

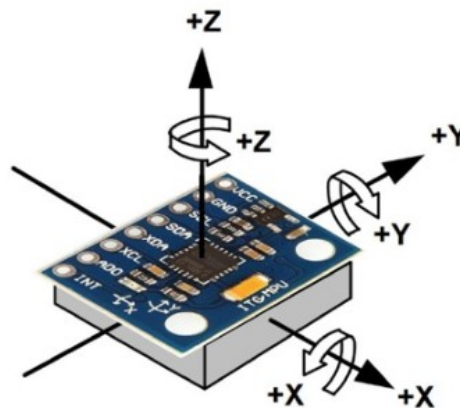
Analyse préalable Classe MPU6050 pour le Rpi

Table des matières

Analyse préalable Classe MPU6050 pour le Rpi.....	1
1. Présentation générale.....	1
2. Câblage.....	2
3. Calibration.....	3
3.1. Correction des erreurs de biais initiales.....	3
3.2. Calibration du capteur.....	3
4. Filtrage passe bas numérique.....	4
5. Détection de la chute libre (Free fall).....	5
6. Gestion des interruptions.....	6
7. Les interruptions d'événements de l'accéléromètre.....	6

1. Présentation générale

Le MPU6050 est un accéléromètre 3 axes ET un gyroscope 3 axes associé avec un processeur de mouvement numérique (DMP)



- tension d'alimentation est comprise entre 2.375V et 3.45V
- utilise le bus I2C
- utilise les interruptions
- Le gyroscope a une plage allant jusqu'à $\pm 2000^\circ/\text{sec}$
- Le gyroscope a une consommation nominale de 3.6mA
- L'accéléromètre a une plage allant jusqu'à $\pm 16g$
- L'accéléromètre a une consommation nominale de 500 μ A

Le processeur de mouvement numérique intégré (DMP)

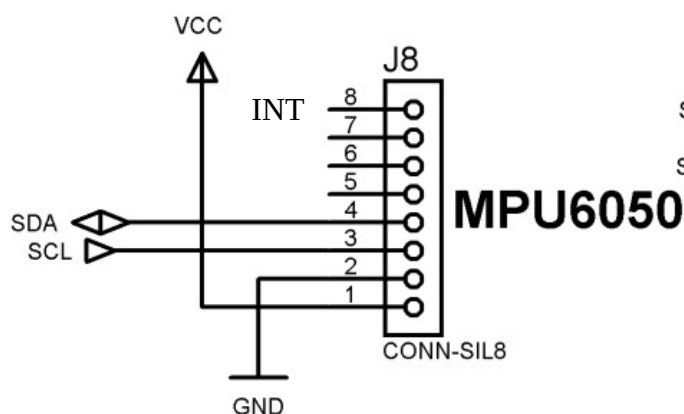
Le but du DMP est de décharger à la fois les exigences de synchronisation et la puissance de traitement du rpi. Typiquement, les algorithmes de traitement de mouvement doivent être exécutés à

une fréquence élevée, souvent autour de 200 Hz, afin de fournir des résultats précis avec une faible latence. Cela est nécessaire même si l'application se met à jour à une fréquence beaucoup plus basse par exemple, une interface utilisateur peut se mettre à jour aussi lentement que 5 Hz, mais le traitement de mouvement doit toujours s'exécuter à 200 Hz. Le DMP peut être utilisé comme un outil pour minimiser la consommation d'énergie, simplifier la synchronisation, simplifier l'architecture logicielle et économiser des MIPS précieux sur le rpi.

- Le moteur interne Digital Motion Processing™ (DMP™) prend en charge les algorithmes de traitement de mouvement 3D et de reconnaissance des mouvements.
- Le MPU-60X0 collecte les données du gyroscope et de l'accéléromètre tout en synchronisant l'échantillonnage des données à un taux défini par l'utilisateur. L'ensemble complet de données obtenu par le MPU-60X0 comprend des données de gyroscope 3 axes, des données d'accéléromètre 3 axes et des données de température. La sortie calculée par le MPU pour le processeur système peut également inclure des données d'orientation provenant d'un magnétomètre numérique 3 axes tiers.
- Le FIFO met en mémoire tampon l'ensemble complet des données, réduisant ainsi les exigences de synchronisation sur le processeur système en permettant à ce dernier de lire en rafale les données du FIFO. Après avoir lu en rafale les données du FIFO, le processeur système peut exécuter une autre tâche pendant que le MPU collecte davantage de données.
- L'interruption programmable prend en charge des fonctionnalités telles que la reconnaissance d'absence ou présence de mouvement ou la chute libre.
- Filtres passe-bas programmables numériquement.

2. Câblage

Le signal d'interruption du MPU6050 est disponible sur la broche **INT** (8) du module sortie push-pull. A relier à une broche d'entrée numérique du rpi **GPIO 26**.



3. Calibration

Le MPU6050 nécessite une calibration individuelle car il n'est pas possible de le calibrer en usine. En effet, cette calibration consiste à ajuster les décalages (offsets) spécifiques à chaque MPU6050. Ainsi, si vous possédez plusieurs de ces capteurs, vous devrez les calibrer chacun séparément.

