de matériel.

1.4.1.6 Humanitaire

La documentation de situations de guerre est encore un exemple parfait de l'usage de drones dans des conditions dangereuses pour l'homme [14]. L'entreprise Matternet imagine distribuer des médicaments rapidement à des zones sinistrées ou difficiles d'accès à l'aide d'un réseau de drones [15]. Enfin, on peut encore citer l'initiative Conservation Drones qui vise à développer des prototypes bas coût, simple d'accès pour la surveillance de forêts, l'évaluation de leur biodiversité et la détection d'activités illégales.

1.5 Configurations des drones

L'ensemble des applications se heurtent à plusieurs défis techniques. En premier lieu, celui du choix de la configuration du drone. Les drones peuvent se classer selon les catégories suivantes :

1.5.1 Voilure Fixe

Il s'agit de la configuration d'un avion conventionnel doté d'une paire d'ailes assurant la sustentation, un fuselage, une dérive et un empennage arrière. Ces configurations offrent le plus d'endurance, mais nécessitent de l'espace pour être opérées (décollage et atterrissage). L'aile volante eBee (figure 1.3) a par exemple 1 mètre d'envergure pour une vitesse de croisière de 10 m/s qu'elle peut assurer pendant 45 minutes. Plus grand, le Techpod vole en croisière à 16 m/s pendant 2 heures avec ses 2.5 mètres d'envergure. Une vitesse minimum leur est aussi nécessaire pour voler ce qui limite leur manœuvrabilité. Leur rayon d'action est par contre relativement important allant jusqu'à 100 kilomètres.



FIGURE 1.3 – L'aile volante eBee et Techpod

1.5.2 Hélicoptères

Ce sont des drones à voilure tournante munis d'un rotor principal et d'un rotor de queue ou d'un double rotor coaxial contrarotatif. Les rotors sont équipés d'un plateau collectif et d'un plateau cyclique pour commander l'hélicoptère. La possibilité de voler en stationnaire oriente vers le choix d'une telle plateforme. Les vitesses de translation et le rayon d'action des hélicoptères sont inférieurs à celle d'un avion à voilure fixe de dimensions comparables.



FIGURE 1.4 – Hélicoptères

1.5.3 Multi rotors

Équipés de 3, 4, 6 ou 8 rotors sans mécanisme particulier, ces drones utilisent les différences de vitesse entre des combinaisons de rotors pour s'orienter. Ils sont structurellement extrêmement robustes car ils se passent du plateau cyclique des hélicoptères classiques qui comprennent de nombreuses pièces fragiles. Les réparations peuvent souvent se limiter au remplacement d'une hélice. Le Falcon 8 (figure 1.5) peut se sustenter pendant 20 minutes et atteindre la vitesse de 10 m/s, mais surtout décoller avec une charge utile de 750g.



FIGURE 1.5 – Le Falcon 8

1.5.4 Hybride

Plus rares, ils se composent d'éléments des autres catégories. En général, ils disposent d'une ou plusieurs voilures pour fonctionner comme un avion, mais ont également la puissance et les mécanismes de commande pour le vol stationnaire. Ces configurations tentent d'allier la précision d'une voile tournante et l'endurance d'une voile fixe. Le Quadshot (figure 1.6), avec ses 1m d'envergure, peut voler environ 8 minutes en stationnaire et jusqu'à 15 minutes en vol d'avancement.



FIGURE 1.6 – Le Quadshot

1.5.5 Ailes battantes

Pour ces drones, la puissance et le contrôle sont assurés grâce à des ailes battantes à de hautes fréquences 120Hz ou à plus basses fréquences [16]. Dotées d'une capacité de vol stationnaire comme les voilures tournantes, ces configurations permettent des manœuvres agiles tout en étant plus discrètes que les voilures tournantes [17]. Cela en fait des plateformes adaptées pour une utilisation intérieure. Le NAV (figure 1.7) ne pèse lui que 20g pour des vitesses de déplacements allant jusqu'à 4.8 m/s pour ses 16 centimètres. Le Robobee préfigure des capacités futures : il bat ses ailes à 120 Hz pour sustenter sa masse de moins d'un dixième de gramme. Cependant il n'emporte pas l'énergie nécessaire au vol ni de système embarqué pour le contrôle. Ces deux éléments sont déportés. Le projet français OVMI (Objet Volant Mimant l'Insecte) [18] est un projet pluridisciplinaire qui a étudié la construction et la structure des ailes, la modélisation et la commande d'un micro drone à aile battante ainsi que la transmission d'énergie sans fil vers le drone.

Ces différentes configurations ont chacune leurs avantages et inconvénients qu'il s'agit de pondérer selon la mission envisagée.



FIGURE 1.7 – Nano Air Vehicle d' Aerovironment



FIGURE 1.8 – Le Robobee

1.6 Conclusion

Dans ce premier chapitre on a présenté les drones en générales ainsi qu'un historique, les principaux domaines d'application de ces véhicules récemment destinés au grand public et quelques configurations existantes. Dans les prochains chapitres on se focalise sur la configuration multi rotors plus précisément sur les Quadri rotor.

Chapitre 2

Modélisation et commande du
Quadri-rotor

Chapitre 2

Modélisation et commande du Quadri-rotor

2.1 Introduction

La modélisation des véhicules aériens se distingue des modélisations d'autres robots par leurs complexités. Car, ces derniers dépendent de plusieurs facteurs et paramètres. Dans ce chapitre, on va expliquer les différents mouvements d'un quadri rotor et les forces qui agissent sur ces engins volants. Ce qui conduit au modèle dynamique non linéaire du quadri rotor et la version simplifiée. On termine par une simulation.

2.2 Description générale du Quadri-rotor

Un quadri-rotor est un robot mobile aérien à quatre rotors définis dans l'espace par 6 degrés de liberté (ddl) [19] [20]. Ces 4 rotors sont généralement placés aux extrémités d'une croix. L'électronique de contrôle est habituellement placée au centre de la croix. Afin d'éviter à l'appareil de tourner sur lui-même autour de l'axe de lacet (Yaw), il est nécessaire que deux hélices tournent dans un sens, et les deux autres dans l'autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélice tournant dans le même sens soit placé aux extrémités opposées d'une branche de la croix.

Le fonctionnement d'un quadri-rotor est assez particulier. En faisant varier astucieusement les vitesses des moteurs, il est possible de le faire monter/descendre, de l'incliner à gauche/droite (Pitch) ou en avant/arrière (Roll) ou encore de le faire pivoter sur lui-

même (Yaw) [21]. Le quadri-rotor possède six ddl : trois mouvements de rotation et trois mouvements de translation. Ces six degrés doivent être commandés à l'aide de quatre déclencheurs seulement, donc c'est un système sous actionné (le nombre des entrées inférieure au nombre des sorties).

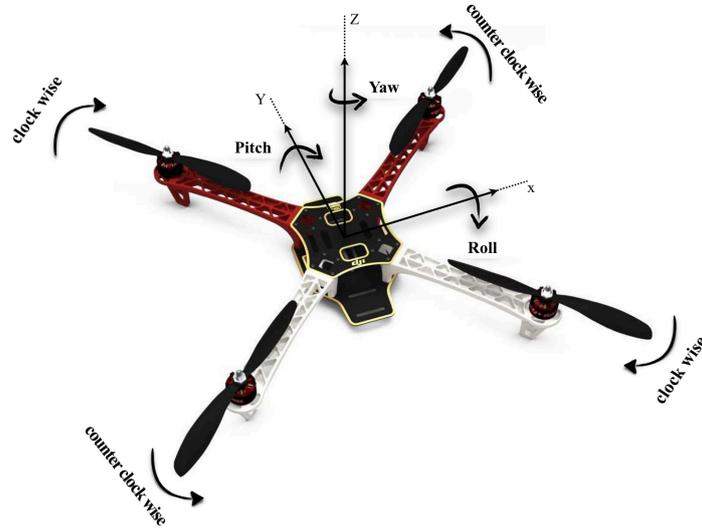


FIGURE 2.1 – Structure générale d'un Quadri-rotor

2.3 Mouvements du Quadri-rotor

Dans les quadri-rotors, le rotor avant gauche et le rotor arrière droit tournent dans le sens des aiguilles de la montre. Les deux autres (avant droit et arrière gauche) tournent dans le sens inverse. Ceci neutralise effectivement le couple réactif non désiré et permet au véhicule de planer sans tourner hors de la commande [22].

Les mouvements de base de quadri-rotor sont réalisés en variant la vitesse de chaque rotor, changeant de ce fait la poussée produite. Le quadri-rotor incline vers la direction des rotors plus lents, qui tient compte alors de la translation le long de cet axe. Par conséquent, comme à un hélicoptère classique, les mouvements sont couplés. Ce qui signifie que le quadri-rotor ne peut pas réaliser la translation sans roulement ou tangage. Ce qui signifie aussi qu'un changement de la vitesse d'un rotor se traduit dans un mouvement en au moins trois degrés de liberté. Par exemple, l'augmentation de la vitesse des propulseurs 1 et 4 aura comme conséquence un mouvement de roulis (le quadri-rotor s'incline). Lorsque l'équilibre entre les rotors qui tournent dans le sens des aiguilles d'une montre et les rotors qui tournent dans le sens inverse est perturbé, un mouvement de rotation horizontal

(lacet) autour de l'axe vertical est produit. Cet accouplement est la raison pour laquelle nous pouvons commander les six degrés de liberté de quadri-rotor avec seulement quatre commandes.

Le quadri-rotor a quatre mouvements principaux :

- Mouvement vertical
- Mouvement de lacet
- Translations horizontales selon l'axe x
- Translations horizontales selon l'axe y

2.3.1 Mouvement vertical

Afin de planer, toute la force de portance devrait seulement être le long de l'axe z avec une grandeur exactement opposée à la force de pesanteur. D'ailleurs, la force de portance créée par chaque rotor doit être la même que celles des autres rotors, pour empêcher le véhicule de se renverser.

Le mouvement ascendant et descendant est obtenu par la variation de la vitesse de rotation des moteurs. Si la force de portance est supérieure au poids du quadri-rotor, le mouvement est ascendant. Si la force de portance est inférieure au poids du quadri-rotor, le mouvement est descendant.

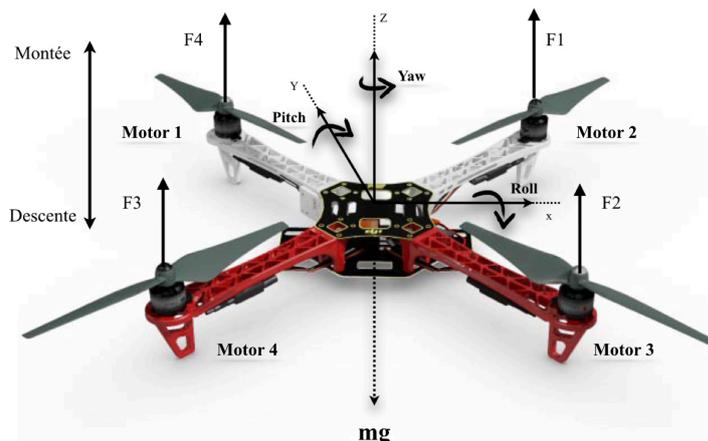


FIGURE 2.2 – Mouvement vertical

2.3.2 Mouvement de lacet

Afin de réaliser le mouvement du lacet (rotation autour de l'axe z), appelé aussi le mouvement yaw, on doit faire varier la vitesse d'une manière croisée des moteurs 1 et 3 ou bien 4 et 2 comme illustré dans la figure 2.3. Ce mouvement n'est pas un résultat direct de la poussée produite par les propulseurs. Mais, il est produit par les couples réactifs produits par la rotation des rotors. La direction de la force de poussée ne doit pas se décaler pendant le mouvement. L'augmentation de la force de portance dans une paire de rotors doit être égale à la diminution des autres paires, pour s'assurer que la force de poussée demeure la même.

