



École Polytechnique de l'Université de Tours

64, Avenue Jean Portalis
37200 TOURS, France
Tél. +33 (0)2 47 36 14 14

www.polytech.univ-tours.fr

Département Informatique industrielle

Guide de l'administrateur			
Projet :	Système de stabilisation motorisé pour caméra		
Réalisé par :	Sébastien LAPORTE	Encadré par :	Ameur SOUKHAL
Début du projet :	22/09/2015	Fin du projet :	10/02/2015
Historique des modifications			
Version	Date	Note de version	
1.0	30/01/2016	Version de dépôt initial	

Table des matières

1.	Introduction.....	3
2.	Mise en place des outils	3
3.	Paramètres du programme du microcontrôleur	3
3.1.	Coefficients PID pour chaque axe (Yaw, Pitch, Roll).....	3
3.2.	Activation / désactivation de chaque axe.....	4
3.3.	Fréquence des signaux PWM pour la commande des moteurs	4
3.4.	Fréquence de la boucle d'asservissement PID	4
3.5.	Fréquence de rafraichissement des données brutes du capteur	4
3.6.	Fréquence de la boucle de l'algorithme de Madgwick	4
4.	Réglage des paramètres par liaison série.....	5
5.	Etalonnage du capteur	7
6.	Tests	8
7.	Table des figures	9
8.	Annexes.....	9

1. Introduction

Ce document est destiné à l'administrateur du système de stabilisation motorisé pour caméra réalisé dans le cadre d'un projet de fin d'étude à Polytech Tours. Ce guide décrit les démarches à suivre pour la configuration du système.

Les configurations présentées dans ce document doivent être effectuées par l'administrateur pour préparer le système après sa fabrication. Ce paramétrage est à effectuer une seule fois dans la vie du système.

2. Mise en place des outils

Pour agir sur les différents paramètres du système, l'administrateur doit mettre en place l'environnement de développement constitué de l'IDE Arduino et du logiciel Teensy. Le programme développé n'utilise pas de bibliothèque externes à l'environnement Arduino.

- Page de téléchargement de l'IDE Arduino : <https://www.arduino.cc/en/Main/Software>
- Page de téléchargement du logiciel Teensy : https://www.pjrc.com/teensy/td_download.html

3. Paramètres du programme du microcontrôleur

Pour effectuer les réglages du système, l'administrateur a la possibilité d'agir sur un ensemble de paramètres. Il est possible d'afficher des valeurs dans un terminal via une liaison série, désactiver un ou plusieurs axes, agir sur les fréquences d'exécution des fonctions du système ou encore de modifier directement les valeurs liées au PID de chaque axe.

Les paramètres suivants permettent à l'administrateur d'afficher certaines informations dans un terminal série.

```
#define DEBUG          0      /* Affiche les informations d'initialisation */
#define PRINT_RAW_VALUES 0      /* Affiche les données brutes du capteur */
#define PRINT_YPR_VALUES 0      /* Affiche l'orientation de la caméra en YPR */
#define PRINT_PID_COMPUTING 0    /* Affiche les données de calcul du PID */
#define AXE_TO_PRINT     0      /* Affiche la consigne calculée d'un axe */
#define LOCK_YAW         0      /* Verrouillage de l'axe Yaw : Moteur immobile */
```

Pour effectuer ces opérations, il est nécessaire de connecter le microcontrôleur à un ordinateur et d'ouvrir le programme Arduino. L'ensemble des réglages du programme s'effectue dans le fichier nommé « parametres.h » :

