

Manuel de configuration de la DroFly v3	29/05/2014
version 1	page 1/11

Manuel de configuration de la carte DroFly v3

Table des matières

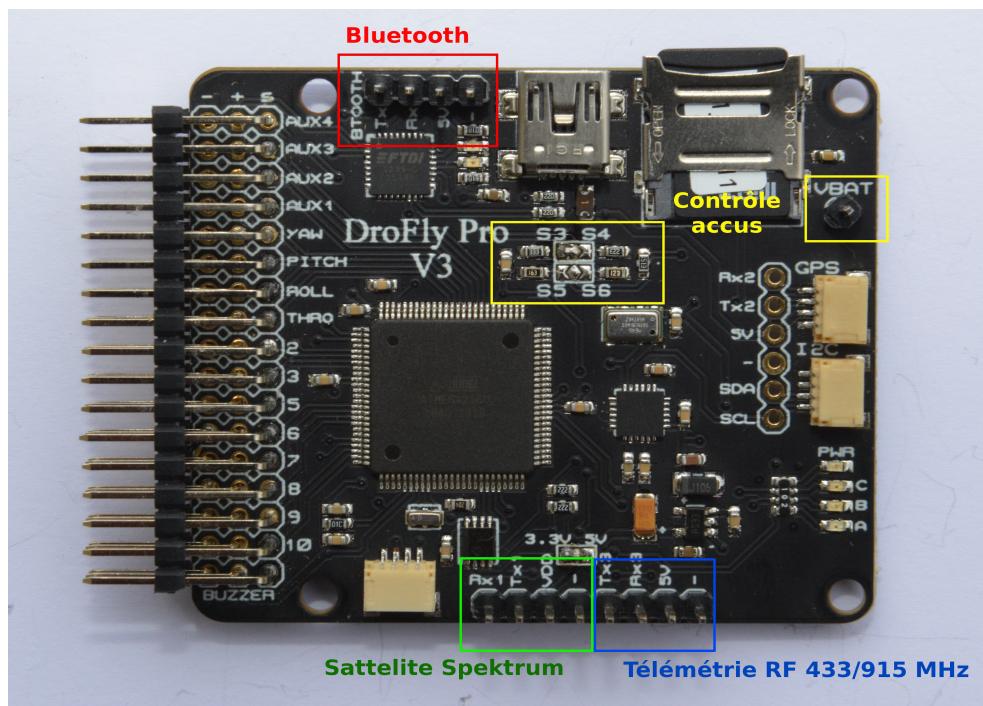
1) Composants.....	2
2) Connecteurs du contrôleur de vol.....	2
3) Logiciels.....	5
4) Formatage de la carte microSD.....	6
5) Initialisation du GPS.....	8
6) Configuration logicielle du contrôleur de vol.....	9

1) Composants

Afin de pouvoir utiliser la carte DroFly v3, vous devez disposer des éléments suivants :

- 1 contrôleur de vol DroFly V3
- 1 carte GPS NEO-6M Ublox avec boussole intégré
- 1 boîtier de protection pour la carte GPS
- 2 câbles droits JST 4-pin
- 1 barrette de broches simples mâles (au minimum 15 broches)
- 1 barrette de broches triples mâles (au minimum 16 broches)
- 1 carte microSD

