**TD Etude de la structure matérielle associée à la fonction**

**« Acquisition et mesure d’accélération linéaire et de vitesse angulaire »**

**CORRECTION**

**Objectif**: étudier la structure matérielle associée à la fonction « Acquisition et mesure d’accélération linéaire et de vitesse angulaire », afin d’être en mesure, ultérieurement, d’élaborer le programme permettant d’acquérir, de mesurer puis de transmettre à la passerelle, les informations d’accélération linéaire et de vitesse angulaire auxquelles la carte End Device est soumise.

**Consignes** : avant de répondre aux questions de ce TD, lire ***attentivement***, ***au moins deux fois*** dans leur intégralité, et en s’aidant du logiciel de traduction en ligne DeepL si nécessaire :

* la documentation du composant **LSM6DS3** (<https://www.st.com/resource/en/datasheet/lsm6ds3.pdf>) ;
* la note d’application AN4650 du composant **LSM6DS3** (<https://www.st.com/content/ccc/resource/technical/document/application_note/12/98/b4/44/a5/bf/4e/c5/DM00157511.pdf/files/DM00157511.pdf/jcr:content/translations/en.DM00157511.pdf>).

1. **Localisation de la fonction objet de l’étude**



1. **Schéma structurel associé**

|  |
| --- |
|  |

1. **Présentation fonctionnelle du composant LSM6DS3**
   1. Préciser, parmi les propositions qui suivent, à quel type de composant le LSM6DS3 appartient : capteur, microcontrôleur, actionneur.

**Le composant LSM6DS3 est un capteur.**

* 1. Préciser les deux paramètres physiques auxquels le composant est sensible (indiquer le symbole et l’unité de la grandeur associée à chacun de ces paramètres physiques et préciser la correspondance de ces unités avec celles du Système International). Indiquer quel est le troisième paramètre physique –très secondaire- auquel le composant est sensible.

**Il est sensible au paramètre accélération linéaire d’une part, et au paramètre vitesse angulaire d’autre part (page 1/102), et ce, selon les 3 axes x, y et z pour chacun des deux paramètres.**

**Il permet donc de mesurer :**

* **l’accélération linéaire a, exprimée en g (9,81 m.s-2) selon les 3 axes (ax, ay et az) ;**
* **la vitesse angulaire ω, exprimée en dps (degré par seconde) selon les 3 axes ωx, ωy et ωz.**

**Dans le système international, ces grandeurs s’expriment respectivement en m.s-2 et en rad.s-1.**

**On a donc :**

* **1 g = 9,81 m.s-2 (ou 1 m.s-2 = 0,102 g) ;**
* **1 dps = 0,017 rad.s-1 (ou 1 rad.s-1 = 57,32 dps).**

**Le composant intègre par ailleurs un capteur de température (page 1/102, rubrique « Features »).**

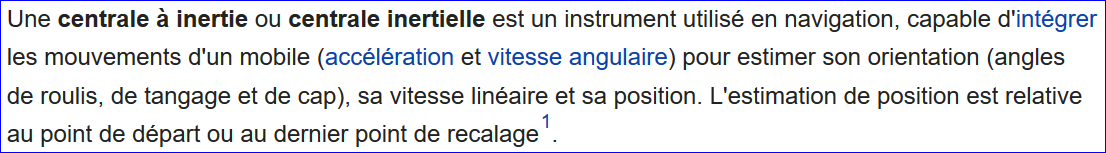
* 1. Préciser la différence entre un gyroscope et un gyromètre, et justifier la raison pour laquelle on peut affirmer que, contrairement à ce que dit la documentation constructeur, le composant LSM6DS3 ne constitue pas un gyroscope, mais un gyromètre.

**Un gyroscope permet de mesurer la position angulaire α d’un mobile, alors qu’un gyromètre permet d’en mesurer la vitesse angulaire . Le composant est donc bien un accéléromètre-gyromètre et non un accéléromètre-gyroscope.**

* 1. Préciser les plages de variation possibles de chacun des deux paramètres physiques que le composant est capable de mesurer.

**Pour ce qui concerne fonction accéléromètre, le constructeur propose 4 gammes de mesures possibles (±2, ±4, ±8 ou ±16 g), et pour le gyromètre, 5 gammes de mesures possibles (±125, ±250, ±500, ±1000 et ±2000 dps). 🡪 Page 1/102.**

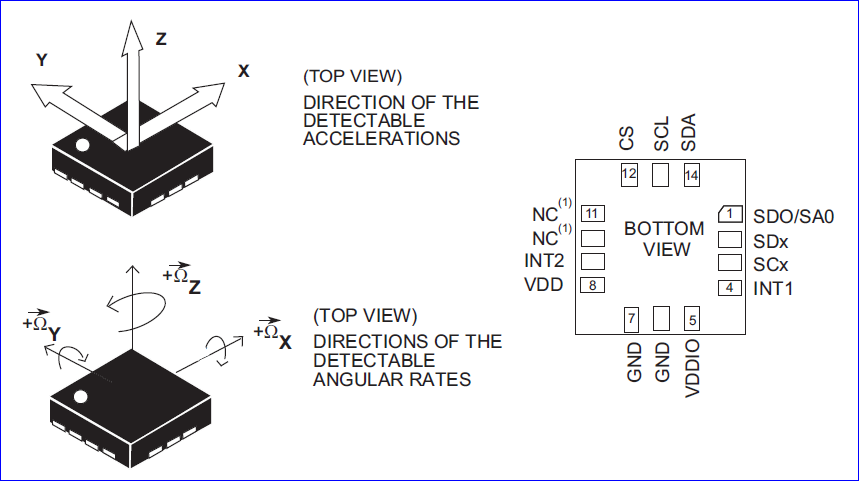
Ce composant peut réaliser ce qu’on appelle une centrale à inertie (ou centrale inertielle), dont une définition, extraite de Wikipedia (<https://fr.wikipedia.org/wiki/Centrale_%C3%A0_inertie>), est donnée ci-dessous :



* 1. Indiquer, par rapport à la représentation physique du composant, où se situent les axes x, y et z, et donner leur appellation en français et en anglais, en se référant aux termes utilisés en dans le domaine de la navigation aérienne ou maritime.