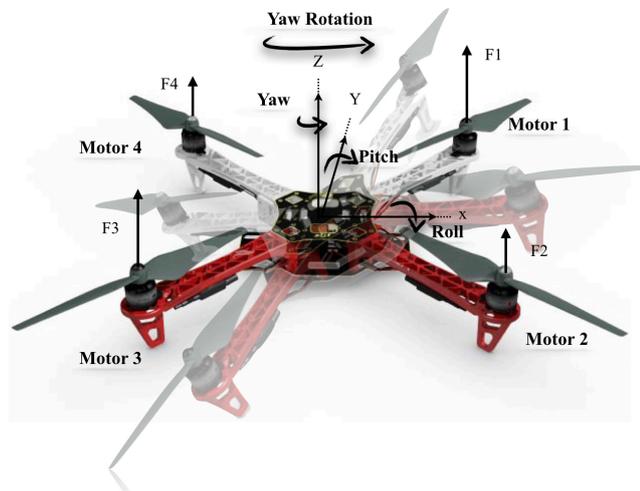


FIGURE 2.3 – Mouvement lacet

2.3.3 Mouvement de Translations horizontales

Les quadri-rotors peuvent faire deux mouvements de translation, selon les axes x et y . Ces translations permettent au quadri-rotor de se déplacer facilement dans l'espace.

2.3.3.1 Mouvement de translation selon l'axe x

La translation selon l'axe x est le résultat d'une rotation du véhicule au tour de l'axe y (pitch). Cette rotation est produite en augmentant la vitesse des moteurs 3 et 4, ce qui permet au drone de s'incliner d'un degré θ par rapport au Pitch. La figure 2.4 illustre davantage le principe de ce mouvement.

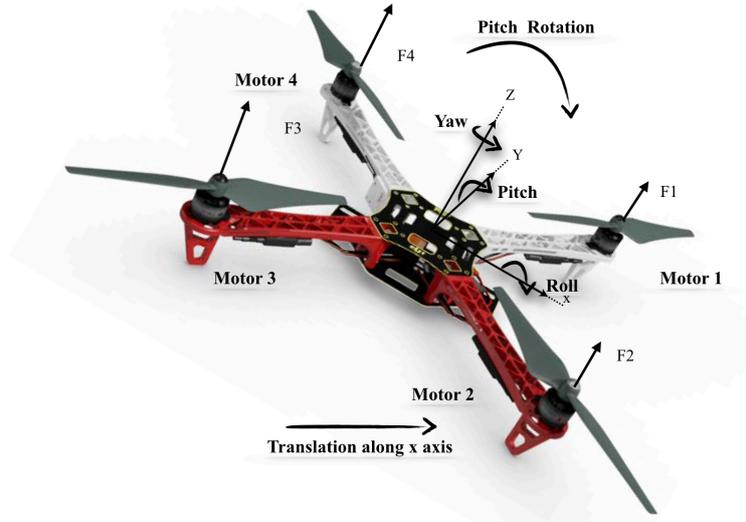


FIGURE 2.4 – Translation selon l'axe x

2.3.3.2 Mouvement de translation selon l'axe y

Cette translation n'est pas très différente de la précédente. Pour une translation le long de l'axe y, il faut une rotation d'un degré ϕ autour de l'axe x (roll). Pour ce faire, on doit agir sur la vitesse des moteurs 1 et 4, ce qui crée des forces F1 et F4 supérieures à F2 et F3.

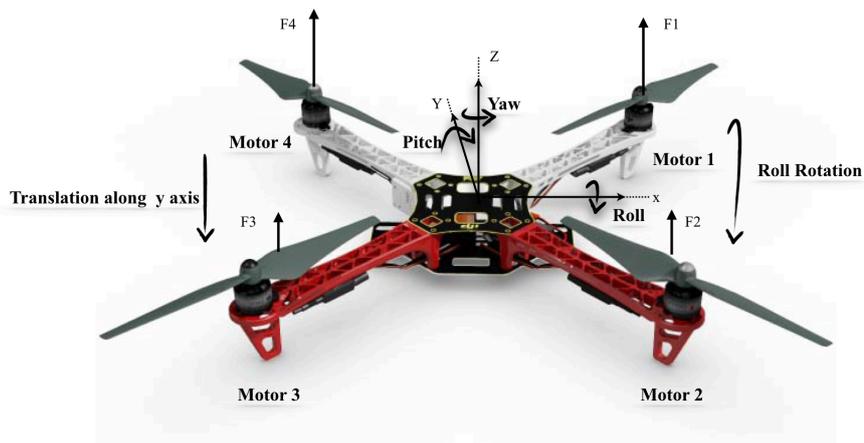


FIGURE 2.5 – Translation selon l'axe y

2.4 Modélisation du quadri rotor

La commande du quadri-rotor est une phase compliquée. Car, ce système est purement non linéaire avec six d.d.l et seulement quatre entrées (les quatre vitesses des moteurs). Dans cette section, on va présenter le modèle dynamique non linéaire du quadri-rotor, puis sa version simplifiée. Une étude de simulation de la dynamique du quadri-rotor sous MATLAB, en utilisant le modèle simplifié et en appliquant le contrôleur PD, fait l'objet de la section 2.5.

2.4.1 Le modèle dynamique non linéaire du quadri-rotor

Plusieurs forces agissent sur le quadri-rotor. Dans cette modélisation, on néglige toutes les forces de frottements. On suppose que le quadri-rotor est symétrique Ce qui donne une matrice d'inertie diagonale. On suppose aussi que la forme du quadri-rotor est rigide [23]. La dynamique des mouvements de translation est donnée par l'équation de Newton :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathcal{R}_A^B \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \quad (2.1)$$

Tel que : \mathcal{R}_A^B : la matrice de rotation entre le repère mobile B attaché au centre de gravité du quadri-rotor et le repère fixe A , obtenue selon la convention Z-X-Y (yaw, roll, pitch), où les rotations sont faites autour des axes mobiles instantanés :

$$\mathcal{R}_A^B = \text{rot}(\psi)_Z . \text{rot}(\phi)_X . \text{rot}(\theta)_Y \quad (2.2)$$

m : masse totale du quadri-rotor.

g : gravitation.

U_1 : signal d'entrée qui contrôle le déplacement selon l'axe Z_b exprimé en fonction des forces de poussée générées par les hélices :

$$U_1 = F_1 + F_2 + F_3 + F_4 \quad (2.3)$$

Tel que : $F_i = k_f \omega_i^2$, $i = 1..4$; k_f : le coefficient de portance ; ω_i : la vitesse du moteur.

La dynamique de rotation est donnée par l'équation d'Euler :

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} LU_2 \\ LU_3 \\ LU_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.4)$$

p, q, r : sont des vitesses angulaires exprimées dans le repère mobile, tel que : p correspond au mouvement de roulis (roll), q au tangage (pitch) et r au lacet (yaw).

L : est la distance entre le center du quadri-rotor et l'axe du moteur.

La matrice d'inertie symétrique du quadri-rotor, selon les trois axes, est donnée par :

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (2.5)$$

Les signaux d'entrée U_2 et U_3 contrôlent les angles ϕ et θ respectivement. Ce qui fait déplacer le quadri-rotor selon l'axe Y_b est selon l'axe X_b . Ces entrées sont exprimées en fonction des forces de poussé générées par les hélices :

$$U_2 = (F_1 + F_2) - (F_3 + F_4) \quad (2.6)$$

$$U_3 = (F_1 + F_4) - (F_2 + F_3) \quad (2.7)$$

Le signal d'entrée U_4 contrôle l'angle ψ exprimé en fonction des forces de traînée générée par les hélices :

$$U_4 = M_1 - M_2 + M_3 - M_4 \quad (2.8)$$

Tel que : $M_i = k_d \omega_i^2$, $i = 1..4$; k_d : le coefficient de drag.

2.4.2 Modèle simplifié du quadri-rotor

Quand le quadri-rotor fonctionne autour du point d'équilibre c'est-à-dire avec des angles proches de zéro, le modèle dynamique simplifié est obtenu en implémentant les approximations suivantes :

$$\sin(\phi) \approx \phi, \quad \sin(\theta) \approx \theta, \quad \cos(\phi) \approx 1, \quad \cos(\theta) \approx 1, \quad \psi \approx \psi_0$$

La dynamique de translation devient :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} (\theta \cdot \cos(\psi) + \phi \cdot \sin(\psi)) \cdot U_1 \\ (\theta \cdot \sin(\psi) - \phi \cdot \cos(\psi)) \cdot U_1 \\ U_1 - mg \end{bmatrix} \quad (2.9)$$

La dynamique de rotation devient :

$$I \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} LU_2 \\ LU_3 \\ LU_4 \end{bmatrix} \quad (2.10)$$

2.5 Commande et simulation du quadri-rotor

Dans ce projet, on n'a utilisé que deux capteurs (un gyroscope et un accéléromètre) pour mesurer l'attitude du quadri-rotor, alors que les positions (x,y,z) ne sont pas mesurées. C'est-à-dire, on ne peut contrôler que l'attitude du quadri-rotor en boucle fermée. Pour la position, l'opérateur constitue le feedback de la boucle de contrôle. Les quatre entrées de référence du système sont les quatre signaux envoyés par la télécommande. Puis, à partir de ces signaux on calcule les angles désirés $(\phi_d, \theta_d, \psi_d)$. Ensuite, on calcule les entrées de contrôle du système (U_1, U_2, U_3, U_4) en utilisant le contrôleur PID.

Le modèle utilisé dans cette simulation est le modèle simplifié (2.9) (2.10). Les paramètres du quadri-rotor sont donnés dans le tableau 2.1. La figure 2.6 présente le modèle Simulink du quadri-rotor . Le gyroscope et l'accéléromètre sont représentés par le bloque nommé GyroACC. Après plusieurs ajustements sur les paramètres de PID, on a obtenu la réponse présentée par la figure 2.7. On voit qu'elle est rapide et ne possède pas de dépassement. Les valeurs retenues des paramètres du régulateur sont : $k_p = 3.5, k_d = 0.5$ et $k_I = 0$.

