Fréquence d’échantillonnage du gyroscope

Présentation

TODO …

Expérience

Objectif

On souhaite mesurer la fréquence effective d’échantillonnage de la centrale inertielle, afin de vérifier qu’elle correspond à celle choisie dans la librairie Arduino. Les valeurs possibles de ces fréquences sont 25, 50,100 et 200Hz – à noter que les valeurs sont apparemment bruitées *(voir commentaires librairie Arduino)* pour 200Hz.

Protocole

La librairie I2Cdev-MPU5060 indique la valeur à donner et facilite l’envoi des bytes des différents registres du mpu afin de modifier la fréquence d’envoi de données du buffer FIFO du capteur, permettant de choisir la fréquence d’échantillonnage du capteur.

On choisit la fréquence d’échantillonnage en modifiant la valeur du byte du registre correspondant via une requête I2C (en modifiant le code de la librairie, changement pris en compte dans l’initialisation du capteur dans le programme arduino).

On mesure le temps entre deux acquisitions capteurs et on le récupère via le moniteur série Arduino.

On calcule la moyenne sur 5 mesures capteurs.

Résultats

Fréquence théo (Hz)	Fréquence exp (Hz)	Mesure période 1	Mesure période 2	Mesure période 3	Mesure période 4	Mesure période 5
20	20.5	48816	48836	48844	48848	48844

20	20,5	48816	48836	48844	48848	48844
22,2	22,8	43940	43924	43952	43552	43916
25	25,7	38864	38872	38872	38860	38892

Fréquence théo (Hz)	Fréquence exp (Hz)	Mesure période 1	Mesure période 2	Mesure période 3	Mesure période 4	Mesure période 5
28,6	29,5	33860	33884	33864	33872	33888
33,3	34,6	28868	28896	28856	28896	28876
40	41,9	23872	23900	23868	23896	23880
50	53	18876	18896	18872	18896	18888
66,7	72	13880	13884	13896	13880	13896
100	111,8	8948	8944	8968	8924	8944
200	273,3	3660	3660	3656	3660	3660

On observe que les fréquences expérimentales du FIFO sont plus élevées que les valeurs effectives données par les développeurs de la librairie.

Il est donc vraisemblable qu'il y ait un problème de code.

Identification de l'erreur

En suivant le protocole, on a réalisé que le code source de mon objet gyro était incorrect : le booléen newDataReady, représentant le déclenchement de l'évènement associé à la réception d'une nouvelle trame de données provenant du gyroscope, n'était jamais réinitialisé à la valeur false.

Du point de vue du code global, l'objet gyro contenait a priori constamment de nouvelles valeurs pour les paramètres dynamiques du système, ce qui n'était absolument pas le cas (cette valeur s'actualise à la fréquence choisie pour le fifo du gyroscope).

La conséquence était que le nombre de calculs effectués était beaucoup trop important sans que cela ne soit utile. On a donc ajouté une méthode setNoDataReady() permettant de réinitialiser la valeur du booléen.

Conclusion

L’expérience m’a permis de prendre conscience d’une erreur de codage de l’objet gyro. J’ai donc décidé de réaliser une seconde expérience similaire pour valider la correction de l’erreur, et ainsi valider la bonne compréhension du protocole de communication et de sa gestion par les composants embarqués.

Afin de pouvoir détecter plus facilement une erreur similaire, et d’avoir plus de précision sur les mesures renvoyées par le capteur, on a décidé d’horodater chaque mesure du gyroscope afin de mieux décrire les acquisitions, plutôt que d’horodater les trames reçues par le programme utilisateur.

Afin d’optimiser la boucle d’asservissement, on souhaite maximiser la fréquence de lecture de données capteur. Il s’agit maintenant de vérifier ou non si les données sont suffisamment fiables, dans le cas où f=200Hz et f=100Hz.