**La documentation précise, page 17/102, la position des axes x, y et z.**

**Selon la façon dont le composant est câble sur sa carte, chacun des 3 axes x, y et z pourra correspondre aux axes de roulis, de tangage ou de cap (on dit aussi lacet), désignés respectivement en anglais, par les termes roll, pitch et yaw axes.**



**On remarquera que les grandeurs vitesse angulaire sont notées ici Ω et non ω.**

* 1. Indiquer par quelle opération mathématique il est possible, comme l’explique l’extrait du site Wikipedia, à partir des informations d’accélération linéaire a et de vitesse angulaire ω, d’estimer, à propos du mobile embarquant le composant LSM6DS3, « son orientation (angles de roulis, de tangage et de cap), sa vitesse linéaire et sa position ».

**L’accéléromètre délivre les informations d’accélérations linéaires ax, ay et az, à partir desquelles on en déduit, après une opération d’intégration, les informations de vitesses linéaires , et ainsi que de positions , et.**

**Le gyromètre délivre les informations de vitesse angulaire ωx, ωy et ωz, à partir desquelles, après intégration, on peut de la même façon calculer les informations de positions angulaires , et .**

* 1. Indiquer dans quels types d’appareils on pourra trouver ce genre de composant.

**Dans un smartphone, une manette de jeu, un casque de réalité virtuelle.**

1. **Les caractéristiques du composant LSM6DS3**

***Caractéristiques électriques***

* 1. Indiquer la référence des broches d’alimentation du composant et justifier la raison pour laquelle le constructeur propose deux broches d’alimentation Vdd distinctes.

**Le composant s’alimente à l’aide des deux broches Vdd et Vdd\_IO (pages 1, 19 et 23/102) :**

* **Vdd constitue l’alimentation interne du composant ;**
* **Vdd\_IO constitue l’alimentation de l’étage d’entrée/sortie du composant (interfaces SPI/I2C).**
  1. Préciser les valeurs d’alimentation possibles que l’on peut appliquer sur ces broches.

**D’après la documentation, page 23/102 :**

* **Vdd : de 1,71 V à 3,6 V ;**
* **Vdd\_IO : de 1,62 V à Vdd+0,1 V (≈ Vdd).**
  1. Indiquer dans quel cas ces deux broches seront reliées au même potentiel et dans quel cas il sera nécessaire de les relier à des potentiels distincts.

**Si le µC est alimenté par une source de tension comprise entre 1,71 V et 3,6 V, on reliera les deux broches d’alimentation Vdd et Vdd\_IOdu composant ensemble et que l’on connectera à la tension d’alimentation du µC.**

**Par contre, si la tension d’alimentation du µC est comprise entre 1,62 V et 1,71 V, on connectera la broche Vdd à n’importe quelle valeur de tension comprise entre 1,71 V et 3,6V, et la broche Vdd\_IO à la tension d’alimentation du µC.**

***Caractéristiques de l’interface de communication avec le microcontrôleur de commande***

* 1. Indiquer les deux interfaces de communication possibles et préciser comment s’effectue la sélection de chacune de ces deux interfaces.

**Interface I2C ou interface SPI (version classique et complète à 4 fils et variante à 3 fils).**

**La sélection de chacun des deux modes s’effectue à partir de la broche CS (table 2 page 19/102) : on sélectionne l’interface I2C en maintenant la broche CS en permanence au niveau haut, sinon, on sélectionne l’interface SPI en connectant cette entrée CS sur la sortie /SS de l’interface SPI du µC qui commande le composant (voir chronogrammes figure 3 page 25/102).**

* 1. Préciser, si l’on valide l’interface SPI du composant, comment configurer cette dernière en interface SPI « classique » à 4 fils ou en variante à 3 fils.

**Le constructeur indique page 40/102, que la sélection SPI 4 fils / 3 fils s’effectue par l’intermédiaire du bit SIM du registre CTRL3\_C : si ce bit est mis à 0 (valeur par défaut), c’est l’interface SPI à 4 fils qui est sélectionnée, sinon, c’est l’interface SPI à 3 fils (page 56/102).**

* 1. Préciser, si on met en œuvre l’interface I2C, combien d’adresses distinctes le composant peut présenter, comment les sélectionner et quelles sont leurs valeurs.

**L’adresse I2C du composant est 110101x où x correspond au niveau logique de la tension appliquée sur la broche SDO/SA0 (page 35/102). Le composant comporte 2 adresses distinctes selon le niveau logique de la tension appliquée sur la broche SDO/SA0 : si cette dernière est connectée à la masse, l’adresse est égale à 1101010, soit 0x6A, et si elle est connectée au potentiel Vdd\_IO, son adresse sera égale à 1101011, soit 0x6B.**

* 1. Indiquer, pour chacune des interfaces de communication, la valeur maximale de la fréquence d’horloge.