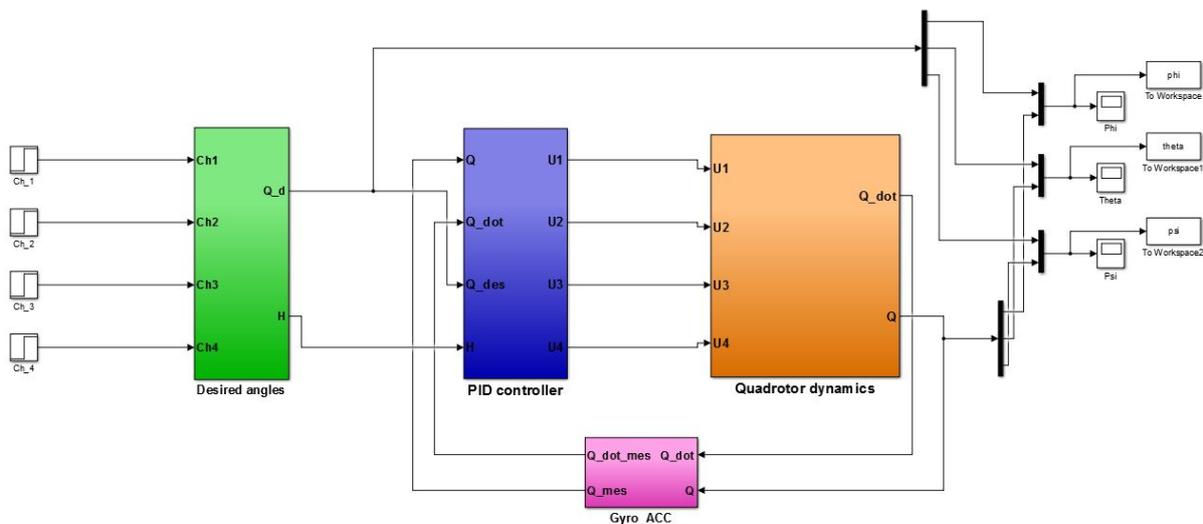
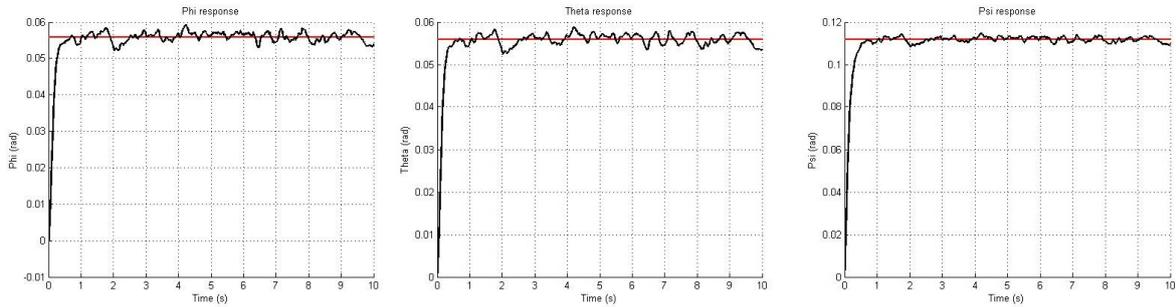


FIGURE 2.6 – Le modèle Simulink du quadri-rotor

TABLE 2.1 – Paramètres du quadri-rotor

Paramètre	Désignation
$I_x = I_y = I_z = 5.10^{-3}$	Les moments d'inertie du quadri-rotor autour des axes x, y, z
$k_f = 3.13.10^{-5}$	Le coefficient de portance
$k_d = 7.5.10^{-7}$	Le coefficient de drag
$L = 0.45$	La distance entre le centre du quadri-rotor et l'axe d'un moteur
$m = 0.95$	La mass totale du quadri-rotor
$g = 9.81$	Accélération gravitationnelle

FIGURE 2.7 – La réponse du quadri-rotor pour $k_p = 3.5$, $k_d = 0.5$ et $k_I = 0$

2.6 Conclusion

Dans ce chapitre, on a présenté le principe de fonctionnement du quadri-rotor en expliquant ses différents mouvements. Ensuite, on a exposé son modèle dynamique. Les résultats de simulation sous MATLAB montrent que, avec des valeurs bien ajustées des gains, le contrôleur PID donne une bonne réponse. C'est avec ces valeurs obtenues que nous allons commencer les essais dans le quadri-rotor réel ou les détails sont bien expliqués au prochain chapitre.

Chapitre 3

Description matérielle du
Quadri-rotor

Chapitre 3

Description matérielle du Quadri-rotor

3.1 Introduction

Le but principal de notre projet est l'étude et la réalisation concrète d'un quadri rotor. Ce chapitre est consacré à la présentation des différentes parties matérielles constituant le quadri-rotor. L'accent est mis sur les caractéristiques techniques de chaque composant, quelle que soit sa nature (mécanique, électronique ou bien électromécanique). Ce qui a permis de faire le choix adéquat de chaque élément parmi une diversité de composants disponibles sur le marché.

3.2 Le Châssis

Le choix du châssis n'est pas facile. Car, il existe une très vaste gamme des châssis de quadri-rotor. Chacun est différent de l'autre dans sa taille, sa forme ou le matériau qui le compose. Pour notre réalisation, on a utilisé un châssis de marque dji modèle Q450 V 3 de forme X construit à partir de fibre de verre, pour les deux plaques du haut et du bas et des bras en nylon polyamide, pour seulement un poids de 270g. L'avantage du châssis choisi est qu'il est doté d'un circuit de distribution de puissance intégré dans sa plaque du bas, afin d'éviter l'encombrement.



FIGURE 3.1 – Châssis dji Q450 V3

La figure 3.2 représente le dessin technique du modèle Q450. Vu que la hauteur du châssis est de 40 mm, la plaque basse touche le sol. Ce qui provoque des chocs à l'atterrissage du véhicule. Alors, on a pensé à l'améliorer en ajoutant des pieds imprimés avec une imprimante 3D (figure 3.3), afin de ramener la hauteur à 55 mm.

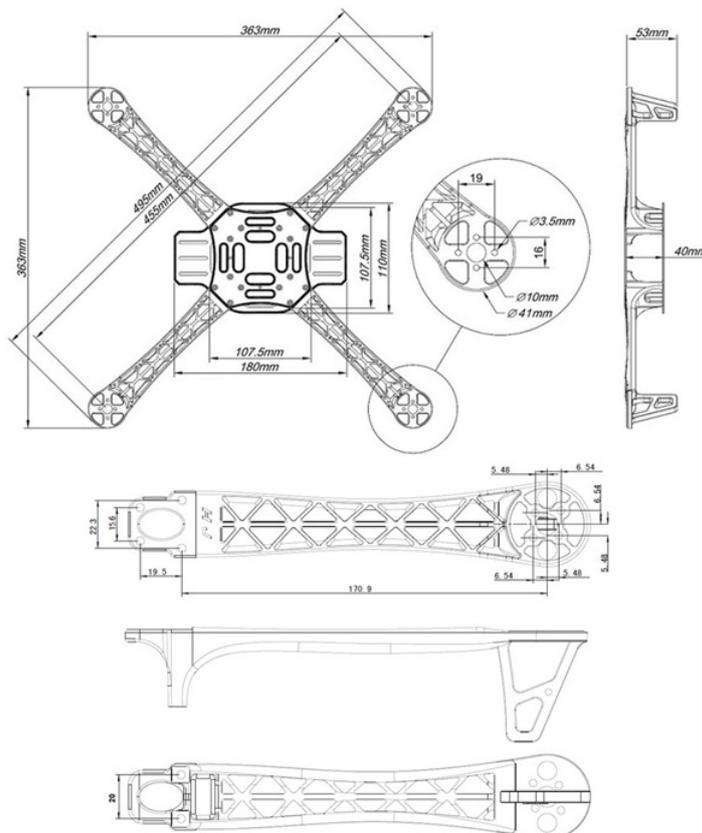


FIGURE 3.2 – Dessin technique du modèle Q450

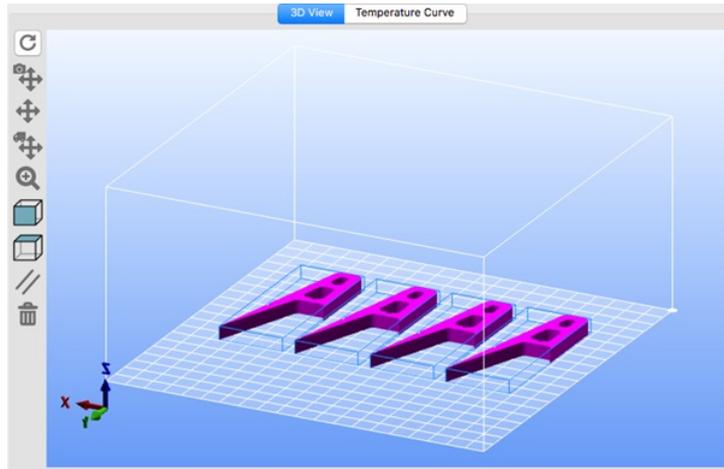


FIGURE 3.3 – Model 3D des pieds sur logiciel Repitier-Host

3.3 Les moteurs Brushless

Les moteurs à courant continu sans balais (BLDC) sont de plus en plus populaires. Les moteurs BLDC sont utilisés dans des industries telles que l'électroménager, l'automobile, l'aérospatiale, la médecine, l'équipement d'automatisation industrielle et l'instrumentation.

Comme son nom l'indique, les moteurs BLDC n'utilisent pas de brosses pour la commutation. A la place, ils sont interconnectés électroniquement. Les moteurs BLDC présentent de nombreux avantages par rapport aux moteurs à courant continu et aux moteurs à induction. On peut citer :

- Meilleure vitesse par rapport aux caractéristiques de couple
- Réponse dynamique élevée
- Longue durée de vie
- Fonctionnement sans bruit
- Gammes de vitesses plus élevées

De plus, le rapport du couple fourni à la taille du moteur est plus élevé. Ce qui le rend très indiqué dans les applications où l'espace et le poids sont des facteurs critiques [24]. C'est le cas dans notre projet.

Lorsque le moteur tourne, le rotor à aimants permanents qui passe devant les bobines du stator induit un potentiel électrique dans les bobines appelé *Back Electromotive Force*, ou BEMF. Ce potentiel est directement proportionnel à la vitesse du moteur est déterminé à partir de la constante de tension du moteur K_v . Comme on l'utilise, on se réfère à la

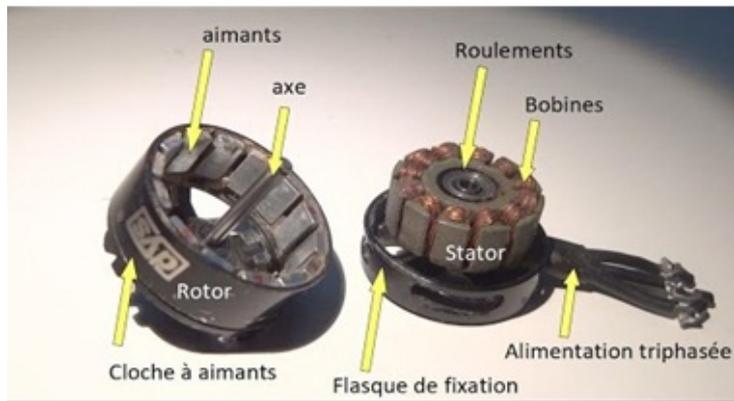


FIGURE 3.4 – Structure d'un moteur Brushless

constante de régime d'un moteur, c'est le nombre de tours par minute (RPM) que le moteur va tourner quand 1v (un volt) est appliqué sans charge attachée au moteur [25].

$$RPM = K_v \times Volts \quad (3.1)$$

Dans notre propre réalisation, on a choisi des moteurs de 1000 K_v . Car, plus K_v augmente, la vitesse augmente et le couple diminue. Dans l'autre cas, plus K_v est faible la vitesse diminue et le couple devient plus important. Un couple élevé permet de générer une force de poussée suffisante pour soulever le quad copter. Pour alimenter les bobines du moteur, on a utilisé le circuit puissance ESC (*Electronic Speed Control*).

