Contrôleur de vol de quadrirotor

DIY



**Table des matières**

[1. Calcul de l’attitude 3](#_Toc487970647)

[1.1 IMU 3](#_Toc487970648)

[1.1.1 Gyroscopes 3](#_Toc487970649)

[1.1.2 Accéléromètres 3](#_Toc487970650)

[1.2 Fusion de données: le filtre complémentaire 4](#_Toc487970651)

[2. Stabilisation mode accro (gyroscopes seuls) 5](#_Toc487970652)

[3. Stabilisation, mode “ANGLE” (gyroscopes et accéléromètres) 5](#_Toc487970653)

[1.3 Roulis tangage 6](#_Toc487970654)

[4. Stabilisation en hauteur 6](#_Toc487970655)

[1.4 Baromètre 6](#_Toc487970656)

[5. Code « CodeDroneDIY » 6](#_Toc487970657)

[1.5 Configurations en “Quad +” 6](#_Toc487970658)

[1.6 Connections 6](#_Toc487970659)

[1.7 Machine à états 7](#_Toc487970660)

[1.8 Réception CPPM 7](#_Toc487970661)

[6. Configuration matérielle 7](#_Toc487970662)

[1.9 Vue d’ensemble 7](#_Toc487970663)

[1.10 ESC 8](#_Toc487970664)

[1.11 Failsafe 8](#_Toc487970665)

[7. Annexes 8](#_Toc487970666)

[1.12 Les modes de vol 8](#_Toc487970667)

[1.13 Réglage PID 8](#_Toc487970668)

[1.14 Arduino UNO rev 3 10](#_Toc487970669)

[1.15 Génération PWM à 400Hz 10](#_Toc487970670)

[8. Bibliographie 11](#_Toc487970671)

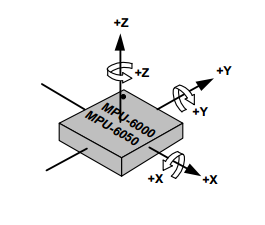
# Calcul de l’attitude

## IMU

|  |  |
| --- | --- |
| Une « Inertial Measurement Unit » est constituée d’un gyroscope 3 axes et d’un accéléromètre 3 axes. Le gyroscope détecte la vitesse angulaire, l’accéléromètre mesure l’accélération sur chaque axe.  Le gyroscope dérive dans le temps, et l’accéléromètre est sensible aux vibrations (bruité), et aux accélérations du quadrirotor. L’orientation ne peut pas être calculée à partir des données du gyroscope seul ou de l’accéléromètre seul. La solution est de fusionner les données des 2 capteurs avec un filtre complémentaire. |  |

### Gyroscopes

La position est calculée par intégration des gyroscopes.



### Accéléromètres

La position est calculée à partir de la mesure de l’accélération de la terre.

#### Roulis calculé avec l’accélération de la terre

Lorsque le quadrirotor est immobile, l’accélération mesurée est la gravité.

x

y

z

g

g

y

ay

az

z

#### Tangage calculé avec l’accélération de la terre

Lorsque le quadrirotor est immobile, l’accélération mesurée est la gravité.

x

y

z

g

g

x

ax

az

z

## Fusion de données: le filtre complémentaire

Le gyroscope dérive.

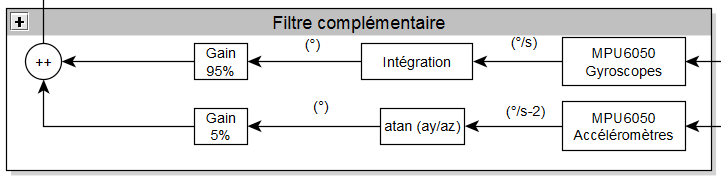
L’accéléromètre n’est pas assez rapide, il est bruité et il n’est utilisable qu’au repos lorsqu’il ne subit que l’accélération de la terre.

On applique un filtre passe-bas à l’accéléromètre et un filtre passe-haut au gyroscope.

Le filtre complémentaire permet de fusionner les données des gyroscopes et des accéléromètres.