**fSPCmax = 10 MHz (Table 6 page 25/102) pour ce qui concerne l’interface SPI, et fSCLmax = 100 kHz en mode « normal » et fSCLmax = 400 kHz en mode « fast » (Table 7 page 26/102) pour ce qui concerne l’interface I2C.**

* 1. Justifier la raison pour laquelle le constructeur donne 2 références distinctes pour la broche 1.

**La double appellation donnée à la broche 1 (SDO/SA0) tient au fait qu’elle remplit deux fonctions distinctes selon l’interface de communication SPI ou I2C mise en œuvre :**

* **en configuration SPI, c’est la fonctionnalité SDO (Serial Data Output) qui est active, ce qui signifie que cette broche est une sortie (qui délivrera donc les valeurs significatives de ax, ay et az ainsi que celles de ωx, ωy et ωz, et qui sera par conséquent connectée sur la broche d’entrée MISO du µC) ;**
* **en configuration I2C, c’est la fonctionnalité SA0 (Slave Address 0) qui est active ; cette broche est une entrée sur laquelle on applique une tension binaire dont le niveau logique associé fixe la valeur du bit de poids faible de l’adresse du composant.**

Le constructeur indique, page 18/102, que le composant peut être configuré selon 2 modes de fonctionnement (Mode 1 et Mode 2) différents. Des explications additionnelles sur le mode 2 sont par ailleurs données pages 70 à 79/119 de la note d’application.

* 1. Indiquer à quoi correspondent ces 2 modes de fonctionnement distincts, et proposer un schéma de câblage partiel (uniquement les signaux associés aux broches 1, 2, 3, 5, 6, 7, 8, 12, 13 et 14 du composant LSM6DS3) :
* d’un LSM6DS3 configuré en mode 1 et piloté par l’interface SPI 4 fils d’un µC ;
* d’un LSM6DS3 configuré en mode 1 et piloté par l’interface I2C d’un µC ;
* d’un LSM6DS3 configuré en mode 2 et associé à 4 composants équipés d’une interface I2C.

**Dans le mode 1, le composant est piloté par l’interface SPI ou I2C d’un µC.**

**C’est le mode « classique » :**





***Mode 1 - Interface SPI 4 fils***  ***Mode 1 - Interface I2C***

**Remarque : le constructeur indique que les broches inutilisées SDx et SCx doivent être connectées à la masse ou à VDDIO (page 19/102) : dans ce cas, elles sont connectées à la masse.**

**Dans le mode 2, le composant est de la même façon piloté par l’interface SPI ou I2C d’un µC, mais il peut servir également de « hub » sur lequel on peut connecter jusqu’à 4 composants distincts équipés d’une interface I2C. L’interface I2C du composant LSM6DS3 (broches SDx et SCx) fonctionne alors en mode Maitre alors que l’interface I2C de chacun des 4 autres composants fonctionne en mode Esclave.**





***Mode 2 – Interface SPI 4 fils*** ***Mode 2 – Interface I2C***

**Remarque : le constructeur indique que les résistances de Pull Up de l’interface I2C constituée par les broches SDx et SCx sont optionnelles (figure 25, page 70/119 de la note d’application) ; dans cet exemple, elles n’ont en effet pas été intégrées.**

***Caractéristiques de transfert Na = f(a) et Nω = f(ω)***

* 1. Donner la signification de l’abréviation anglo-saxonne FS, et préciser à quoi les abréviations LA\_FS et G\_FS proposées Table 3, page 20/102, font référence. Proposer une traduction française à ces abréviations.

**FS signifie « Full Scale », que l’on traduit mot à mot en français par « Pleine Echelle ».**

**LA\_FS (LA comme Linear Acceleration) désigne la plage de variation Δa mesurable, et LA\_G (G comme Gyroscope) désigne la plage de variation Δω mesurable.**

**Pour chacun des 2 paramètres physiques mesurés (a et ω), le constructeur propose 3 gammes de mesures : ±2, ±4, ±8 ou ±16 g pour l’accélération linéaire a, et ±125, ±250, ±500, ±1000 et ±2000 dps pour la vitesse angulaire ω.**

* 1. Relever dans la documentation, la référence des registres destinés au stockage des nombres Nax, Nay, Naz, Nωx, Nωy et Nωz significatifs des valeurs de ax, ay, az,ωx, ωy et ωzet indiquer sur combien de bits ces 6 nombres sont codés. En déduire le nombre de valeurs distinctes que peuvent prendre ces 6 nombres Nax, Nay, Naz, Nωx, Nωx et Nωx.

**D’après la documentation constructeur, pages 65 à 68/102 :**

* **Stockage des valeurs de ax (Nax) : registres OUTX\_L\_XL et OUTX\_H\_XL ;**
* **Stockage des valeurs de ay (Nay) : registres OUTY\_L\_XL et OUTY\_H\_XL ;**
* **Stockage des valeurs de az (Naz) : registres OUTZ\_L\_XL et OUTZ\_H\_XL ;**
* **Stockage des valeurs de ωx (Nωx) : registres OUTX\_L\_G et OUTX\_H\_G ;**
* **Stockage des valeurs de ωy (Nωy) : registres OUTY\_L\_G et OUTY\_H\_G ;**
* **Stockage des valeurs de ωz (Nωz) : registres OUTZ\_L\_G et OUTZ\_H\_G.**

**Chacun de ces registres a un format de 8 bits, et on peut donc en déduire que les nombres Nax, Nay, Naz, Nωx, Nωy et Nωz sont codés sur 16 bits et qu’ils peuvent prendre par conséquent 65536 valeurs distinctes.**

* 1. Proposer la traduction française des abréviations LA\_So et G\_So proposées Table 3, page 20/102, et en donner une définition.