3.4 Electronic Speed Control

Les moteurs Brushless sont de plus en plus fréquemment utilisés en raison de leurs performances nettement supérieures à celles des moteurs à courant continu « classiques ». Mais, contrairement à ces derniers, leur commande est plus complexe, étant donné qu'il s'agit de moteurs synchrones : il faut générer 3 signaux de fréquence proche de la fréquence de rotation, et déphasés de 120° . L'utilisation d'un ESC est la méthode la plus simple. Car, un ESC est un circuit électronique dédié au contrôle de moteurs électriques. Ils possèdent un microcontrôleur, un circuit de puissance (régulation, pont en H) et dans le cas des moteurs brushless, un dispositif d'acquisition. Ils permettent de gérer :

- La vitesse angulaire
- La direction
- Le freinage

Ils sont couramment utilisés dans les systèmes radiocommandés et, par conséquent,

possèdent une interface de commande de type servomoteur. Comme on le voit sur la figure 3.5 on a deux fils d'alimentation en continue puis un bronchement de commande depuis microcontrôleur pour générer une PWM comprise entre 1000 μ s et 2000 μ s, puis les trois fils pour le moteur Brushless. Le modèle choisit dans notre projet c'est des ESC de 30A. Ce qui donne une marge de sécurité pour ne pas abîmer les ESC [25].



FIGURE 3.5 – Electronic Speed Control

3.5 Les hélices

Le choix des moteurs et des hélices pour un drone est sans doute la tâche la plus fastidieuse lors de la conception d'un drone. Les couples moteurs/hélices sur un drone sont aussi importants que les couples moteurs/roues sur une voiture. En effet, les drones de type Quadri-rotor ne reposent que sur ces deux éléments pour défier les lois de la gravité. Les caractéristiques techniques des hélices présentent deux valeurs : la longueur des pales et le pas géométrique, exprimés tous les deux en pouces. Le pas c'est la distance que l'hélice parcourt en faisant un tour complet sans « glisser », autrement dit, en gardant une trajectoire linéaire. Le pas d'une hélice varie en fonction du nombre des pals de celle-ci (généralement deux ou trois pals), du poids de la machine et de la pression atmosphérique. C'est pourquoi il est important de faire la distinction entre le pas géométrique, qui est la valeur théorique et inflexible, et le pas effectif, qui est le pas réel.

Pour notre quadri rotor nous avons choisi des hélices a deux pals de dimension 10"4.5" en plastique, deux types ClockWise (sens horaire) et deux autres Counter ClockWise (sens trigonométrique). Ces derniers ne tournent pas d'une manière linéaire pour cela, on doit les calibrer. On s'est inspiré de la manière de calibrage d'une roue de voiture en utilisant un plomb sur l'extrémité de la jante. Dans le cas d'une hélice, un bout de ruban adhésif

isolant suffit pour la faire tourner correctement, afin d'éviter les vibrations qui vont falsifier les mesures du gyroscope.



FIGURE 3.6 – Hélices 10"4.5" calibré

3.6 Carte Arduino

Arduino Uno est une carte basée sur un microcontrôleur ATmega328P. Il possède 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un quartz à 16 MHz, une connexion USB, une prise d'alimentation, un connecteur ICSP et un bouton de réinitialisation. Il contient tout le nécessaire pour supporter le microcontrôleur [26].

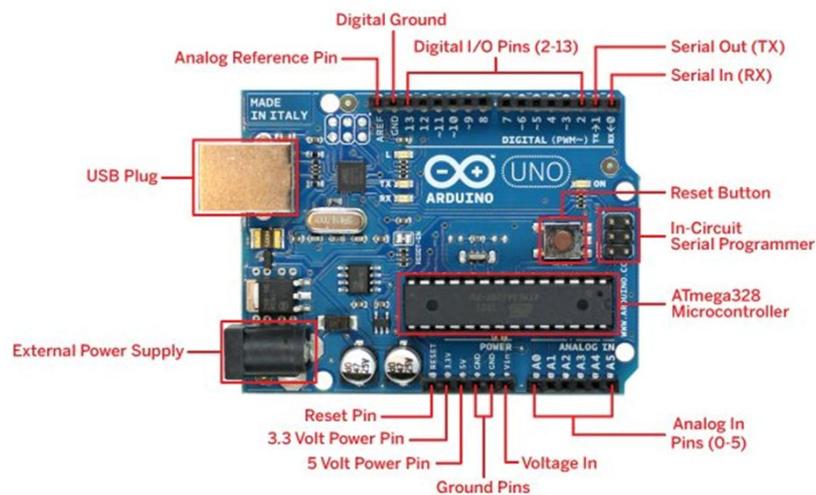


FIGURE 3.7 – Carte Arduino Uno

3.7 L'unité de mesure inertielle

Les capteurs IMU (Inertial Measurement Unit) sont l'un des types les plus inévitables de capteurs utilisés aujourd'hui dans toutes sortes de gadgets électroniques. On les voit dans les smartphones les contrôleurs de jeu, etc. Les capteurs IMU nous aident à obtenir l'attitude d'un objet, attaché au capteur dans un espace tridimensionnel. Ces valeurs généralement en angles nous aident donc à déterminer l'attitude du quadri-rotor. Ainsi, ils sont utilisés dans les smartphones pour détecter son orientation. On a pu travailler avec un accéléromètre et un gyroscope séparément. Cependant, ils ne sont pas aussi précis que lorsqu'ils sont combinés. Et parmi le lot, on a trouvé que le MPU 6050 d'Invensense était le capteur IMU le plus fiable et le plus précis. En plus, du faible coût par rapport aux autres capteurs, le MPU 6050 fonctionne beaucoup mieux. D'ailleurs, c'est le seul capteur utiliser. La communication avec le microcontrôleur se fait via un bus I2C. Le capteur renvoie deux valeurs principales : l'accélération selon x,y,z mesurée avec son accéléromètre et l'orientation mesurée avec le gyroscope [27].

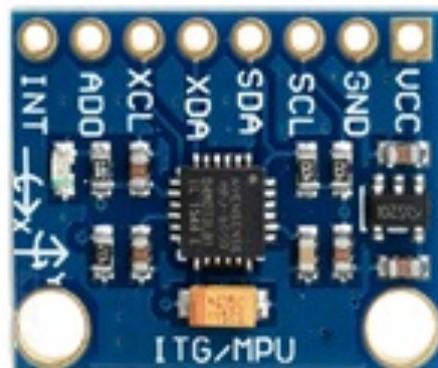


FIGURE 3.8 – MPU 6050

3.8 Télécommande radio

Une Télécommande radio est un instrument permettant de commander un robot à distance. Elle est généralement équipée de deux leviers permettant le contrôle de la puissance des moteurs et de la direction. Nous avons choisi la télécommande radio FLYSKY FS-i6 c'est une télécommande à six chaînes, mais pour commander le quad copter on n'a besoin que de quatre chaînes (Roll, Yaw, Pitch, Accélération). Ce modèle de télécommande donne en sortie une modulation par position de pulse (PPM) [28].



FIGURE 3.9 – Télécommande radio et récepteur

3.9 La batterie

Les batteries au lithium polymère (appelées batteries «LiPo») sont un type de batterie plus récent qui est maintenant utilisé dans de nombreux appareils électroniques. Ils ont gagné en popularité dans l'industrie de la radiocommande au cours des dernières années. On utilise ce type de batterie pour leur puissance élevée et un temps de charge plus long. Nous avons choisi une batterie de 3 cells de 3,7v, 2200mAh 30c, mais que signifient ces paramètres ?

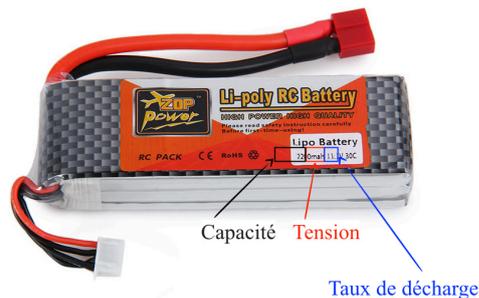


FIGURE 3.10 – Batterie «LiPo»

3.9.1 Tension / nombre de cellules

Une cellule LiPo a une tension nominale de 3,7V. Donc, un pack à deux cellules (2S) est à 7.4V, un pack à trois cellules (3S) à 11.1V et ainsi de suite. Pour la batterie 11.1V

ci-dessus, cela signifie qu'il y a trois cellules en série (la tension est additionnée).

3.9.2 Capacité

La capacité d'une batterie est essentiellement une mesure de la puissance de la batterie. L'unité de mesure est ici exprimée en milliampères par heure (mAh). Par exemple la batterie peut générer 2,2 A pendant une heure.

$$2200mAh = 2,2A/heure$$

3.9.3 Taux de décharge «C»

La notation « C » est simplement une mesure de la vitesse à laquelle la batterie peut être déchargée en toute sécurité et sans nuire à la batterie.

3.10 Conclusion

Ce chapitre est consacré à la présentation des différentes parties matérielles constituant le quadri-rotor. L'étude des caractéristiques techniques nous a permis de faire le choix d'une bonne combinaison d'éléments parmi une variété de composants disponibles sur le marché. La conception, la réalisation et la programmation du système seront présentées dans le prochain chapitre.

Chapitre 4

Réalisation et programmation
du Quadri-rotor

Chapitre 4

Réalisation et programmation du Quadri-rotor

4.1 Introduction

Ce chapitre expose le fond de notre travail. Il s'agit de la présentation des détails de la réalisation du système, tels que : la conception, le choix et le dimensionnement des composants, étapes de réalisation, etc. La dernière partie du chapitre est consacrée à la programmation et la mise en marche du système.

4.2 Schéma synoptique et principe de fonctionnement

Le fonctionnement global du Quadri-rotor est illustré par la figure 4.1. Un module de communication radio permet à l'utilisateur d'envoyer ses commandes de pilotage à travers la télécommande. Ces dernières servent de consignes pour l'unité de commande. Cette unité reçoit aussi l'état (position et orientation) du Quadri-rotor à travers le module de mesure (Gyroscope et Accéléromètre) et génère les signaux de commande de modulation en largeur d'impulsion PWM pour l'étage de puissance qui alimente les quatre actionneurs qui font tourner les hélices. Pour une meilleure stabilité, on introduit la tension de la batterie à l'unité de commande afin qu'elle soit compensée par la commande en cas de décharge.