2) Connecteurs du contrôleur de vol

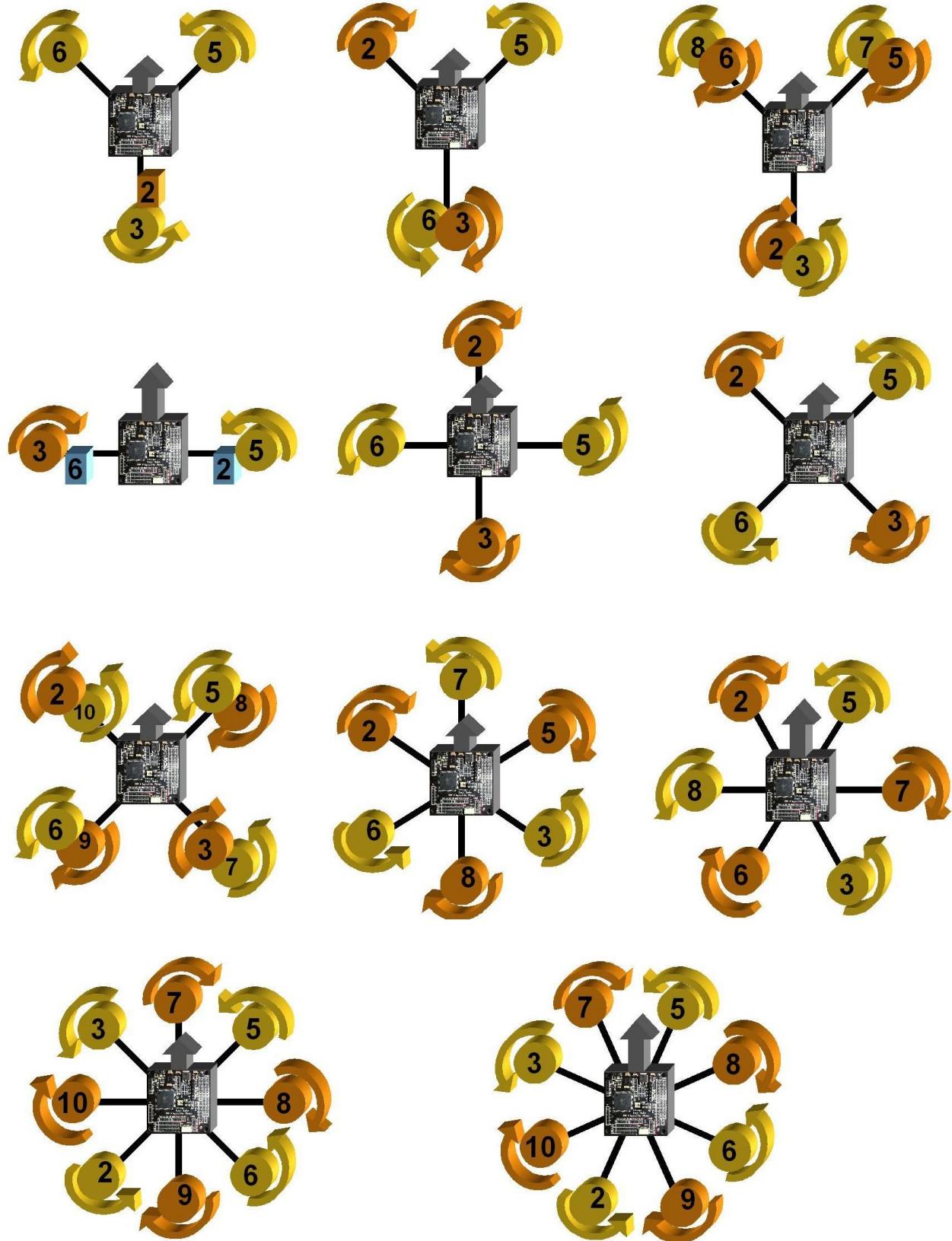


La carte DroFly dispose de 3 dispositif E/S SÉRIE :

- SERIAL 1 (RX1, TX1) pour les Satellite Spektrum
- SERIAL 2 (RX2,TX2) pour le connecteur BlueTooth
- SERIAL 3 (RX3,TX3) pour de la télémestrie RF433/915 Mhz

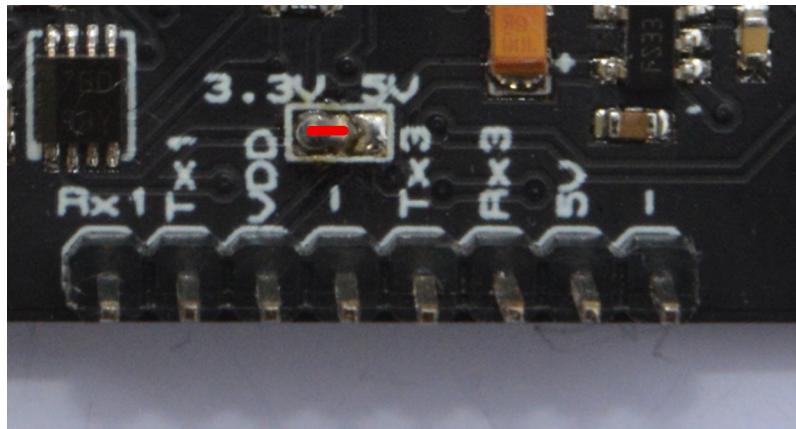
La carte DroFly dispose de 17 broches PWM qui peuvent être reliées aux ESC (broches 2, 3, 4, 5, 6, 7, 8, 9, 10) ou au récepteur de la télécommande (broches THRO, ROLL, PITCH, YAW) ou à des canaux auxiliaires si la télécommande 2.4 GHz le permet (broches AUX1, AUX2, AUX3, AUX4),