Il consiste:

* à appliquer un filtre passe bas sur les données de l’accéléromètre car ses données sont exploitables sur la durée, il faut éliminer les variation brusques.
* à appliquer un filtre passe haut sur les données du gyroscope car ses données sont fiables sur le court terme mais prennent de l’erreur dans le temps à cause de sa dérive



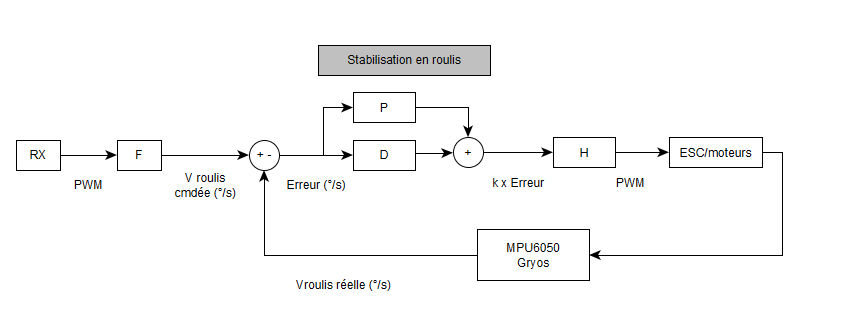
La constante de temps du filtre est un compromis entre l’élimination les accélérations dues aux mouvements du quadricoptère et la compensation de la dérive des gyroscopes :

* Trop basse, les parasites des accéléromètres ne sont pas filtrés
* Trop haute, la mesure dérive à cause des gyroscopes

|  |  |
| --- | --- |
|  |  |

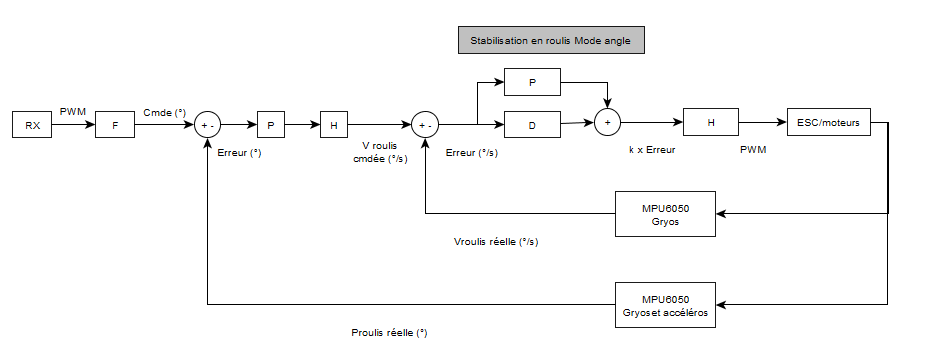
J’ai choisi une constante de temps de 5 sec, soit un coeff de 0.9995 pour un tour de boucle de 2.49ms, pour éliminer les accélérations du quadrirotor qui s’ajoutent à l’accélération de la terre.

# Stabilisation mode accro (gyroscopes seuls)



# Stabilisation, mode “ANGLE” (gyroscopes et accéléromètres)

## Roulis tangage



# Stabilisation en hauteur

## Baromètre

|  |  |  |
| --- | --- | --- |
| Flight controller | Barometer | Relative accuracy (hPa) |
| Julien | BMP180 | ±0.12 |
| Naze32 acro | MS5611 | ±1.5 |
| Naze32 full | BMP280 | ±0.12 |
| Omnibus F4 Pro | BMP280 | ±0.12 |

# Code « CodeDroneDIY »

## Configurations en “Quad +”

|  |  |
| --- | --- |
| **Quad +** | **Quad X** |
| x  y  ESC2 PORTB2  ESC3 PORTB3  ESC1 PORTB1  ESC0 PORTB0 | x  y  ESC2 PORTB2  ESC3 PORTB3  ESC1 PORTB1  ESC0 PORTB0  +  +  roll >0  pitch >0  roll <0  pitch <0  roll < 0  pitch > 0  roll > 0  pitch < 0 |

## Connections

PB0 ESC0

PB1 ESC1

PB2 ESC2

PB3 ESC3

PD2 receiver

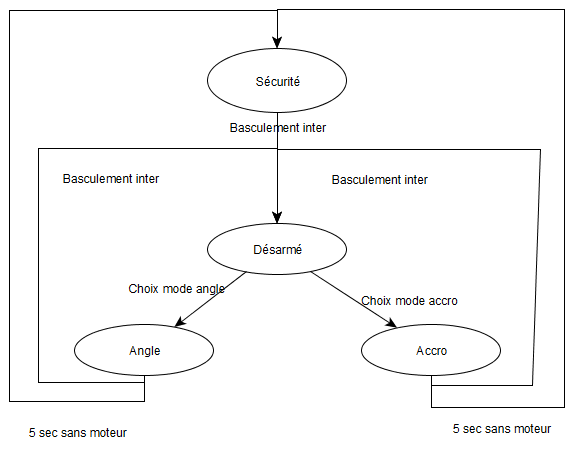
PC2 potentiometer

PC4 SDA MPU6050

PC5 SCL MPU6050

## Machine à états

|  |  |  |
| --- | --- | --- |
| **Etats** | **Evènements** | **Transitions** |
| * Désarmé * Angle * Accro | * Désarmement * Choix du mode * 5 sec sans gaz | * Désarmé vers angle * Désarmé vers accro * Angle vers désarmé * Accro vers désarmé |



## Réception CPPM

# Configuration matérielle

250 mm : trop nerveux : pour voler vite et bas, entre les obstacles. Faible autonomie.