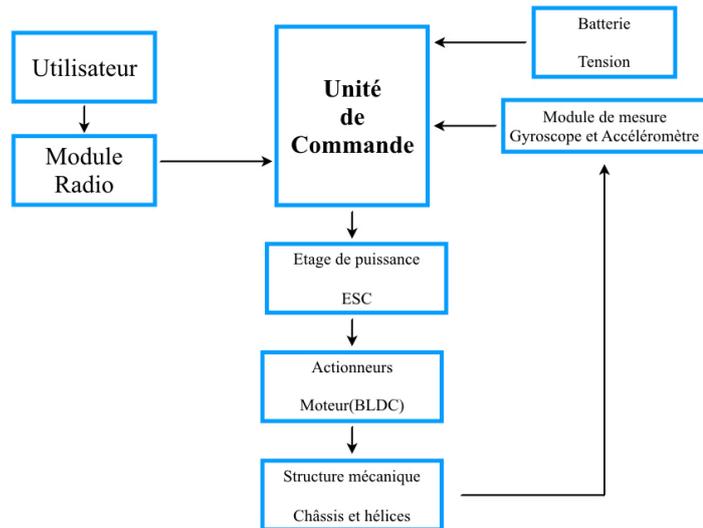


FIGURE 4.1 – Schéma synoptique du système

4.3 Description électrique du système

On commence d’abord par la présentation du noyau de notre réalisation qui est le microcontrôleur ATmega 328P.

4.3.1 Caractéristiques du microcontrôleur ATmega 328P

L’ATmega328 / P offre les fonctionnalités suivantes : 32 Ko de mémoire Flash programmable intégrée avec capacités lecture / écriture, 1 Kbytes d’EEPROM, 2 Kbytes de mémoire SRAM, 23 lignes d’E / S à usage général, 32 registres de travail à usage général, compteur temps réel (RTC) , trois temporisateurs / compteurs flexibles avec modes de comparaison et PWM, 1 USART programmable en série, 1 interface série 2 fils orientés octet (I2C), un CAN 10 bits à 6 canaux (8 canaux dans les packages TQFP et QFN / MLF), un temporisateur de Watchdog programmable avec oscillateur interne et un port série SPI [26].

Ce microcontrôleur répond bien aux besoins de notre réalisation, vu sa disponibilité, son coût réduit et sa facilité de programmation. En plus, il dispose des ressources et des performances nécessaires pour la commande temps réel du quadri-rotor.

4.3.2 Réalisation du circuit de commande autour de la carte Arduino

Au début de la réalisation de notre projet, on a utilisé une carte Arduino Uno basée sur un microcontrôleur Atmega 328 P, avec une carte perforer montée sur la carte comme shield, afin de mettre les composants nécessaires et de faire les essais et de tester les différents programmes (figure 4.2). Sur la figure on voit récepteur radio, une LED d'indication, diode de protection et deux résistances. Pour plus d'ergonomie, le module capteur MPU6050, l'étage de puissance (*Electronic Speed Control*) et les moteurs sont montés directement sur le châssis du quadri-rotor. Le principe de fonctionnement et la description électrique du système et la même que celle du circuit final, de la figure 4.4, qui sera exposé dans la section suivante.

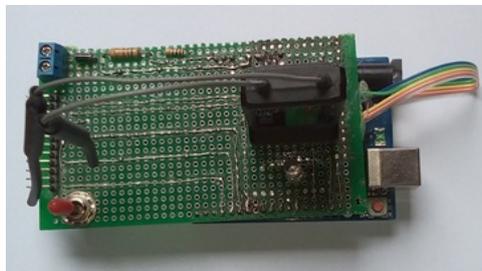


FIGURE 4.2 – Circuit prototype du Quadri-rotor

4.3.3 Réalisation du circuit final

Après avoir finalisé les travaux sur la carte de développement, on a conçu un circuit qui embarque tous les composants électroniques, afin d'avoir une forme compacte, prendre une place fixe, pour éviter les vibrations durant le vol, et bien sûr, côté esthétique, pour avoir un bon design final du quadri-rotor.

La conception du circuit est faite sur logiciel Eagle Autodesk en deux parties. La première concerne le schéma électrique (figure 4.3) et la seconde concerne le schéma d'implantation PCB (*Printed Circuit Board*).

Après la schématisation du circuit, on passe à la conception du circuit final du PCB où on a donné la forme précise. Sur le logiciel Eagle, on choisit les composants avec les packages (formes et brochages) désirés. On les a placés sur la carte, puis on a fait le routage. Après vérification, on a passé la commande de réalisation en industrie pour avoir un circuit professionnel.

Vu que nous avons choisi des composants SMD, il n'est pas évident de les souder sur

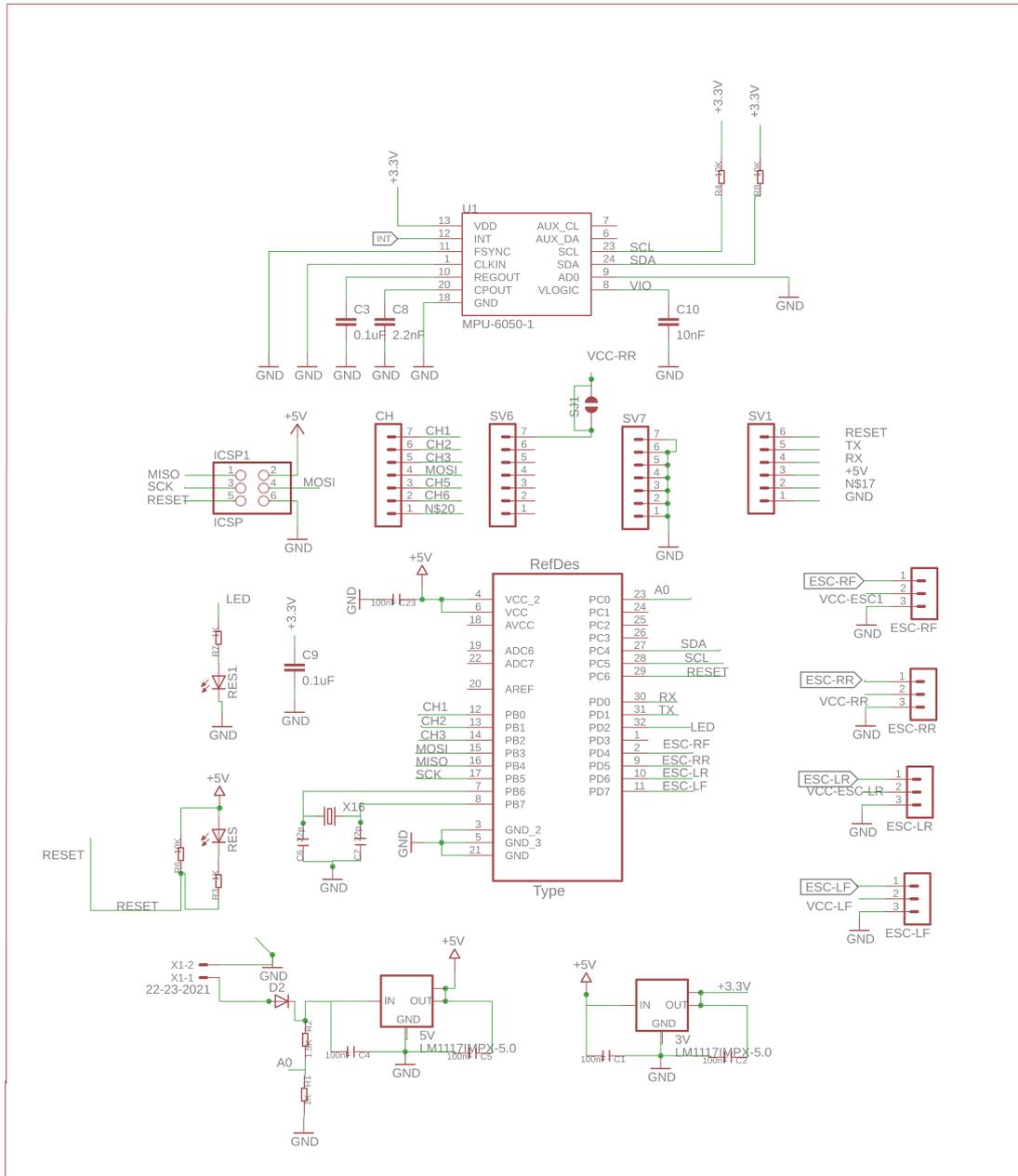


FIGURE 4.3 – Schéma du circuit sur Eagle

la carte avec un fer à souder. Pour cela, on s'est déplacé au Centre de Développement des Technologies Avancées (CDTA), pendant notre période de stage, pour souder les composants d'une manière professionnelle en utilisant le matériel nécessaire.

Les étapes suivies pour le soudage :

- Application de pâte à braser sur les pistes de soudage.
- Placement des composants dans leur place à l'aide d'un microscope, une caméra microscopique et une machine pick and place.
- Mise du circuit au four pendant 9 minutes.
- Vérification de toutes les pistes s'il n'y a pas de court circuit puis correction.

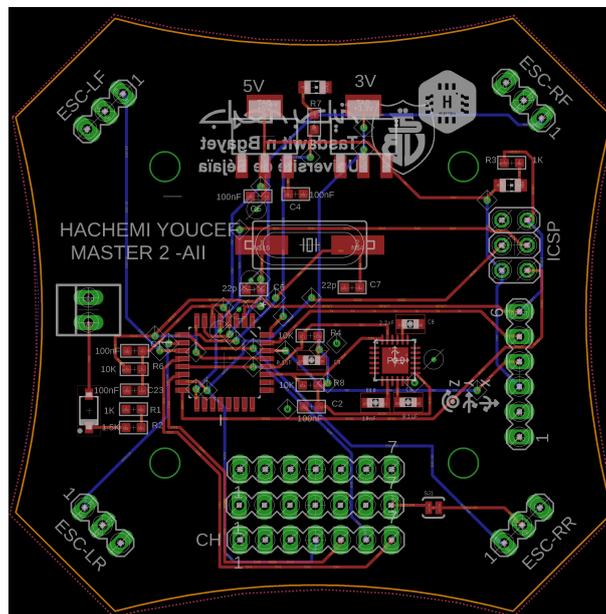


FIGURE 4.4 – PCB final sur Eagle

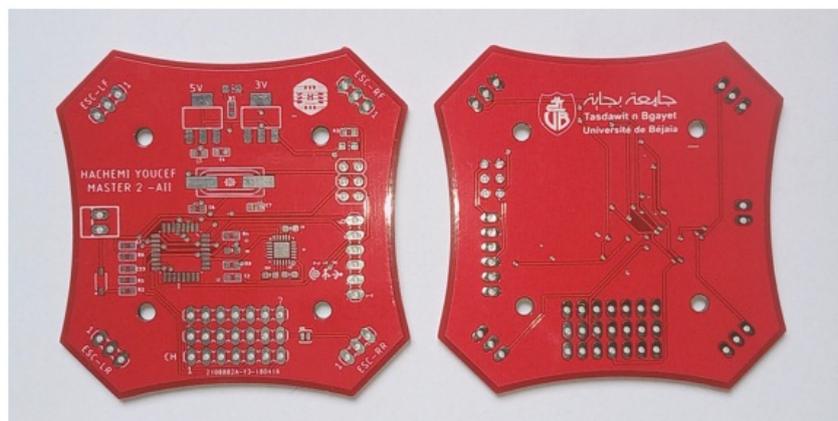


FIGURE 4.5 – PCB réalisé en Industrie

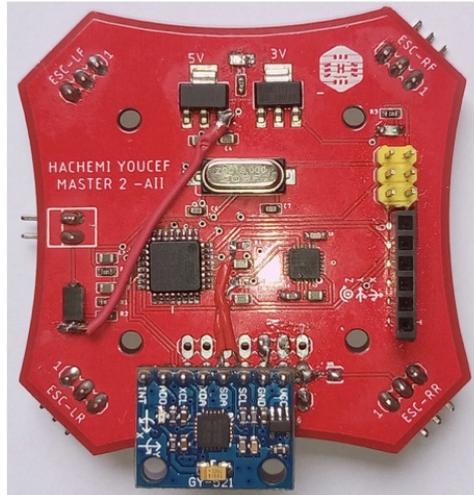


FIGURE 4.6 – Circuit final

4.3.4 Mise en marche du circuit

Après avoir apporté les corrections sur le circuit, on passe au premier essai. Mais avant ça, les nouveaux microcontrôleurs ne contiennent pas de bootloader, donc pour une première utilisation nous devons les flasher. Pour cela, on utilise une carte Arduino Uno avec le programme ISP dans exemple, pour qu'on l'utilise comme programmeur et graver le bootloader sur le microcontrôleur de notre Carte via le port ICSP en utilisant la fonction *graver la séquence d'initialisation* disponible sur Arduino IDE. Une fois que le bootloader est installé sur le microcontrôleur, il est possible de le programmer avec l'UART. Pour que notre carte puisse être programmée avec l'ordinateur, on doit disposer d'un convertisseur UART → USB, pour cela on utilise une carte Arduino, mais cette fois si sans microcontrôleur, Les schémas des figures 4.7 et 4.8 illustrent mieux les deux procédures.