Les schémas ci-dessous indiquent sur quelles broches brancher les ESC suivant le type multicoptère

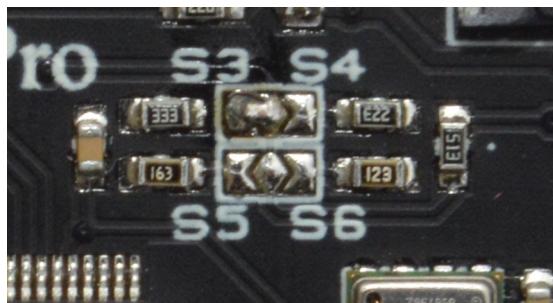


Manuel de configuration de la DroFly v3	29/05/2014
version 1	page 4/11

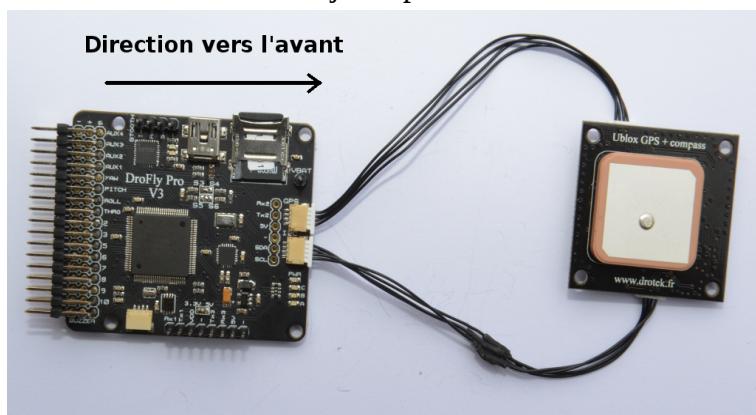
Pour réduire l'encombrement du récepteur 2.4 GHz, on peut utiliser des récepteurs Sattelite (par exemple de Spektrum). Or ces récepteurs doivent être alimentés en 3.3 volt. Il faut donc souder un point de tension sur la carte qui se trouve juste à côté de SERIAL 1.



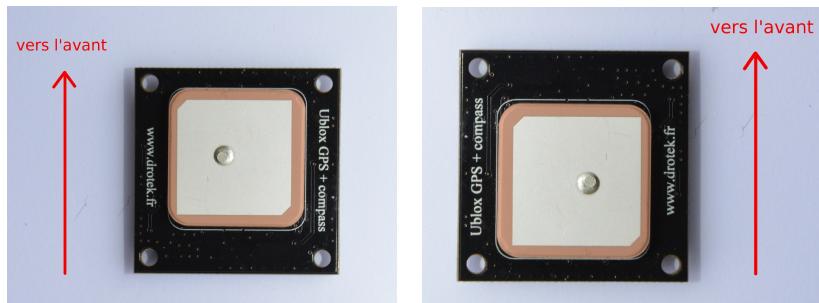
La carte DroFly permet de contrôler le niveau des batteries via des ponts diviseurs de tension. Il est nécessaire de souder le pont correspondant à votre tension d'accumulateur (S3 pour les batteries 3S, S4 pour les 4S, S5 pour les 5S et S6 pour les 6S). Il faut ensuite souder une broche sur VBAT et la relier à la batterie.



La carte DroFly est reliée à une carte déportée intégrant un récepteur GPS et un magnétomètre par deux câbles droit JST 4-pin via des connecteurs « GPS » et « I2C ». Sur le châssis du multicoptère, les connecteurs « GPS » et « I2C » sont toujours placés vers l'avant.



La carte GPS peut être orientées de différentes manières. Il faudra adapter la configuration logicielle au mode d'orientation choisi. Les libellés sur la carte permettent de se repérer.



La carte DroFly permet de sauvegarder l'itinéraire de vol sur une carte microSD. Il faudra aussi utiliser un sketch particulier pour formater la carte microSD.

La carte GPS a aussi besoin d'être initialisée avant son utilisation. Il est donc conseillé de le faire avant d'installer la carte dans le châssis.