**LA\_So, exprimé en mg/LSB, traduit la variation de la grandeur d’entrée a en fonction de la grandeur de sortie N, c’est-à-dire très exactement l’inverse de la sensibilité du capteur ! L’appellation Sensitivity (sensibilité en français) donnée par le constructeur est donc FAUSSE ! Cette grandeur LA\_So correspond en fait à la résolution de l’accéléromètre, c’est à dire à la plus petite valeur Δa (en mg) qu’il est capable de détecter.**

**De la même façon, la grandeur G\_So, exprimée en mdps/LSB, correspond à la résolution du gyromètre, c’est-à-dire à la plus petite valeur Δω en mdps qu’il est capable de détecter.**

|  |
| --- |
| **Pour en finir avec les ambiguïtés.**  **En métrologie, la résolution d’un capteur correspond à la plus petite valeur de la grandeur d’entrée e qu’il est capable de détecter.**  **La sensibilité S d’un capteur correspond quant à elle à l’évolution de la grandeur de sortie s par rapport à la grandeur d’entrée e, soit .**  **Dans le cas qui nous intéresse, si l’on considère à titre d’exemple l’accéléromètre, dans sa gamme de mesures ±2 g, on peut écrire :**   * **sa résolution est égale à 0,061 mg (page 20/102) ;** * **sa sensibilité S est égale à 1÷0,061 soit 16,393 LSB/mg ou encore 16,393 mg-1.**   **Par ailleurs, pour tout dispositif de conversion analogique-numérique, qu’il s’agisse d’un convertisseur analogique-numérique ou d’un capteur numérique tel que le LSM6DS3, il faut distinguer les deux notions de résolution, selon qu’elles correspondent à la définition anglo-saxonne (resolution sans accent) ou française (résolution avec accent) :**   * **en anglais, la « resolution » correspond au nombre de bits dans lequel le nombre de sortie Ns est exprimé ;** * **en français, la « résolution » correspond à la définition donnée plus haut, c’est-à-dire à la plus petite valeur de la grandeur d’entrée e que le dispositif est capable de détecter.**   **Dans notre cas, la « resolution » du capteur est donc de 16 bits, alors que sa « résolution » est égale à 0,061 mg.**  **Enfin précisons que ce qu’on appelle en français le quantum q, égal par définition à la variation de la grandeur d’entrée Δe provoquant une incrémentation ou une décrémentation du nombre de sortie N, est appelé par les anglo-saxons 1 LSB.**  **Dans ce cas, les français écrivent q = 0,061 mg et les anglo-saxons 1 LSB = 0,061 mg.**  **Ces appellations multiples sont extrêmement regrettables, mais c’est ainsi !** |

* 1. Déduire, en considérant les valeurs numériques des grandeurs LA\_So et G\_So indiquées Table 3, page 20/102, le nombre de valeurs distinctes de Na et de Nω possibles. Comparer ce résultat par rapport à celui exprimé en réponse à la question 4.11 et conclure.

**Pour ce qui concerne l’accéléromètre, on peut raisonner de deux façons.**

1. **Après avoir remarqué que , on peut en déduire que les nombres Na ne pourront prendre que l’ensemble des 65536 valeurs possibles, et qu’on ne pourra pas tout à fait atteindre les valeurs d’extrémités de gamme (±2, ±4, ±8 ou ±16 g) ;**
2. **On peut considérer que les valeurs de LA\_So données par le constructeur (0,061 mg/LSB ; 0,122 mg/LSB ; 0,244 mg/LSB et 0,488 mg/LSB) sont des valeurs approchées (par défaut), mais qu’en réalité, les valeurs de Na, sur chacune des 4 gammes de mesures, prendront 65536 valeurs distinctes (et non 65574).**

**Dans ce cas, on obtiendra des quantums de pour la gamme ±2g, pour la gamme ±4 g, pour la gamme ±8 g et pour la gamme ±16 g.**

**Pour le gyromètre, sachant que , on peut en déduire que les nombres Nω pourront prendre 57142 valeurs distinctes (la deuxième hypothèse b) considérée précédemment ne peut s’appliquer dans cas).**

* 1. Indiquer alors, en décimal, entre quelles valeurs min et max les nombres Na et Nω peuvent varier.

**Ces nombres étant signés, on en déduit :**

* **;**
* **.**
  1. Préciser à quoi correspondent les grandeurs LA\_TY\_Off et G\_TY\_Off mentionnées Table 3, page 20/102.

**Ces deux grandeurs permettent de quantifier l’erreur d’offset du capteur.**

**Ainsi, si ce dernier est dans une position fixe, les valeurs de Nax, Nay, Nωx, Nωy et Nωz ne seront pas strictement nulles, et pourront traduire des valeurs de ax, de ay et de az pouvant varier entre – 40 et +40 mg, et de ωx, de ωy et de ωz pouvant varier entre -10 et +10 dps.**

* 1. Indiquer ce que signifie l’abréviation ODR, et en déduire à quoi correspondent les grandeurs LA\_ODR et G\_ODR mentionnées Table 3, page 21/102.