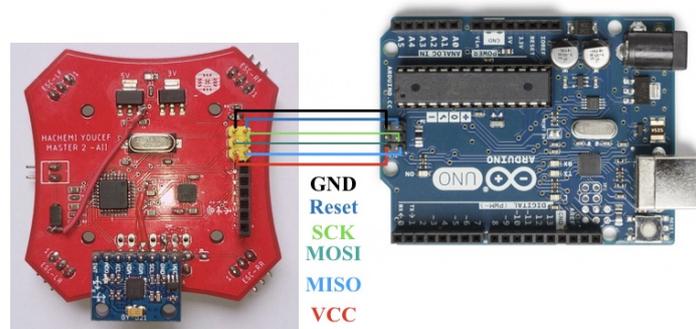


FIGURE 4.7 – Schéma brochage ICSP

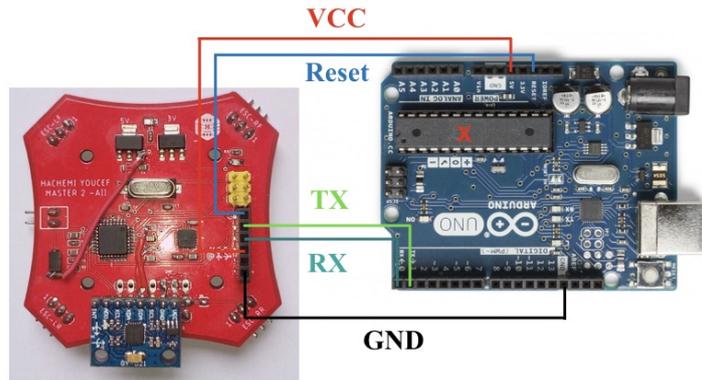


FIGURE 4.8 – Utilisation de la carte Arduino comme Convertisseur UART - USB

4.4 Programme de contrôle du quadri-rotor

Le programme final du système est constitué de plusieurs parties. Les différentes parties sont illustrées par l'organigramme du programme principale (figure 4.9).

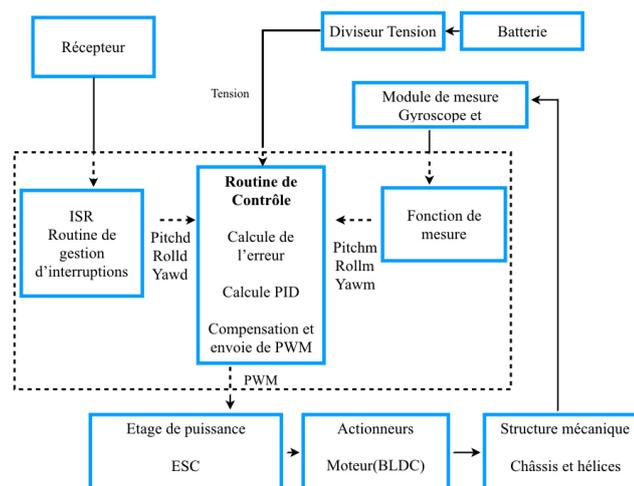


FIGURE 4.9 – Organigramme du programme principal

4.4.1 Conversion de signaux de commande en PWM

La commande est donnée par la position d'une impulsion courte (courte pulse) dans une fenêtre de temps bien définie, autrement dit modulation par position de pulse (PPM). En utilisant la routine d'interruption (Interrupt subroutine) du microcontrôleur Atmega 328 p sur les pins PB0, PB1, PB2, PB3, cela nous permet de mesurer la largeur d'impulsion du signal envoyé par la télécommande. L'interruption est générée lorsqu'il y a un front

montant sur pin PB0 par exemple, un temps (T_0) est pris avec le timer du microcontrôleur. Ensuite, lorsque le front est descendant, le temps (T_1) est pris. La largeur est calculée avec la différence ($T_1 - T_0$). La figure 4.10 explique mieux le principe.

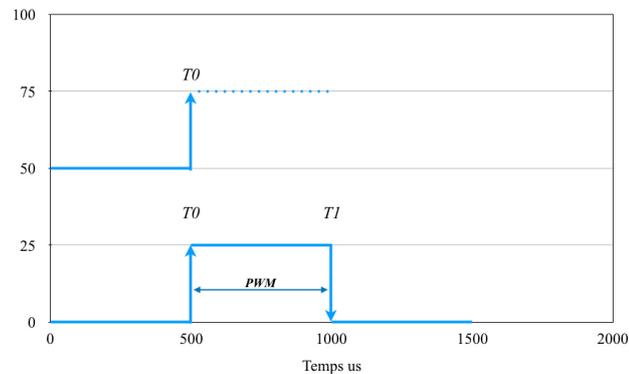


FIGURE 4.10 – Principe de calcul de la PWM

4.4.2 Compensation de décharge de la batterie

Vu qu'après un certain de temps du vol, la batterie se décharge et les forces de poussées générées par les moteurs diminuent. Alors, on doit compenser cette diminution par programme. Pour cela, on utilise un diviseur de tension afin de convertir la tension de la batterie qu'est de 11.4 v à 5 V pour qu'elle soit lue par la pin A0 [26].

4.4.3 Calcul de la commande

Comme on l'a déjà évoqué dans le deuxième chapitre, on a utilisé une régulation PID pour stabiliser et commander le Quadri-rotor. Pour cela, on doit implémenter la fonction du PID de l'équation (4.1) dans notre système en utilisant une programmation en langage C++.

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \quad (4.1)$$

4.4.4 Principe de fonctionnement du PID

Le seul but du régulateur PID c'est de maintenir les valeurs de gyroscope égaux aux valeurs reçues de la radiocommande et de calculer les pulsations de chaque moteur.

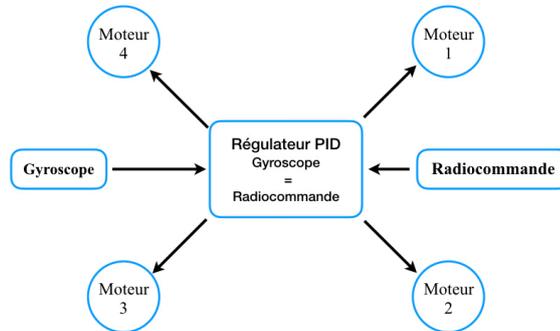


FIGURE 4.11 – Principe de fonctionnement du PID

4.4.4.1 Terme Proportionnel

Le terme P est proportionnel à la valeur actuelle de l'erreur. Par exemple, si l'erreur est grande et positive, la sortie de contrôle sera proportionnellement grande et positive, en tenant compte du facteur de gain k_p [28]. La formule du terme P utilisé dans le programme du quadri-rotor est présenté dans l'équation ci-dessous.

$$P_{sortie} = (Gyroscope - radiocommande) \times k_p \quad (4.2)$$

Avec :

Gyroscope : les valeurs mesurées du gyroscope.

radiocommande : la consigne envoyée par le pilote.

k_p : le gain p.

4.4.4.2 Terme Intégral

Le terme I prend en compte les valeurs passées de l'erreur et les intègre au fil du temps pour produire le terme I. Pour la programmation, l'intégrale est remplacée par une

somme :

$$I_{sortie} = I_{pre} + (Gyroscope - radiocommande) \times k_i \quad (4.3)$$

Avec :

I_{pre} : valeur de la sortie précédente.

k_i : gain I.

4.4.4.3 Terme Dérivé

Le terme D est la meilleure estimation de la tendance future de l'erreur, en fonction de son taux de variation actuel. Il est parfois appelé "contrôle anticipatif", car il cherche effectivement à réduire l'effet de l'erreur en exerçant une influence de contrôle générée par le taux de changement d'erreur. Plus le changement est rapide, plus l'effet de contrôle ou d'amortissement est important [29]. On traduit ce terme de dérivé par soustraction de l'erreur précédente.

$$D_{sortie} = (Gyroscope - radiocommande) - (Gyroscope_{pre} - radiocommande_{pre}) \times k_d \quad (4.4)$$

Avec :

$Gyroscope_{pre}$: Valeur précédente mesurée par le gyroscope.

$radiocommande_{pre}$: la consigne précédente envoyée par le pilote.

k_d : Le gain D.

La sortie totale du PID est donnée par l'addition des trois termes P, I et D. Pour que le quadri-rotor soit stable, on doit ajuster les paramètres des gains k_p , k_i et k_d . Car, les paramètres obtenus dans l'étude de simulation ne sont pas fiables. Mais, elles servent de valeurs initiales pour un ajustement empirique plus fin.

4.4.5 Technique d'Ajustement des paramètres du PID

Afin de déterminer les valeurs des gains du régulateur PID, on a procédé avec une méthode expérimentale. On explique les étapes suivies pour avoir un drone stable.

On commence par les paramètres du Yaw. Car, on n'a pas de problème avec la hauteur. Cette dernière est contrôlée avec l'accélération et un gain $k_p = 3$ suffit. Il faut savoir que les paramètres du Roll et Pitch sont les mêmes, donc chaque ajustement est fait au même

temps pour les deux axes. Au début par mesure de sécurité, on doit commencer avec $k_p = 1$, $k_d = 0$, $k_i = 0$, pour vérifier est-ce que la régulation est faite dans le bon sens. Ensuite, on commence avec $k_p = 0$, $k_d = 3$, $k_i = 0$, puis on augmente le k_d avec un pas de 3 jusqu'à ce que le quad copter commence à osciller. Après, on diminue le k_d à la valeur où la réponse devienne lisse. Puis, on diminue cette dernière de 25%.

Avec un $k_p = 0.2$ on a pu décoller sans aucun problème, mais il n'est pas vraiment stable. Donc, on augmente le k_p avec un pas de 0.02 jusqu'à ce qu'il commence à surcompenser (dépassement), puis on diminue le k_p de 50%.