Grand châssis pour privilégier la stabilité : 450mm

## Vue d’ensemble

|  |  |
| --- | --- |
| **ESC** | Afro 20A |
| **Moteurs** | Multistar 2216-800Kv |
| **Hélices** | 10x4.5 SF Props 2pc CW 2 pc CCW Rotation (Orange) |
| **Batterie** | Zippy Flightmax 3000mAh 4S |
| **Récepteur** | OrangeRx R617XL CPPM DSM2/DSMX 6 ch |
| **Contrôleur** | Arduino UNO rev3 |
| **IMU** | MPU6050 |
| **Chassis** | Diatone Q450 Quad 450 V3 |

## ESC

Simonk ESC : 500Hz, 1060us à 1860us de largeur d’impulsion

ESC classique : 490Hz, 1 à 2ms de largeur d’impulsion

## Failsafe

Pour la sécurité, définir le « failsafe » pour couper les gaz.

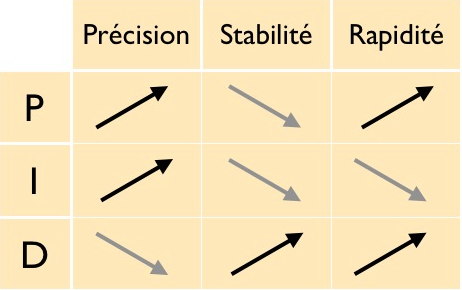
Pour programmer le « failsafe », mettre les commandes de la télécommande dans la configuration souhaitée lors de la perte de réception radio, et « binder » la télécommande. La configuration utilisée pendant le « bind » défini le « failsafe. »

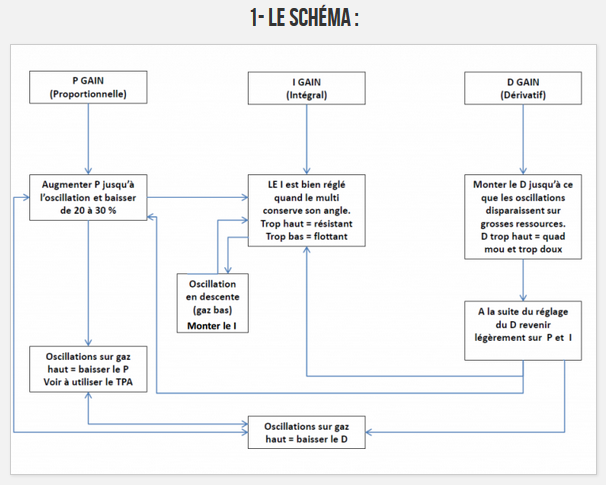
# Annexes

## Les modes de vol

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mode** | Gyro | Accé | Baro | Bouss | GPS | Rqes |
| ACRO/Gyroscope uniquement | X |  |  |  |  | Un mode généralement par défaut et son vol plus « acrobatique » (le quadrirotor ne peut faire de la mise à niveau automatique) |
| ANGLE (Stable/Niveau/Accéléromètre) | X | X |  |  |  | Mode stable ; va essayer de maintenir le niveau du modèle par rapport au sol (mais pas à une position fixe). |
| HORIZON | X |  |  |  |  | Combine l’effet stable avec des commandes et des acrobaties RC lentes et avec des commandes RC rapides. |
| BARO (Maintien de l’altitude) | X | X | X |  |  | Le baromètre est utilisé afin de conserver une certaine hauteur (fixée) lorsqu’aucune autre commande n’est reçue. |
| MAG (Tenue du cap) | X | X |  | X |  | Mode verrouillage de cap (direction à la boussole), pour essayer de maintenir son orientation en lacet. |
| HEADFREE (CareFree/ orientation indépendante du déplacement) | X | X |  | X |  | Maintient l’orientation (lacet) du quadrirotor et se déplace toujours dans la même direction 2D pour le même mouvement du manche en ROULIS/TANGAGE. |
| GPS/ Retour à la base |  | X |  | X | X | Utilise automatiquement une boussole et un GPS pour rentrer à la base, au point de départ GPS. |
| GPS/ Points de passage |  | X |  | X | X | Suit automatiquement les points de cheminement GPS pré-configurés de manière autonome. |
| GPS/ Maintien de position |  | X |  | X | X | Maintient la position actuelle en utilisant le GPS et le baromètre (si disponible). |

## Réglage PID





Le P

C’est le **P** qui va résoudre les problèmes de **vibrations**. Le I quant à lui, joue sur l’inertie de la machine et sur sa réactivité.