3) Logiciels

La DroFly utilisera comme logiciel le code Multiwii 2.3 auquel s'ajoute le support des cartes SD et l'architecture de la nouvelle carte écrit par Olivier Fernandez (wareck@cegetel.net)¹ et le support des waypoints par Eosbandi. Tout ce travail a été effectué par Olivier Fernandez.

Le code source doit être compilé avec le compilateur Arduino dans sa version 1.0.5.

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

Le code source pour la DroFly est disponible via GitHub :

https://github.com/Wadudu/MultiWii_Wad

Le code comprend trois jeux de sources :

- le code pour le formatage de la carte microSD
- le code pour l'initialisation de la carte GPS
- le code pour le contrôleur de vol

Le téléchargement des binaires se fait via le port USB. Il est recommandé de ne pas brancher le connecteur BlueTooth pour faire la mise à jour logicielle du contrôleur de vol.

¹ Consulter son site sur <http://ferntronix.fr/fr/multi-copters/drofly/la-carte-droflypro-v3-de-drotek>

4) Formatage de la carte microSD

Le formatage de la carte microSD se fait via du code à compiler et téléverser sur la carte DroFly. Les sources sont disponibles dans le répertoire Sdcard_ToolBox. Deux sketchs sont disponibles :

- SDcard_formater pour formater la carte microSD
- SDcard_info pour lire le contenu de la carte :

Pour compiler le code, il faut supprimer le dossier "Robot_Control" dans les librairies Arduino. La librairie "SDfat" étant commune au logiciel Multiwii et à Robot_Control, Arduino peut faire un mixage des versions de librairie et générer ces codes erreur. Vous devez quitter complètement Arduino, supprimer le dossier, et relancer Arduino.

Après la compilation, il faut redémarrer la carte et se connecter un émulateur de terminal (HyperTerminal sous Windows ou minicom sous Linux). La vitesse du port série est 9600 bauds. Il vaut mieux faire cette opération avant d'avoir installé la carte sur le châssis.

```
eric@terminus: ~/dev/multiwii/MultiWii_Wad/src
SDcard Formater v1.1
ce programme permet d'effacer et/ou formater les cartes SD/SDHC.

La commande Effacement permet un effacement rapide.
la carte sera remplie de 0x00
Les cartes de plus de 2 GB seront formatee en FAT32.
Les cartes de plus petite taille seront formatees en FAT16.

!!Attention toutes les donnees sur la carte seront perdues!!
Tappez 'O' pour continuer: █
```

Taper « O ». Trois options sont disponibles : « E » ou « F » ou « Q »

```
eric@terminus: ~/dev/multiwii/MultiWii_Wad/src
Les options sont:
E - Effacement rapide de la carte sans formatage.
F - Formatage de la carte et effacement. (recommande)
Q - Formatage rapide sans effacement.

Entrez une option: F
Card Size: 3724 MB, (MB = 1,048,576 bytes)

Effacement
.....
Les donnees sont supprimees, remplacees par des : 0x00
Fin de l'effacement

Formatage
Blocks/Cluster: 64
FAT32
.....
Fin du formatage
█
```

Pour vérifier le formatage, il faut compiler et téléverser SDcard_info.

```
eric@terminus: ~/dev/multiwii/MultiWii_Wad/src

SDCard info V1.1
SdFat library version: 20131225

Appuyez sur une touche :

Initialisation de la carte réussie.

Taille de la carte: 3905 MB (MB = 1,000,000 bytes)

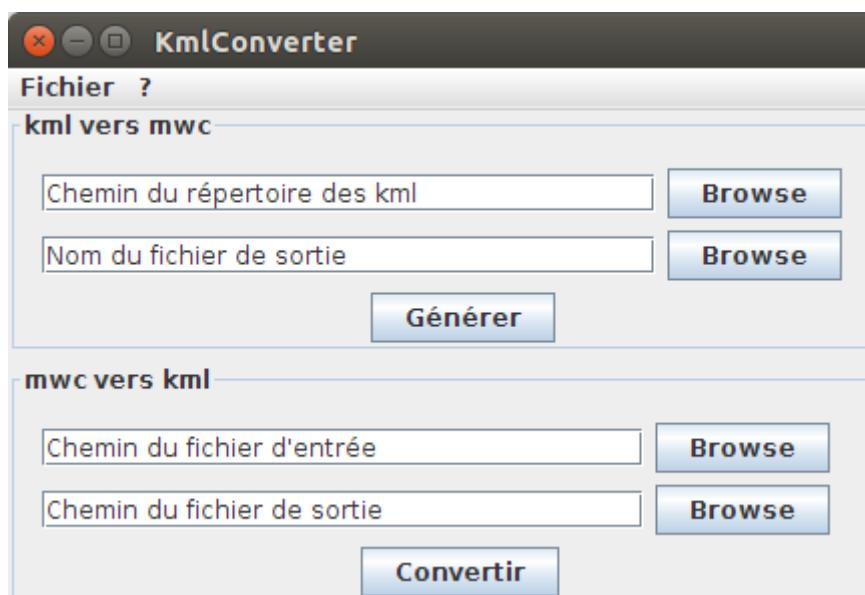
Le formatage est en FAT32, la taille du cluster est de 32768 bytes

fichiers présents: (nom date heure taille):

Fin du processus!
Appuyez sur une touche pour recommencer.
```