Pour le gain de k_i , on doit procéder avec un pas minimal de 0.01, ensuite on augmente jusqu'à avoir une oscillation rapide. Puis, on diminue cette valeur de 50%.

Maintenant il reste qu'à tester et de varier les paramètres astucieusement pour avoir des valeurs optimales. Cela repose sur l'interprétation du comportement du quadri-rotor. Après plusieurs tests et bien sûr quelques crashes, on a pu avoir un drone stable avec :

$$k_p = 0.8 \quad k_d = 12.75 \quad k_i = 0.002$$



FIGURE 4.12 – Quadri-rotor réel en vol

4.5 Conclusion

Dans ce chapitre, on a présenté les détails de notre réalisation. On a eu des difficultés dans l'implémentation des différents algorithmes sur le microcontrôleur, dues principalement aux limitations de ressources. Une taille mémoire réduite et aussi une vitesse d'exécution faible nous ont obligé de minimiser et d'optimiser le programme au maximum afin qu'il puisse être embarqué. On tient à signaler que la réalisation d'un tel robot et très

dangereuse, vue la vitesse de rotation des moteurs et de type de batterie utilisées. Ce qui peut entraîner des accidents graves. On a pu avoir un quadri-rotor stable avec réponse rapide. Ce qui est l'objectif de notre projet.

Conclusion générale

Conclusion générale

Dans ce mémoire, on a présenté notre travail concernant la réalisation et le contrôle d'un Quadri rotor. Le choix et l'assemblage des composants a été fait après une étude des caractéristiques. La programmation du système a été faite en langage C++. Le code implanté sur le microcontrôleur Atmega 328 P, est généré après compilation avec Arduino IDE. La commande des actionneurs du quadri rotor a été faite à base des régulateurs PID.

Après une étude générale des drones (description des UAV, historique, domaines d'application et les configurations existantes), on s'est intéressé à la modélisation du Quadri-rotor qui fait l'objet de notre étude. L'étude des différents mouvements et la simulation nous ont permis de mieux comprendre l'aérodynamique du quadri rotor.

Le fond du travail concernant la présentation du matériel utilisé et le choix des différents composants, ainsi que les étapes suivies pour la réalisation et la mise en marche du circuit de commande sont exposés dans les chapitres 3 et 4.

On estime que les résultats obtenus sont très concluant. En fait, le quadri rotor arrive à décoller d'une manière stable et sa réponse aux commandes du pilote est assez précise.

Ce travail nous a permis de consolider les connaissances acquises durant la formation, notamment en robotique et les systèmes embarqués. La réalisation du système nous a permis d'avoir un contact avec les professionnels du domaine.

Des améliorations peuvent être apportées, notamment en utilisant des calculateurs plus performants, des capteurs pour l'évitement d'obstacles, incorporation du GPS et caméra, implantation d'algorithmes plus performants pour avoir plus d'autonomie, supervisions en temps réels sur ordinateur et smartphone, etc.

Bibliographie

Bibliographie

- [1] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, D. Nakazawa. Autonomous Flying Robots. Springer, (2010)
- [2] Young, Warren R. (1982). The Helicopters. The Epic of Flight. Chicago : Time-Life Books.
- [3] Chunhua Zhang and John M. Kovacs. The application of small unmanned aerial systems for precision agriculture : a review. Precision Agriculture, 13(6) :693 712, July 2012.
- [4] S Adams, C Friedland, and M Levitan. Unmanned Aerial Vehicle Data Acquisition for Damage Assessment in Hurricane Events. In Proceedings of the 8th International Workshop on Remote Sensing for Disaster Management, 2010.
- [5] CRASAR, Center for Robot Assisted Search and Rescue, Texas AM University, 2013. <http://crasar.org/disasters/>.
- [6] Robin R. Murphy, Eric Steimle, Chandler Grison, Charlie Cullins, Mike Hall, and Kevin Pratt. Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. Journal of Field Robotics, 25(3) :164 180, March 2008.
- [7] V Lippiello and B Siciliano. Wall inspection control of a VTOL unmanned aerial vehicle based on a stereo optical flow, 2012.
- [8] M Fumagalli, R Naldi, A Macchelli, R Carloni, S Stramigioli, and L Marconi. Modeling and control of a flying robot for contact inspection, 2012.
- [9] Luke Wallace, Arko Lucieer, Christopher Watson, and Darren Turner. Development of a UAV-LiDAR System with Application to Forest Inventory. Remote Sensing, 4(12) :1519 1543, May 2012.
- [10] Adam C. Watts, Vincent G. Ambrosia, and Everett A. Hinkley. Unmanned Aircraft Systems in Remote Sensing and Scientific Research : Classification and Considerations of Use. Remote Sensing, 4(12) :1671 1692, June 2012.

- [11] David Grémillet, William Puech, Véronique Garçon, Thierry Boulinier, Yvon Le Maho, and Others. Robots in Ecology : Welcome to the machine. Open journal of ecology, 2(2) :49 57, 2012.
- [12] Karen Anderson and Kevin J Gaston. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. Frontiers in Ecology and the Environment, March 2013.
- [13] DroneJournalisme.org, 2018. <http://www.dronejournalism.org/>.
- [14] Andrew Stobo Sniderman and Mark Hanis. Drones for Human Rights. New York Times, 2012.
- [15] Matternet Web Site, 2018. <http://matternet.us/>.
- [16] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood. Controlled Flight of a Biologically Inspired, Insect-Scale Robot. Science, 340(6132) :603 607, May 2013.
- [17] Robert C Michelson and Steven Reece. Update on apping wing micro air vehicle research- ongoing work to develop a apping wing, crawling entomopter. In 13th Bristol International RPV/UAV Systems Conference Proceedings, Bristol England, volume 30, pages 30 31, 1998.
- [18] Hala Rifai. Modélisation et commande d’un robot biomimétique volant. PhD thesis, GIPSA- lab - Grenoble Images Parole Signal Automatique, November 2008.
- [19] Hormann, Kai, and Alexander Agathos, *The point in polygon problem for arbitrary polygons*, Computational Geometry 20.3 (2001) : 131-144.
- [20] A. Tayebi, and S. Mcgilvray “Attitude stabilisation of a four rotor aerial robot”, Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, December 2004.
- [21] L. Gautier, B. Hamilton, J. Hazebrouck, and T. Tournette “Pilotage d’un quadri-rotor via un FPGA”, Tuteur ESIEE engineering, Journées pédagogiques, 2008.
- [22] L. Besnard “Control of à quadrotor vehicle using sliding mode disturbance observer”, Master Thesis, Alabama university, 2006.
- [23] K. Valavanis, George J. Vachtsevanos. Handbook of Unmanned Aerial Vehicles, Springer 2015.
- [24] Microchip Technology Incorporated 2003, Padmaraja Yedamale, “Brushless DC motor fundamentals”.

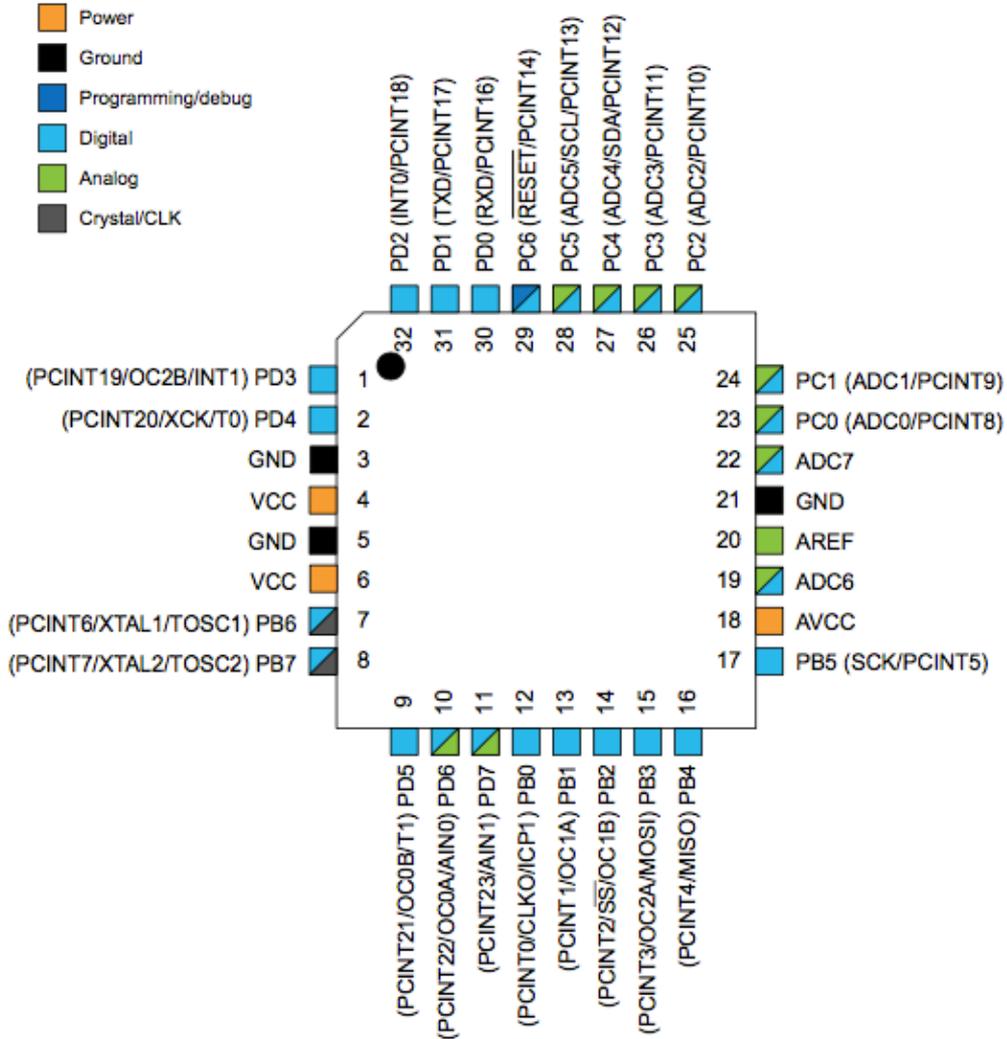
- [25] Microchip Technology Incorporated 2002, Ward Brown, "Brushless DC Motor Control Made Easy".
- [26] Cooperation, Atmel. "Atmel ATmega328P Datasheet." (2011).
- [27] InvenSense, MPU-6050 Datasheet, Nov. 2010
- [28] Fly sky FS-i6 Pdf User Manuals.
- [29] JUMO. "JUMO Campus – Control Engineering Basic principles and tips for practitioners". campus.jumo.de. Retrieved 2017-12-05.