3.1. Coefficients PID pour chaque axe (Yaw, Pitch, Roll)

Ces paramètres sont propres à la structure mécanique. Ils sont également liés aux fréquences de la boucle d'asservissement et à la fréquence de rafraichissement des données brutes du capteur. Ces coefficients peuvent être ajustés directement dans le programme ou peuvent être mis à jour depuis une application grâce au port série.

```
#define KP_Y 0.001 /* kp Yaw */
#define KI_Y 0.003 /* ki Yaw */
#define KD_Y 1.0   /* kd Yaw */
#define KP_P 0.8   /* kp Pitch */
#define KI_P 0.008 /* ki Pitch */
#define KD_P 1000.0 /* kd Pitch */
#define KP_R 0.001 /* kp Roll */
#define KI_R 0.005 /* ki Roll */
#define KD_R 1.0   /* kd Roll */
```

3.2. Activation / désactivation de chaque axe

Afin de faciliter le réglage des coefficients PID il est possible de désactiver chaque axe du système. Lorsqu'un axe est désactivé cela a pour conséquence de ne pas commander le moteur correspondant.

```
#define ENABLE_YAW      1    /* 0 : désactivation, 1 : activation */
#define ENABLE_PITCH    1    /* 0 : désactivation, 1 : activation */
#define ENABLE_ROLL     1    /* 0 : désactivation, 1 : activation */
```

3.3. Fréquence des signaux PWM pour la commande des moteurs

La modification de cette fréquence agit sur les perturbations sonores dégagées par les moteurs. En effet, si la fréquence de fonctionnement n'est pas assez importante, un sifflement se produit lors de la commande. Celle-ci est également en lien avec la consommation énergétique du système. Il est nécessaire de garder la fréquence la plus basse possible pour limiter la consommation. Lors de tests, nous nous sommes aperçu qu'une fréquence de 18 kHz est un bon compromis.

```
#define PWM_FREQUENCY    18000    /* Hz */
```

3.4. Fréquence de la boucle d'asservissement PID

La fréquence de cette boucle agit sur le temps de réaction du système. La modification de cette fréquence a également pour conséquence de modifier la validité des paramètres des PID.

```
#define PID_FREQUENCY    50000    /* Hz */
```

3.5. Fréquence de rafraichissement des données brutes du capteur

La fréquence de rafraichissement des données du capteur est fixée à 200 Hz par la configuration de celui-ci. L'augmentation de cette fréquence n'aura pas pour effet d'augmenter le nombre d'échantillons. En revanche, il est possible de baisser cette fréquence pour limiter le nombre d'échantillons.

```
#define UPDATE_CPT_FREQUENCY    200    /* Hz */
```

3.6. Fréquence de la boucle de l'algorithme de Madgwick

La fréquence de la boucle de l'algorithme de Madgwick est liée avec la fréquence de rafraichissement des données brutes du capteur. En effet, cet algorithme est itératif et doit avoir une fréquence 5 fois supérieure à celle du rafraichissement des données brutes du capteur pour pouvoir obtenir une estimation correcte de l'orientation.

```
#define MADGWICK_FREQUENCY    1000    /* Hz */
```

4. Réglage des paramètres par liaison série

Pour faciliter le réglage des coefficient PID nous avons mis en place une interface en nous appuyant sur un projet existant. Ce programme Processing permet d'agir sur les paramètres PID ainsi que d'activer ou désactiver chaque axe. Une deuxième fenêtre permet de tracer les données liées à l'asservissement.



Figure 1 : Interface de réglage

Le programme Processing proposé peut être éventuellement remplacé par une autre solution. La mise à jour des paramètres est effectuée en transmettant l'identifiant du paramètre et sa valeur séparée par un espace tel que : « identifiant valeur ». Le tableau ci-dessous identifiant les paramètres.