**ODR signifie Output Data Rate, et correspond au nombre de mesures que le capteur effectue par seconde, ou encore la fréquence avec laquelle ces mesures s’effectuent.**

**Par conséquent :**

* **LA\_ODR désigne les différentes valeurs de fréquences avec lesquelles l’accéléromètre peut effectuer une mesure d’accélération linéaire a (de 12,5 à 6664 mesures d’accélération linéaire par seconde) ;**
* **G\_ODR désigne les différentes valeurs de fréquences avec lesquelles le gyromètre peut effectuer une mesure de vitesse angulaire ω (de 12,5 à 1666 mesures de vitesse angulaire par seconde).**

Le composant LSM6DS3 est un composant relativement complexe. Sa documentation s’avère insuffisante pour comprendre toutes les fonctionnalités qu’il intègre.

Heureusement, le constructeur propose une note d’application expliquant en détails le fonctionnement interne du composant, afin de permettre aux techniciens supérieurs et aux ingénieurs d’être en mesure de le mettre en œuvre de façon pertinente.

Nous nous intéressons dans ce qui suit à cette note d’application AN4650.

1. **La programmation du composant LIS3MDL**

***Les différents modes de fonctionnement du composant LSM6DS3***

Le constructeur indique, page 20/119, que le composant peut être validé partiellement (validation de l’accéléromètre seul ou du gyromètre seul) ou totalement (validation de l’accéléromètre et du gyromètre).

Il précise également que l’accéléromètre peut être configuré selon 4 modes de fonctionnement distincts et le gyromètre selon 5 modes de fonctionnement distincts.

* 1. Préciser le nom du registre ainsi que la référence et la valeur des bits permettant d’invalider l’accéléromètre. Même question pour invalider le gyromètre.

**Invalidation de l’accéléromètre : bits ODR\_XL[3:0] = 0000 (registre CTRL1\_XL) ;**

**Invalidation du gyromètre : bits ODR\_G[3:0] = 0000 (registre CTRL2\_G).**

* 1. Préciser le nom des différents modes de fonctionnement associés à l’accéléromètre et au gyromètre, et indiquer le nom du registre ainsi que la référence des bits permettant la sélection de ces différents modes de fonctionnement.

**Pour ce qui concerne l’accéléromètre, d’après la table 4 page 20/119 :**

* **Mode High Performance : bits ODR\_XL[3:0] du registre CTRL1\_XL et bit XL\_HM\_MODE = 0 du registre CTRL6\_C ;**
* **Mode Normal : bits ODR\_XL[3:0] du registre CTRL1\_XL et bit XL\_HM\_MODE = 1 du registre CTRL6\_C ;**
* **Mode Low Power : bits ODR\_XL[3:0] du registre CTRL1\_XL et bit XL\_HM\_MODE = 1 du registre CTRL6\_C ;**
* **Mode Power Down : bits ODR\_XL[3:0] = 0000 du registre CTRL1\_XL.**

**Pour ce qui concerne le gyromètre, d’après la table 5 page 21/119 :**

* **Mode High Performance : bits ODR\_G[3:0] du registre CTRL2\_G et bit G\_HM\_MODE = 0 du registre CTRL7\_G ;**
* **Mode Normal : bits ODR\_G[3:0] du registre CTRL2\_G et bit G\_HM\_MODE = 1 du registre CTRL7\_G ;**
* **Mode Low Power : bits ODR\_G[3:0] du registre CTRL2\_G et bit G\_HM\_MODE = 1 du registre CTRL7\_G ;**
* **Mode Power Down : bits ODR\_G[3:0] = 0000 du registre CTRL2\_G.**
* **Mode Sleep : bit SLEEP\_G = 1 du registre CTRL4\_C (table 55 page 57/102).**
  1. Expliquer la différence entre ces différents modes de fonctionnement.

**Le paragraphe 3.3 page 22/119 précise que ce mode de fonctionnement « High Performance » est optimal en termes de niveau de bruit, alors que le mode « Normal » est moins énergivore.**

**En mode « Low Power », comme on peut l’imaginer, la consommation est encore plus faible.**

**On constate, d’après la table 6 page 21/119, que le mode High Performance consomme en effet davantage que le mode Normal, qui consomme lui-même plus que le mode Low Power.**

**Le mode Low Power consomme certes peu, mais est réservé aux 3 valeurs d’ODR les plus faibles (12,5 Hz, 26 Hz et 52 Hz) ; le mode Normal permet d’atteindre les 2 valeurs d’ODR suivantes (104 et 208 Hz), alors qu’en mode High Performance, toutes les valeurs d’ODR, y-compris les plus élevées, sont possibles.**

**On voit l’intérêt, en termes de consommation, lorsqu’on n’a besoin que d’un seul capteur (accéléromètre ou gyromètre), de pouvoir déconnecter l’autre capteur (table 6 page 21/119).**

**Enfin on pourra remarquer que le gyromètre consomme beaucoup plus que l’accéléromètre.**

***Le traitement de signal au sein du composant LSM6DS3***

Le constructeur décrit la structure interne du composant LSM6DS3, depuis le paramètre physique d’entrée (accélération linéaire a et vitesse angulaire ω) jusqu’aux registres de sortie OUT\_XL et OUT\_G (paragraphes 3.8 et 3.9, pages 25 à 35/119).

Le traitement de l’accélération linéaire a (paragraphe 3.8, pages 25 à 30/119)

Le constructeur propose le synoptique de traitement de signal lié à l’accéléromètre suivant :



L’entrée de ce synoptique correspond au signal électrique significatif de l’accélération linéaire ax, ay ou az à laquelle le composant LSM6DS3 est soumis, et la sortie à la paire de registres (OUTX\_L\_XL,OUTX\_H\_XL),(OUTY\_L\_XL,OUTY\_H\_XL) ou(OUTZ\_L\_XL,OUTZ\_H\_XL).