3.1. Correction des erreurs de biais initiales

Les registres d'offset permettent de corriger les biais initiaux des mesures d'accélération et de rotation. Ces biais peuvent être dus à des imperfections de fabrication, des variations de température, ou des changements environnementaux. En ajustant les valeurs d'offset, on peut obtenir des lectures plus précises et fiables.

3.2. Calibration du capteur

Lors de la calibration, les valeurs de sortie du capteur sont comparées à des valeurs de référence connues. Les différences observées sont utilisées pour calculer les valeurs d'offset qui sont ensuite stockées dans les registres correspondants. Cette étape est essentielle pour assurer que le capteur fournit des mesures précises et cohérentes.

Pour définir les biais de l'accéléromètre dans les registres matériels appropriés, il est nécessaire de prendre en compte les valeurs de trim d'usine présentes dans ces registres, qui doivent être ajoutées aux biais calculés de l'accéléromètre. Au démarrage, ces registres contiennent des valeurs non nulles. Par ailleurs, le bit 0 de l'octet de poids faible doit être conservé, car il est utilisé pour les calculs de compensation de température. Les registres de biais de l'accéléromètre attendent une entrée de biais de 2 048 LSB par g, ce qui implique que les biais de l'accéléromètre calculés précédemment doivent être divisés par 8.

<https://www.firediy.fr/article/calibrer-le-capteur-mpu6050-avec-un-arduino-drone-ch-5>

4. Filtrage passe bas numérique

Le filtre passe-bas numérique intégré dans le MPU6050 peut être configuré avec différentes bandes passantes. Le prix à payer pour ce filtrage est le léger retard du signal numérique par rapport au signal analogique.

le réglage du filtre passe-bas numérique (DLPF) pour les gyroscopes et les accéléromètres est configurable dans le registre de CONFIG adresse 0x1A

L'accéléromètre et le gyroscope sont filtrés selon la valeur de DLPF_CFG comme indiqué dans le tableau ci-dessous.

DLPF_CFG (Fs = 1 kHz)		
	Bande passante (Hz)	Délai (ms)
0	260	0
1	184	2.0
2	94	3.0
3	44	4.9
4	21	8.5
5	10	13.8
6	5	19.0

Note : Fs = Fréquence d'échantillonnage (sampling frequency).

<https://www.i2cdevlib.com/devices/mpu6050#registers>

5. Détection de la chute libre (Free fall)

Un des problèmes était que les documents ne mentionnent pas le détecteur de chute et d'autres registres basés sur le mouvement dans le fichier de description des registres ni dans la spécification du produit. Mais j'ai trouvé une autre version en ligne avec un lien qui mentionne tous les registres.

La chute libre est détectée en vérifiant si les mesures de l'accéléromètre sur les trois axes ont une valeur absolue inférieure à un seuil programmable par l'utilisateur (seuil d'accélération). Pour chaque échantillon où cette condition est vraie (échantillon qualifiant), un compteur est incrémenté. Pour chaque échantillon où cette condition est fausse (échantillon non qualifiant), le compteur est décrémenté. Une fois que le compteur atteint un seuil programmable par l'utilisateur (seuil du compteur), une interruption de chute libre est déclenchée et un indicateur est activé. L'indicateur est effacé une fois que le compteur a décrémenté jusqu'à zéro. Le compteur ne s'incrémente pas au-delà du seuil du compteur et ne se décrémente pas en dessous de zéro.

L'utilisateur dispose de plusieurs paramètres de configuration pour affiner la détection de chute libre. Tant le seuil d'accélération que le seuil du compteur sont configurables par l'utilisateur. Le registre FF_THR permet à l'utilisateur de définir un seuil. Le registre FF_DUR permet à l'utilisateur de définir la durée en incréments de 1 ms.

Le MPU-60X0 offre aussi une capacité de détection de mouvement avec une fonctionnalité similaire à celle de la détection de chute libre. **Les mesures de l'accéléromètre sont passées à travers un filtre passe-haut numérique configurable (DHPF) afin d'éliminer le biais dû à la gravité.** Un échantillon de mouvement qualifiant est un échantillon dont la valeur absolue après filtrage haute fréquence de tout axe dépasse un seuil programmable par l'utilisateur. Un compteur s'incrémente pour chaque échantillon qualifiant et se décrémente pour chaque échantillon non qualifiant. Une fois que le compteur atteint un seuil programmable par l'utilisateur, une interruption de mouvement est déclenchée. L'axe et la polarité ayant déclenché l'interruption sont signalés dans le registre MOT_DETECT_STATUS.

<code>#define FF_THR</code>	<code>0x1D</code>
<code>#define FF_DUR</code>	<code>0x1E</code>
<code>#define MOT_THR</code>	<code>0x1F</code>
<code>#define MOT_DUR</code>	<code>0x20</code>
<code>#define ZRMOT_THR</code>	<code>0x21</code>
<code>#define ZRMOT_DUR</code>	<code>0x22</code>

Le registre FF_THR configure le seuil de détection pour la détection d'un événement de chute libre. L'unité de FF_THR . Une chute libre est détectée lorsque la valeur absolue des mesures de

l'accéléromètre pour les trois axes est inférieure au seuil de détection. Cette condition incrémente le compteur de durée de chute libre (registre 30). L'interruption de chute libre est déclenchée lorsque le compteur de durée de chute libre atteint la durée spécifiée dans FF_DUR.