Identifiant	Fonction	
en_roll_axis	0 : Désactivation de l'axe Roll	1 : Activation de l'axe Roll
en_pitch_axis	0 : Désactivation de l'axe Pitch	1 : Activation de l'axe Pitch
en_yaw_axis	0 : Désactivation de l'axe Yaw	1 : Activation de l'axe Yaw
Kp_Roll	Paramètre KP du PID de l'axe Roll	
Ki_Roll	Paramètre KI du PID de l'axe Roll	

Kd_Roll	Paramètre KD du PID de l'axe Roll
Kp_Pitch	Paramètre KP du PID de l'axe Pitch
Ki_Pitch	Paramètre KI du PID de l'axe Pitch
Kd_Pitch	Paramètre KD du PID de l'axe Pitch
Kp_Yaw	Paramètre KP du PID de l'axe Yaw
Ki_Yaw	Paramètre KI du PID de l'axe Yaw
Kd_Yaw	Paramètre KD du PID de l'axe Yaw
freq_cpt	Fréquence de la boucle d'acquisition des données du capteur en Hz
freq_PID	Fréquence de la boucle d'asservissement en Hz
freq_madgwick	Fréquence de la boucle de l'algorithme de Madgwick en Hz

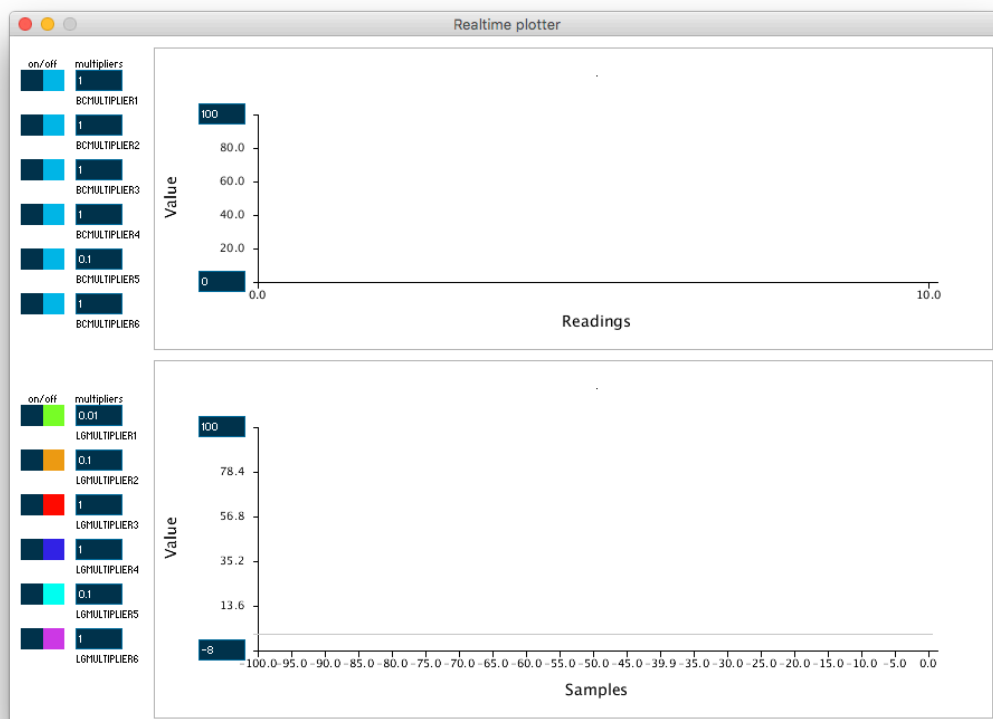


Figure 2 : Interface de tracée des valeurs d'asservissement

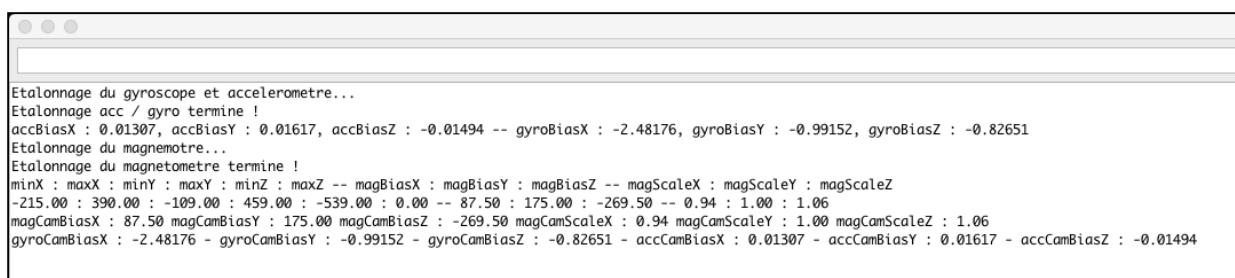
L'illustration ci-dessous montre l'interface permettant d'afficher les valeurs liées à l'asservissement. Chaque axe peut donc y être représenté pour assister l'administrateur dans le réglage de l'application.