La carte microSD permet de sauvegarder les coordonnées GPS du vol. L'application KmlConverter convertit le fichier généré par le logiciel de la DroFly en un fichier importable dans GoogleEarth. L'application est une application JAVA qui peut être lancée par la commande :

```
java -jar kmlconverter.jar
```

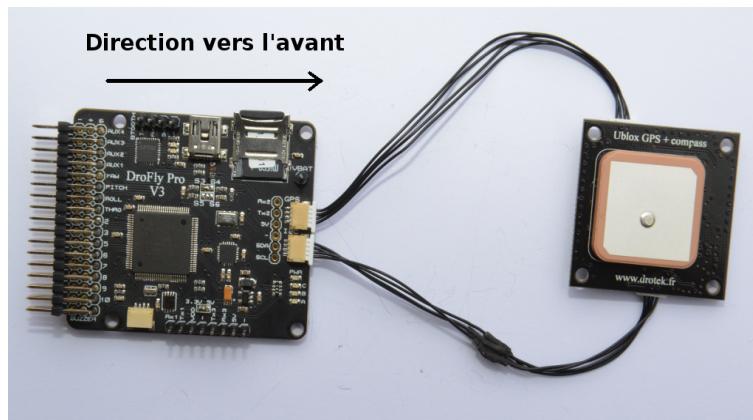


Manuel de configuration de la DroFly v3	29/05/2014
version 1	page 8/11

5) Initialisation du GPS

L'initialisation de la carte GPS se fait via le sketch GPS_autoconfig en chargeant le fichier GPS_autoconfig.ino.

Comme pour le formatage de la carte micoSD, il vaut mieux faire cette opération avant d'installer le contrôleur de vol sur le châssis. Avant tout téléchargement sur la carte DroFly, la carte GPS doit être reliée au contrôleur de vol par des câbles droit JST dans la configuration suivante :



Une fois la compilation et le téléchargement effectué, il faut débrancher et rebrancher le câble USB pour lancer l'initialisation. L'opération peut être vérifiée avec un émulateur de terminal avec un débit de 115200 bauds.

```
eric@terminus: ~/dev/multiwii/MultiWii_Wad/src
Configuring u-Blox GPS initial state...
Setting Navigation Mode... Success!
ACK Received! B5 62 05 01 02 00 06 24 32 5B
Setting Data Update Rate... Success!
ACK Received! B5 62 05 01 02 00 06 08 16 3F
Deactivating NMEA GLL Messages Success!
ACK Received! B5 62 05 01 02 00 06 01 0F 38
Deactivating NMEA GSA Messages Success!2
ACK Received! B5 62 05 01 02 00 06 01 0F 38
Deactivating NMEA GSV Messages Success!7
ACK Received! B5 62 05 01 02 00 06 01 0F 38
Deactivating NMEA RMC Messages Success!3
ACK Received! B5 62 05 01 02 00 06 01 0F 38
Deactivating NMEA VTG Messages Success!
ACK Received! B5 62 05 01 02 00 06 01 0F 38
Setting Port Baud Rate... Success!
```