Le traitement de signal à l’intérieur du composant est numérique, c’est pourquoi l’étage d’entrée est constitué d’un filtre anti-repliement, d’un échantillonneur-bloqueur et d’un convertisseur analogique-numérique (l’échantillonneur-bloqueur, non représenté ici, étant supposé intégré au convertisseur).

Ensuite, le signal est donc numérique, et traité à l’aide d’un certain nombre de filtres, permanent pour l’un d’entre eux (LPF1), et optionnels pour les autres (HP, SLOPE et LPF2), la sélection de ces derniers s’effectuant à l’aide de deux multiplexeurs logiciels représentés ici par des trapèzes.

Les caractéristiques des filtres (essentiellement la fréquence de coupure) sont programmables.

On ne s’intéressera pas dans ce chapitre, aux blocs fonctionnels représentés en bleu.

* 1. Préciser si le filtre analogique passe-bas d’entrée est implanté systématiquement, quel que soit le mode de fonctionnement de l’accéléromètre, ou bien dans certaines conditions seulement. En déduire la fonction première de ce filtre.

**La documentation précise, page 25/119, que ce filtre n’est implanté qu’en mode « High Performance », et comme on a vu précédemment (réponse à la question 4.3), que ce mode était optimal en termes de niveau de bruit, on en déduit logiquement que la fonction première de ce filtre passe bas est de filtrer le bruit mécanique et électrique de fréquence élevée (des valeurs de et de élevées et non significatives de l’accélération linéaire).**

* 1. Rappeler la fonction d’un filtre anti-repliement au sein d’une chaine de traitement numérique de signal, ainsi que la relation d’inégalité liant les grandeurs Fe (fréquence d’échantillonnage) et Fc (fréquence de coupure du filtre).

**Un filtre anti-repliement a pour fonction d’empêcher le phénomène de recouvrement spectral inhérent à l’opération de numérisation d’un signal analogique.**

**Il faut pour cela, respecter le théorème de Shannon qui indique que le signal doit être échantillonné avec une fréquence Fe au moins deux fois supérieure à la composante spectrale du signal d’entrée de fréquence la plus élevée fmax : Fe > 2.fmax.**

**Or, le spectre des signaux analogiques est rarement borné (), c’est pourquoi on implante un filtre passe-bas anti-repliement en entrée de chaîne de traitement de signal, dont la fréquence de coupure Fc doit respecter le théorème de Shannon, donc telle que Fe > 2.Fc.**

* 1. Préciser la relation entre la fréquence d’échantillonnage et la grandeur ODR, et vérifier la cohérence des deux premières colonnes de la table 8, page 25/119.

**Dans notre cas, la valeur associée à la grandeur ODR correspond à la fréquence d’échantillonnage Fe.**

**Et précisément, les indications portées dans les deux premières colonnes de la table 8 ne respectent pas toujours le théorème de Shannon (en tout cas pas pour les 3 premières valeurs d’ODR égales à 12,5 Hz, 26 Hz et 52 Hz).**

**Je ne sais expliquer cela …, c’est quand même problématique sur le plan pédagogique.**

* 1. Indiquer le nom des registres ainsi que la référence des bits permettant de choisir la fréquence de coupure Fc du filtre passe-bas anti-repliement.

**La sélection des 4 valeurs de fréquence de coupure Fc possibles (50, 100, 200 et 400 Hz) s’effectue à l’aide des 2 bits BW\_XL[1:0] (registre de commande CTR1\_XL) et du bit XL\_BW\_SCAL\_ODR (registre de commande CTRL4\_C) à positionner à 1.**

* 1. Indiquer le nom du registre ainsi que la référence des bits permettant de sélectionner la fréquence de coupure du filtre numérique passe-bas LPF1.

**La fréquence de coupure de ce filtre n’est pas programmable, elle dépend de la valeur d’ODR sélectionnée. Par conséquent les valeurs de fréquence de coupure annoncées table 9 page 26/119 sont sélectionnées en même temps que l’on choisit la valeur d’ODR (bits ODR\_XL[3:0] du registre CTRL1\_XL).**

* 1. Indiquer à quoi correspond la grandeur « Total BW [Hz] » de la colonne de droite de la table 10, page 27/119, et justifier les valeurs indiquées.

**Il s’agit de la fréquence de coupure (ou la bande passante) du filtre passe-bas global constitué par la mise en cascade du filtre analogique passe-bas anti-repliement et du filtre numérique passe-bas LPF1.**

**Cette fréquence de coupure correspond évidemment à la fréquence de coupure la plus faible de chacun des deux filtres.**

Nous nous intéressons désormais aux filtres optionnels.

* 1. Indiquer le nom des registres ainsi que la référence et la valeur des bits permettant de mettre en œuvre :

1. aucun filtre optionnel,
2. le filtre numérique passe-haut,
3. le filtre numérique « SLOPE »,
4. le filtre numérique passe-bas LPF2.

**La sélection de ces filtres s’effectue à partir du registre CTRL8\_XL :**

* **aucun filtre : bit HP\_SLOPE\_XL\_EN à positionner à 0,**
* **filtre numérique passe-haut : bits HPCF\_XL[1:0] à positionner à 11, 01 ou 10, bit HP\_SLOPE\_XL\_EN à positionner à 1 et bit LPF2\_XL\_EN à positionner à 0,**
* **filtre numérique « SLOPE » : bits HPCF\_XL[1:0] à positionner à 00, bit HP\_SLOPE\_XL\_EN à positionner à 1 et bit LPF2\_XL\_EN à positionner à 0,**
* **filtre numérique passe-bas LPF2 : bit HP\_SLOPE\_XL\_EN à positionner à 1 et bit LPF2\_XL\_EN à positionner à 1.**
  1. Préciser, parmi ces filtres, ceux dont la fréquence de coupure est programmable et indiquer dans ce cas, le nom du registre ainsi que la référence des bits permettant de procéder à la sélection de la fréquence de coupure.