6. Gestion des interruptions

Le MPU-60X0 dispose d'un système d'interruption programmable qui peut générer un signal d'interruption sur la broche **INT**. Les indicateurs de statut signalent la source de l'interruption. Les sources d'interruption peuvent être activées et désactivées individuellement.

La fonctionnalité d'interruption est configurée via le registre de configuration des interruptions. Les éléments configurables incluent la configuration de la broche INT, la méthode de verrouillage et de nettoyage de l'interruption, et les déclencheurs pour l'interruption. Les éléments pouvant déclencher une interruption sont :

1. Le générateur d'horloge verrouillé sur un nouvel oscillateur de référence (utilisé lors du changement de sources d'horloge) ;
2. De nouvelles données disponibles à la lecture (à partir du FIFO et des registres de données) ;
3. **Les interruptions d'événements de l'accéléromètre chute libre, absence de mvt;**
4. Le MPU-60X0 n'a pas reçu d'accusé de réception d'un capteur auxiliaire sur le bus I2C secondaire.

L'état des interruptions peut être lu à partir du registre de statut des interruptions.

Pour plus d'informations concernant les interruptions, veuillez vous référer au document "MPU-60X0 Register Map and Register Descriptions". Pour des informations concernant les interruptions d'événements de l'accéléromètre du MPU-60X0, veuillez vous référer à la section 8.

7. Les interruptions d'événements de l'accéléromètre

Pour programmer le MPU6050 afin d'obtenir un signal d'interruption lorsqu'une chute libre est détectée, il faut configurer plusieurs registres du capteur pour activer et définir les conditions d'interruption. Voici les étapes générales à suivre :

1. **Initialisation du MPU6050** : Configurer le capteur et initialiser la communication I2C.

```
void MPU6050_init() {  
    I2C_Write(MPU6050_ADDRESS, PWR_MGMT_1, 0x00); // Sortir du mode veille  
    I2C_Write(MPU6050_ADDRESS, SMPLRT_DIV, 0x07); // Régler le taux  
    d'échantillonnage à 1 kHz / (1 + 7) = 125 Hz  
    I2C_Write(MPU6050_ADDRESS, CONFIG, 0x00); // Configure le filtre passe-bas  
    I2C_Write(MPU6050_ADDRESS, GYRO_CONFIG, 0x18); // Sensibilité du gyroscope à  
    ±2000 °/s
```

```
I2C_Write(MPU6050_ADDRESS, ACCEL_CONFIG, 0x01); // Sensibilité de  
l'accéléromètre à ±2g }
```

2. Configuration des registres d'interruption :

- Activer l'interruption de détection de mouvement libre.
- Définir les seuils et les durées pour la détection de la chute libre.

```
void MPU6050_configure_free_fall_interrupt() {  
I2C_Write(MPU6050_ADDRESS, INT_ENABLE, 0x00); // Désactiver toutes les interruptions  
d'abord  
I2C_Write(MPU6050_ADDRESS, MOT_THR, 0x14); // Régler le seuil de détection de  
mouvement (21*LSB) -> environ 0.84g  
I2C_Write(MPU6050_ADDRESS, MOT_DUR, 0x01); // Régler la durée minimale de détection de  
mouvement (1*LSB) -> 1 ms  
I2C_Write(MPU6050_ADDRESS, INT_ENABLE, 0x40); // Activer l'interruption de détection de  
mouvement libre  
}
```

3. Gestion des interruptions : Programmer le RPI pour gérer l'interruption générée par le MPU6050.

Le registre 0x38 (Interrupt Enable) possède deux bits séparés pour activer/désactiver les interruptions de détection de chute et de détection de mouvement sur la broche INT du capteur.

Par conséquent, le code suivant activera l'interruption de détection de mouvement sur la broche INT.

```
imu.writebyte(INT_PIN_CONFIG, 0x20);  
imu.writebyte(INT_ENABLE, 0x40);  
imu.writebyte(MOT_THR, 0x01);  
imu.writebyte(MOT_DUR, 0x01);
```

La détection de mouvement fonctionne pour le moment, la prochaine étape consiste à vérifier la détection de chute et son exactitude.

quel est le seuil de détection de chute et quelle est sa plage ? Le document indique qu'un seuil dans une plage de 1 mg peut être défini en utilisant le registre FF_THS. J'ai donc pensé qu'en le réglant à la valeur 0xFF, cela correspondrait à environ 250 mg, n'est-ce pas ?

Cependant, régler à 0xFF était trop sensible. Si je touchais simplement le capteur, une interruption était également déclenchée.

La valeur que l'on pourrait vouloir définir pour la détection pourrait être autour de 0xD0, etc. ? Pour mon cas, cela semble fonctionner, mais je dois encore faire quelques expériences.