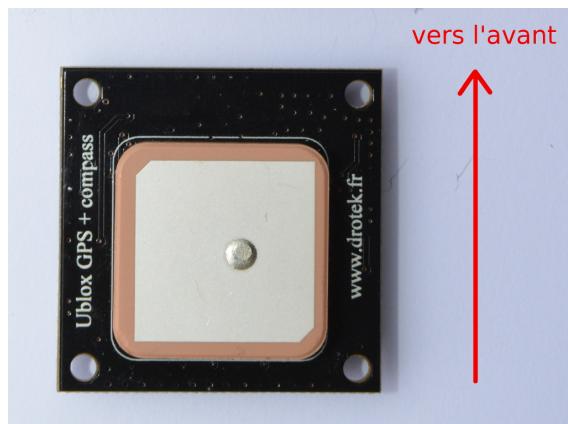
6) Configuration logicielle du contrôleur de vol

Le mode d'orientation du boîtier GPS impose une modification dans le fichier def. Les changements se font aux lignes suivantes :

```
#if defined (DROTEK_DROFLY_V3_GPS)
#define HMC5883
#define MAG_ORIENTATION(X, Y, Z) {imu.magADC[ROLL] = Y;
imu.magADC[PITCH] = X; imu.magADC[YAW] = Z;}
#endif
```

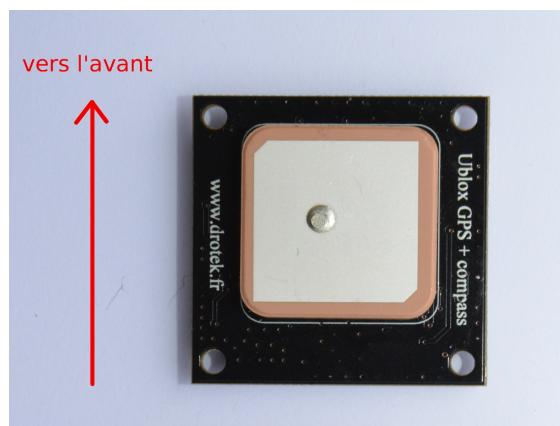
Si le GPS est positionné selon l'image ci-dessous, on doit avoir sur une même ligne :

```
#define MAG_ORIENTATION(X, Y, Z) {imu.magADC[ROLL] = Y;
imu.magADC[PITCH] = X; imu.magADC[YAW] = Z;}
```



Si le GPS est positionné selon l'image ci-dessous, on doit avoir sur une même ligne :

```
#define MAG_ORIENTATION(X, Y, Z) {imu.magADC[ROLL] = -Y;
imu.magADC[PITCH] = -X; imu.magADC[YAW] = Z;}
```



Manuel de configuration de la DroFly v3	29/05/2014
version 1	page 10/11

Dans le fichier config.h, les définitions suivantes doivent être décommentées pour pouvoir utiliser le GPS :

```
#define DROTEK_DROFLY_V3_GPS
#define GPS_SERIAL 2
#define GPS_BAUD    57600
#define UBLOX
```

En début de fichier, il faut définir le type de chassis. Ici, un quadrioptère a été configuré :

```
//#define GIMBAL
//#define BI
//#define TRI
//#define QUADP
#define QUADX
//#define Y4
//#define Y6
//#define HEX6
//#define HEX6X
//#define HEX6H // New Model
//#define OCTOX8
//#define OCTOFLATP
//#define OCTOFLATX
//#define FLYING_WING
//#define VTAIL4
//#define AIRPLANE
//#define SINGLECOPTER
//#define DUALCOPTER
//#define HELI_120_CCPM
//#define HELI_90_DEG
```

La sauvegarde des fichiers de log sur la carte microSD se fait en décommentant les définitions suivantes :

```
#define LOG_PERMANENT 4096
#define MWI_SDCARD
#define LOG_PERMANENT_SD_ONLY
#define LOG_GPS_POSITION 2
#define CSPIN 53
```

Manuel de configuration de la DroFly v3	29/05/2014
version 1	page 11/11

L'activation logicielle du contrôle de batterie se fait dans config.h. Pour une batterie 3S, il faut activer les définitions suivantes :

```
#define VBAT
#define VBATSCALE      131
#define VBATNOMINAL    126 // 12,6V full battery nominal voltage
#define VBATLEVEL_WARN1 107 // (*) (***) 10,7V
#define VBATLEVEL_WARN2 99 // (*) (***) 9.9V
#define VBATLEVEL_CRIT  93 // (*) (***) 9.3V - critical condition
#define NO_VBAT         16 // Avoid beeping without any battery
```

Le buzzer peut s'avérer nécessaire pour disposer d'une alarme sonore :

```
#define BUZZER
```