**La fréquence de coupure des filtres numériques passe-bas et passe-haut est programmable à l’aide des bits HPCF\_XL[1:0] du registre CTRL8\_XL, conformément aux tables 11 et 12 de la page 29/119, alors que celle du filtre numérique « SLOPE » est fixe (Fc = ODR\_XL/4).**

On se propose pour terminer, de caractériser le filtre « SLOPE » (page 30/119), dont la présence sera justifiée ultérieurement.

* 1. Ecrire l’équation de récurrence du filtre sous sa forme canonique (en remplaçant les échantillons « Slope » et « acc » par respectivement « s » et « e »), et donner l’appellation du filtre ainsi constitué.

**L’équation de récurrence devient . Il s’agit de l’équation de récurrence d’un filtre numérique dérivateur, ce que la figure 6 met clairement en évidence.**

Le traitement de la vitesse angulaire ω (paragraphe 3.9, pages 31 et 32/119)

Le constructeur propose le synoptique de traitement de signal lié au gyromètre suivant :



L’entrée de ce synoptique correspond au signal électrique significatif de la vitesse angulaire ωx, ωy ou ωz à laquelle le composant LSM6DS3 est soumis, et la sortie à la paire de registres (OUTX\_L\_G,OUTX\_H\_G),(OUTY\_L\_G,OUTY\_H\_G) ou(OUTZ\_L\_G,OUTZ\_H\_G).

* 1. En se basant sur les explications données précédemment, proposer une description du traitement de signal effectué au sein du gyromètre, et indiquer les bits et les registres auxquels il faudra accéder pour le mettre en œuvre.

**Nous retrouvons le filtre analogique passe-bas anti-repliement, dont la fréquence de coupure est fixe et inconnue, suivi du convertisseur analogique-numérique.**

**On trouve ensuite un filtre numérique passe-bas, dont la fréquence de coupure dépend à la fois du mode de fonctionnement (High Performance, Normal ou Low Power) et de la valeur d’ODR choisie (tables 13 et 14, page 31/119), sélectionnée à l’aide des bits ODR\_G[3:0] du registre CTRL2\_G.**

**Enfin, on peut, en positionnant le bit HP\_G\_EN du registre CTRL7\_G à 1, ajouter un traitement de filtrage numérique passe-haut, dont les 4 fréquences de coupure possibles (table 15, page 32/119) sont programmables à l’aide des bits HPCF\_G[1:0] du registre CTRL7\_G.**

On retiendra de l’étude de la structure interne du capteur LSM6DS3, que les caractéristiques des différents filtres (type de filtre et fréquence de coupure) seront ajustées de façon empirique et expérimentale, en fonction de l’application à réaliser (pour prendre un exemple simple, on imagine aisément que les caractéristiques des filtres de ce capteur seront programmées de façon bien différente, selon qu’il est intégré dans le casque de réalité virtuelle d’un joueur qui se bat contre des créatures virtuelles extrêmement agressives ou bien dans celui d’un interne en chirurgie effectuant une opération du cœur virtuelle dans le cadre de sa formation).

***La lecture des valeurs de Na et Nω significatives des grandeurs physiques a et ω***

Le paragraphe 4, pages 36 à 44/119, décrit les différentes manières d’accéder par programme, aux valeurs de Nax, Nay, Naz, Nωx, Nωy et Nωz significatives respectivement des grandeurs physiques ax, ay, az,ωx, ωy et ωz.

* 1. Préciser les 2 méthodes permettant de savoir à quel instant une nouvelle valeur de Nax, de Nay, de Naz, de Nωx, de Nωy et/ou de Nωz est disponible, et indiquer alors, pour chacun des deux cas, les actions logicielles à effectuer pour récupérer cette ou ces valeur(s).

**Comme pour tout capteur, on peut opérer par scrutation (polling) ou par interruption.**

**En mode scrutation, il suffit de lire le bit du registre d’état (STATUS\_REG) XLDA pour l’accéléromètre ou GDA pour le gyromètre : s’il est à 1, cela signifie qu’une nouvelle donnée N est disponible à laquelle on accèdera par une opération de lecture des registres de sortie OUT\_XL et/ou OUT\_G adéquat(s).**

**Le mode interruption consiste à transmettre sur l’une des broches INT1 ou INT2, une image matérielle (donc une tension) du bit XLDA ou GDA : le signal ainsi élaboré présente une impulsion active au niveau haut à chaque fois qu’une nouvelle valeur de N est disponible (figure 9, page 37/119). Il suffit alors, pour connaitre la valeur de N, de connecter ce signal sur une entrée d’interruption du µC, qui aura pour effet, à chaque fois qu’une nouvelle valeur de N est disponible, de déclencher l’exécution d’une fonction d’IT consistant à lire les registres de sortie OUT\_XL et/ou OUT\_G concerné(s).**

**Pour valider ce signal sur la sortie INT1, il faut positionner à 1 le bit :**

* **INT1\_DRDY\_XL du registre INT1\_CTRL (accès aux valeurs de Nax, Nay et/ou Naz) ;**
* **INT1\_DRDY\_G du registre INT1\_CTRL (accès aux valeurs de Nωx, Nωy et/ou Nωz).**

**De la même façon, pour valider ce signal sur la sortie INT2, il faut positionner à 1 le bit :**

* **INT2\_DRDY\_XL du registre INT2\_CTRL (accès aux valeurs de Nax, Nay et/ou Naz) ;**
* **INT2\_DRDY\_G du registre INT2\_CTRL (accès aux valeurs de Nωx, Nωy et/ou Nωz).**
  1. Expliquer à quoi correspondent les formats « Little-Endian » et « Big-Endian » et indiquer le bit et le registre auquel il faut accéder pour sélectionner le format dans lequel on souhaite mémoriser les valeurs de N dans les registres de sortie. Vérifier alors la pertinence des valeurs indiquées tables 19 et 20, page 39/119.