Annexes

Annexe 1

Brochage Atmega 328 P 32-pin TQFP

Figure 5-3. 32-pin TQFP Top View



Annexe 2

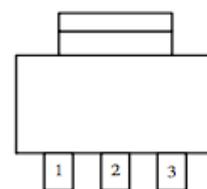
Brochage AMS 1117 - SOT223 régulateur de tension

PIN CONNECTIONS

3 PIN FIXED/ADJUSTABLE VERSION

- 1- Ground/Adjust
- 2- V_{OUT}
- 3- V_{IN}

SOT-223 Top View



Annexe 3

Programme de mesure du Gyroscope MPU6050 :

```
#include <Wire.h> // Inclut la bibliothèque Wire.h pour que nous puissions communiquer avec le gyro

//Declaration des variables
int cal_int;
unsigned long UL_timer;
double gyro_pitch, gyro_roll, gyro_yaw;
double gyro_roll_cal, gyro_pitch_cal, gyro_yaw_cal;
byte highByte, lowByte;

//Setup routine
void setup(){
  Wire.begin(); // Démarre l'I2C en tant que maître
  Serial.begin(9600); // Démarrer la connexion série à 9600bps

  // Le gyroscope est désactivé par défaut et doit être démarré
  Wire.beginTransmission(105); // Commencer la communication avec le gyro (adresse 1101001)
  Wire.write(0x20); // Nous voulons écrire pour enregistrer 20
  Wire.write(0x0F); // Définit les bits de registre comme 00001111 (Activer le gyro et activer tous les axes)
  Wire.endTransmission(); // Fin de la transmission avec le gyro
  Wire.beginTransmission(105); // Commencer la communication avec le gyro (adresse 1101001)
  Wire.write(0x23); // Nous voulons écrire pour enregistrer 23
  Wire.write(0x80); // Définit les bits de registre sur 10000000
  Wire.endTransmission(); // Fin de la transmission avec le gyro

  delay(250); // Donne le temps au gyro de commencer

  // Prenons plusieurs échantillons pour que nous puissions déterminer le décalage gyroscopique moyen

  Serial.print("Starting calibration..."); // afficher le message
  for (cal_int = 0; cal_int < 2000 ; cal_int++){ // Prend 2000 lectures pour l'étalonnage
    gyro_signalen(); // Lire la sortie gyro
    gyro_roll_cal += gyro_roll; //Ajout de la valeur de roll à gyro_roll_cal
    gyro_pitch_cal += gyro_pitch; //Ajout de la valeur de pitch à gyro_pitch_cal
    gyro_yaw_cal += gyro_yaw; //Ajout de la valeur de yaw à gyro_yaw_cal
    if(cal_int%100 == 0)Serial.print("."); //affiche un point tous les 100 lectures
    delay(4); // Attendre 4 millisecondes avant la prochaine boucle
  }
  // Maintenant que nous avons 2000 mesures, nous devons diviser par 2000 pour obtenir le décalage gyroscopique moyen

  Serial.println(" done!");
  gyro_roll_cal /= 2000;
  gyro_pitch_cal /= 2000;
  gyro_yaw_cal /= 2000;
}

//Main program
void loop(){
  delay(250); // Attendre 250 microsecondes pour chaque boucle
  gyro_signalen(); // Lire les signaux gyroscopiques
  print_output(); //Afficher les valeurs
}
```

```

void gyro_signalen(){
  Wire.beginTransmission(105);
  Wire.write(168);
  Wire.endTransmission();
  Wire.requestFrom(105, 6);
  while(Wire.available() < 6);
  lowByte = Wire.read();
  highByte = Wire.read();
  gyro_roll = ((highByte<<8)|lowByte);
  if(cal_int == 2000)gyro_roll -= gyro_roll_cal;
  lowByte = Wire.read();
  highByte = Wire.read();
  gyro_pitch = ((highByte<<8)|lowByte);
  gyro_pitch *= -1;
  if(cal_int == 2000)gyro_pitch -= gyro_pitch_cal;
  lowByte = Wire.read();
  highByte = Wire.read();
  gyro_yaw = ((highByte<<8)|lowByte);
  gyro_yaw *= -1;
  if(cal_int == 2000)gyro_yaw -= gyro_yaw_cal;
}

void print_output(){
  Serial.print("Pitch:");
  if(gyro_pitch >= 0)Serial.print("+");
  Serial.print(gyro_pitch/57.14286,0); //convertir en degre par second
  if(gyro_pitch/57.14286 - 2 > 0)Serial.print(" NoU");
  else if(gyro_pitch/57.14286 + 2 < 0)Serial.print(" NoD");
  else Serial.print(" ---");
  Serial.print(" Roll:");
  if(gyro_roll >= 0)Serial.print("+");
  Serial.print(gyro_roll/57.14286,0); //convertir en degre par second
  if(gyro_roll/57.14286 - 2 > 0)Serial.print(" Rwd");
  else if(gyro_roll/57.14286 + 2 < 0)Serial.print(" Rwu");
  else Serial.print(" ---");
  Serial.print(" Yaw:");
  if(gyro_yaw >= 0)Serial.print("+");
  Serial.print(gyro_yaw/57.14286,0); //convertir en degre par second
  if(gyro_yaw/57.14286 - 2 > 0)Serial.println(" NoR");
  else if(gyro_yaw/57.14286 + 2 < 0)Serial.println(" NoL");
  else Serial.println(" ---");
}

```

Annexe 4

Programme de mesure de la PWM reçu :

```

//Déclaration des variables
byte last_channel_1, last_channel_2, last_channel_3, last_channel_4;
int receiver_input_channel_1, receiver_input_channel_2, receiver_input_channel_3, receiver_input_channel_4;
unsigned long timer_1, timer_2, timer_3, timer_4;

void setup(){
  // Les broches Arduino (Atmega) par défaut sont des entrées, elles n'ont donc pas besoin d'être explicitement déclarées comme entrées
  PCICR |= (1 << PCIE0); // configurer PCIE0 pour activer le scan PCMSK0
  PCMSK0 |= (1 << PCINT0); // régler PCINT0 (entrée numérique 8) pour déclencher une interruption sur changement d'état
  PCMSK0 |= (1 << PCINT1); // régler PCINT0 (entrée numérique 9) pour déclencher une interruption sur changement d'état
  PCMSK0 |= (1 << PCINT2); // régler PCINT0 (entrée numérique 10) pour déclencher une interruption sur changement d'état
  PCMSK0 |= (1 << PCINT3); // régler PCINT0 (entrée numérique 11) pour déclencher une interruption sur changement d'état
  Serial.begin(9600);
}

// Boucle principale du programme
void loop(){
  delay(250);
  print_signals();
}

```

```

ISR(PCINT0_vect){
//Channel 1=====
if(last_channel_1 == 0 && PINB & B00000001 ){ // L'entrée 8 est passée de 0 à 1
  last_channel_1 = 1; // Mémorise l'état actuel de l'entrée
  timer_1 = micros(); // Définit timer_1 sur micros ()
}
else if(last_channel_1 == 1 && !(PINB & B00000001)){ // Entrée 8 modifiée de 1 à 0
  last_channel_1 = 0; // Mémorise l'état actuel de l'entrée
  receiver_input_channel_1 = micros() - timer_1; // La chaine 1 est micros () - timer_1

//Channel 2=====
if(last_channel_2 == 0 && PINB & B00000010 ){
  last_channel_2 = 1;
  timer_2 = micros();
}
else if(last_channel_2 == 1 && !(PINB & B00000010)){
  last_channel_2 = 0;
  receiver_input_channel_2 = micros() - timer_2;
}

//Channel 3=====
if(last_channel_3 == 0 && PINB & B00000100 ){
  last_channel_3 = 1;
  timer_3 = micros();
}
else if(last_channel_3 == 1 && !(PINB & B00000100)){
  last_channel_3 = 0;
  receiver_input_channel_3 = micros() - timer_3;
}

//Channel 4=====
if(last_channel_4 == 0 && PINB & B00001000 ){
  last_channel_4 = 1;
  timer_4 = micros();
}
else if(last_channel_4 == 1 && !(PINB & B00001000)){
  last_channel_4 = 0;
}

// Fonction pour afficher les signaux du récepteur
void print_signals(){
  Serial.print("Roll:");
  if(receiver_input_channel_1 - 1480 < 0)Serial.print("<<<");
  else if(receiver_input_channel_1 - 1520 > 0)Serial.print(">>>");
  else Serial.print("-+-");
  Serial.print(receiver_input_channel_1);

  Serial.print(" Nick:");
  if(receiver_input_channel_2 - 1480 < 0)Serial.print("^^^");
  else if(receiver_input_channel_2 - 1520 > 0)Serial.print("vvv");
  else Serial.print("-+-");
  Serial.print(receiver_input_channel_2);

  Serial.print(" Gas:");
  if(receiver_input_channel_3 - 1480 < 0)Serial.print("vvv");
  else if(receiver_input_channel_3 - 1520 > 0)Serial.print("^^^");
  else Serial.print("-+-");
  Serial.print(receiver_input_channel_3);

  Serial.print(" Yaw:");
  if(receiver_input_channel_4 - 1480 < 0)Serial.print("<<<");
  else if(receiver_input_channel_4 - 1520 > 0)Serial.print(">>>");
  else Serial.print("-+-");
  Serial.println(receiver_input_channel_4);
}
}

```

Annexe 5

Programme de calculé PID pour un seul axe « Roll » :

```
void calculate_pid(){
// Calcule PID pour l'axe Roll
pid_error_temp = gyro_roll_input - pid_roll_setpoint; // Calcule de l'erreur
pid_i_mem_roll += pid_i_gain_roll * pid_error_temp; // Calcule de I
if(pid_i_mem_roll > pid_max_roll)pid_i_mem_roll = pid_max_roll; // Limitation du terme I
else if(pid_i_mem_roll < pid_max_roll * -1)pid_i_mem_roll = pid_max_roll * -1;

pid_output_roll = pid_p_gain_roll * pid_error_temp + pid_i_mem_roll + pid_d_gain_roll * (pid_error_temp - pid_last_roll_d_error); // calcule de la sortie total du PID
if(pid_output_roll > pid_max_roll)pid_output_roll = pid_max_roll;// Limitation du PID
else if(pid_output_roll < pid_max_roll * -1)pid_output_roll = pid_max_roll * -1;

pid_last_roll_d_error = pid_error_temp; // mémorisé l'erreur pour la prochaine itération
```

Résumé

Dans ce travail, nous nous sommes intéressés à la réalisation et le contrôle d'un Quadri rotor. En premier lieux nous avons commencé par une étude des caractéristiques des différents composants du Quadri rotor. Ensuite, on a procédé à l'assemblage, la conception/réalisation d'un circuit électronique de commande et, enfin, la programmation et la mise en marche. Le travail a été effectué en deux phases. Durant la phase de développement, le circuit de commande a été implémenté autour d'une carte Arduino. Après les tests de validation, on a procédé à la réalisation du circuit final. Le schéma d'implantation a été conçu avec le logiciel Eagle AutoDesk et le circuit imprimé PCB a été réalisé en industrie. La programmation du contrôleur PID utilisé pour commander le système a été fait en langage C++.

Mots-clés : Drone, Quadrirotor, Arduino, PID.

Abstract

In this thesis, we have interested in realization and control of a Quadrotor. At the first step, we have studied the characteristics of the different components of the Quadrotor. Then, we studied the design and realization of an electronic control circuit, with the programming and the setting up. The work was done in two phases. During the development phase, the control circuit was implemented using an Arduino board. After the validation tests, we proceeded to the realization of the final circuit. The layout diagram was designed by Eagle AutoDesk software and the printed PCB was made in industry. The C ++ programming language is used to program the PID controller for our Quadrotor.

Keywords : Drone, Quadrotor, Arduino, PID