5. Etalonnage du capteur

Au premier démarrage du système, il est nécessaire d'effectuer la calibration du capteur MPU-9250. Une routine automatique permet d'effectuer cette calibration de façon automatique. Pour cela, il est nécessaire de connecter le microcontrôleur à un port série pour pouvoir visualiser les instructions. La vitesse de transmission est fixée à 115200 Bauds/s. La calibration s'effectue en deux phase :

- Dans un premier temps, le système va effectuer la calibration de l'accéléromètre et du gyroscope. Le système doit être stable et le capteur positionné parfaitement à l'horizontal. Une fois cette étape terminée, le système affiche les valeurs de calibration dans le terminal à titre indicatif.
- Pour terminer, le système va procéder à l'étalonnage du magnétomètre. Un message informe l'administrateur qu'il est nécessaire de réaliser des mouvements en huit avec le capteur. Pour terminer la calibration, il suffit d'immobiliser le capteur. Cette action permet de mettre fin à la procédure et d'enregistrer les valeurs de calibrations.

L'illustration ci-dessous est un exemple d'étalonnage effectué.



```
Etalonnage du gyroscope et accelerometre...
Etalonnage acc / gyro termine !
accBiasX : 0.01307, accBiasY : 0.01617, accBiasZ : -0.01494 -- gyroBiasX : -2.48176, gyroBiasY : -0.99152, gyroBiasZ : -0.82651
Etalonnage du magnetometre...
Etalonnage du magnetometre termine !
minX : maxX : minY : maxY : minZ : maxZ -- magBiasX : magBiasY : magBiasZ -- magScaleX : magScaleY : magScaleZ
-215.00 : 390.00 : -109.00 : 459.00 : -539.00 : 0.00 -- 87.50 : 175.00 : -269.50 -- 0.94 : 1.00 : 1.06
magCamBiasX : 87.50 magCamBiasY : 175.00 magCamBiasZ : -269.50 magCamScaleX : 0.94 magCamScaleY : 1.00 magCamScaleZ : 1.06
gyroCamBiasX : -2.48176 - gyroCamBiasY : -0.99152 - gyroCamBiasZ : -0.82651 - accCamBiasX : 0.01307 - accCamBiasY : 0.01617 - accCamBiasZ : -0.01494
```

Figure 3 : Exemple étalonnage capteur

6. Tests

Pour vérifier la cohérence des données brutes du capteur il est possible d'afficher les valeurs dans le terminal.

- **Accéléromètre** : Lorsque le capteur est immobile, les valeurs doivent tendre vers 0 pour l'axe X et Y. Sur l'axe Z, la valeur doit tendre vers 1000 mG à cause de l'accélération de la pesanteur.
- **Gyroscope** : Lorsque le capteur est immobile, les valeurs doivent tendre vers 0 sur les trois axes.
- **Magnétomètre** : Les valeurs changent en fonction de la position géographique et en fonction des perturbations électromagnétiques.

7. Table des figures

Figure 1 : Interface de réglage.....	5
Figure 2 : Interface de tracée des valeurs d'asservissement.....	6
Figure 3 : Exemple étalonnage capteur.....	7

8. Annexes

Lien Github du code source du programme : https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Version_finale

Lien Github du programme Processing : <https://github.com/bibi03331/PFE-Polytech/tree/master/Developpement/Livrables/Console>