Montez le P jusqu’à obtenir une machine qui vibre / oscille. Baissez le I également, ça aide.

Descendez le P peu à peu jusqu’à ce que les vibrations disparaissent totalement, même à fond de gaz !

Le I

Il agît sur la **dérive** du multi est est lié au P. En acro, voltige ou FPV, il sera plus bas que pour une machine dédiée à la vidéo. Dans la vidéo en Français, vous apprendrez comment le régler à l’aide d’un truc connu : placer un poids sur un bras

Montez le I jusqu’à obtenir des oscillations faibles en montée et ou en descente. Vous devriez en avoir aussi à fond de gaz.

Si vous avez des oscillations en descente : montez le I

Si vous avez des oscillations en montée : baissez le I

Trouvez une valeur qui vous débarrasse des deux

Si des vibrations réapparaissent, c’est normal. Retouchez le **P**. Baissez le légèrement, c’est selon.

Le D

Ce paramètre est le plus  » personnel  » des trois. Il influence la réactivité de la machine.

Faites de grands mouvements de gauche à droite ou d’avant en arrière pour observer les réactions de la machine. Appréciez et réglez selon vos préférences.

Des vibrations peuvent revenir : corrigez le **P**.

Le Yaw

Si votre machine continue à dériver après un ordre ou si elle dérive seule sur l’axe du lacet, changez la valeur. Elle est très souvent sur **8.5** et est correcte ainsi.

tpa breakpoint

Ce paramètre joue sur le ratio des PID. En effet, la tension et le niveau de gaz sont des variables qui agissent sur le comportement. Le TPA va faire varier vos PID selon ces facteurs.Si vous n’en mettez pas, il se peut que vous ayez des vibrations lorsque vous êtes à fond de gaz avec une lipo chargée à bloc. Pour être précis, les TPA ( Throtlle PID Attenuation ) jouent sur le P. ( Merci **XKin Ai** pour la précision )

## Arduino UNO rev 3

Microcontrôleur ATmega328

Architecture 8 bits RISC

16Mhz => T = 0.0625us

1MIPS par MHz

Dimensions : 68.6 mm x 53.4 mm

Poids : 25 g

## Génération PWM à 400Hz

1. Classer les ESC par ordre croissant de largeur d’impulsion.
2. Utiliser le « Timer1 ».
3. Mettre toutes les sorties au niveau haut en une seule fois à l’aide de la commande « PORTB=0b00001111 ».
4. Mettre à au niveau bas les sortie au fur et à mesure.

|  |
| --- |
| \* For every 2.5 ms period the first 1.5 is unused and can be used for main program logic.  \* From 1.5 to 2.5 ms every interrupt will disturb timing, but without communications most servo controllers are uninteresting.   \* We can use a loop to wait for the next "set servo low" time or we can use a timer interrupt  \* Best precision is if all the servo output pins are on the same AVR port and we directly write to the PORTx register. Then servos with equal timing can be handled with one port write.  \* The situation with the most difficult timing is when two servos are separated by 2-8 uS, less than an interrupt period.  \* After every 2.5 ms servo period we wait 7\*2.5 ms for the next servo control period. We can use these periods to address  other groups of servos if we use demuxes to distribute the servo pulses to the groups. |

# Bibliographie

#### Arduino

[file:///C:/Program%20Files/Arduino/reference/www.arduino.cc/en/Reference/Libraries.html](file:///C:\Program%20Files\Arduino\reference\www.arduino.cc\en\Reference\Libraries.html)

[file:///C:/Program%20Files/Arduino/reference/www.arduino.cc/en/Reference/HomePage.html](file:///C:\Program%20Files\Arduino\reference\www.arduino.cc\en\Reference\HomePage.html)

#### PWM

<https://librepilot.atlassian.net/wiki/display/LPDOC/PWM,+PWMSync,+Oneshot+Output>

<http://forum.arduino.cc/index.php?topic=46487.0>

#### PPM

<http://frskytaranis.forumactif.org/t4426-tuto-pwm-cppm-ccpm-ppm-s-bus-s-port-kesako>

#### PID

<http://www.fpv-passion.fr/docteur-pid/>

#### Quadrirotor

<https://www.mondrone.net/fabriquer-quadricoptere-la-propulsion/>

#### Traitement de données

<https://ericjformanteaching.wordpress.com/2013/10/08/smoothing-sensor-input/>

#### Fusion de données

<http://www.mouser.fr/applications/sensor_solutions_mems/>

<http://www.pieter-jan.com/node/11>

#### Programme python

Andy BAKER

<http://pythonfiddle.com/andy-baker-quadcopter/>

#### Vibrations

<http://ardupilot.org/plane/docs/common-vibration-damping.html>