**La valeur de N, codée sur 16 bits, est mémorisée dans 2 registres de 8 bits contigus (d’adresses n et n+1).**

**Il existe deux façons de mémoriser la valeur de N :**

* **les 8 bits de poids faible de N sont stockés dans le registre d’adresse basse (n) et les 8 bits de poids fort dans le registre d’adresse immédiatement supérieure (n+1) : c’est le format « Little-Endian » ;**
* **les 8 bits de poids faible de N sont stockés dans le registre d’adresse haute (n+1) et les 8 bits de poids fort dans le registre d’adresse immédiatement inférieure (n) : c’est le format « Big-Endian ».**

**On sélectionne le format « Little-Endian » en positionnant le bit BLE du registre CTRL3\_C à 0, et le format « Big-Endian » en positionnant ce même bit à 1 (page 38/119).**

**Le quantum de conversion Na = f(a), pour la gamme ±2 g, étant égal à 0,061 mg (page 20/102 de la documentation du LSM6DS3), on obtient, pour ce qui concerne la table 19 :**

* **pour a = 350 mg,**
* **pour a = 1 g,**
* **pour a = -350 mg,**
* **pour a = -1 g,**

**Le quantum de conversion Nω = f(ω), pour la gamme ±250 dps, étant égal à 8,75 mdps (page 20/102 de la documentation du LSM6DS3), on obtient, pour la table 20 :**

* **pour ω = 100 dps,**
* **pour ω = 200 dps,**
* **pour ω = -100 dps,**
* **pour ω = -200 dps,**

**On obtient quasiment les mêmes valeurs (erreurs d’arrondi sur les valeurs du quantum).**

* 1. Indiquer le mécanisme permettant de lire successivement les valeurs de :
* Nωx, Nωy puis Nωz,
* Nax, Nay puis Naz,
* Nωx, Nωy, Nωz, Nax, Nay, puis Naz,

sans avoir à mentionner l’adresse de chacun des registres OUT\_XL et/ou OUT\_G à l’intérieur desquels ces valeurs de N sont mémorisées. Préciser les bits ainsi que le registre auquel il faut accéder pour mettre en œuvre ce mécanisme.

**Les 12 registres de sortie :**

* **OUTX\_L\_G + OUTX\_H\_G pour le stockage de Nωx,**
* **OUTY\_L\_G + OUTY\_H\_G pour le stockage de Nωy,**
* **OUTZ\_L\_G + OUTZ\_H\_G pour le stockage de Nωz,**
* **OUTX\_L\_XL + OUTX\_H\_XL pour le stockage de Nax,**
* **OUTY\_L\_XL + OUTY\_H\_XL pour le stockage de Nay,**
* **OUTZ\_L\_XL + OUTZ\_H\_XL pour le stockage de Naz,**

**sont positionnés respectivement aux adresses successives 0x22 à 0x2D (table 16, page 44/102 de la documentation du LSM6DS3).**

**Il est possible de faire un accès multiple en lecture en précisant, par l’intermédiaire des 3 bits ROUNDING[2:0] du registre CTRL5\_C, à quelles valeurs de N on souhaite accéder (page 40/119) :**

* **si ROUNDING[2:0] = 001, on accède successivement aux valeurs de Nax, Nay puis Naz, par le biais d’une opération de lecture de 6 octets successifs (48 impulsions d’horloge) ;**
* **si ROUNDING[2:0] = 010, on accède successivement aux valeurs de Nωx, Nωy puis Nωz, par le biais d’une opération de lecture de 6 octets successifs (48 impulsions d’horloge) ;**
* **si ROUNDING[2:0] = 011, on accède successivement aux valeurs de Nωx, Nωy, Nωz, Nax, Nay puis Naz, par le biais d’une opération de lecture de 12 octets successifs (96 impulsions d’horloge).**

**Une fois la lecture terminée, le pointeur d’adresses revient à sa position initiale, ce qui explique l’appellation anglo-saxonne du mécanisme (ROUNDING) que l’on pourrait traduire en français par adressage circulaire.**

**Evidemment, pour accéder aux différents registres de façon classique (en mentionnant expressément leur adresse), il faut invalider cet adressage circulaire, en positionnant ROUNDING[2:0] à 000.**

* 1. Expliquer le mécanisme prévu par le constructeur permettant de déclencher la mise à jour des registres de sortie du gyromètre OUT\_G sur un évènement externe que l’on précisera. Indiquer les bits ainsi que le registre auquel il faut accéder pour mettre en œuvre ce mécanisme.

**Il est possible de déclencher la mise à jour des registres OUT\_G du gyromètre (cette fonctionnalité ne s’applique pas à l’accéléromètre), lorsqu’un évènement qui peut être un front montant (edge-sensitive) ou un niveau haut (level-sensitive) est appliqué sur la broche INT2, alors configurée en entrée (pages 41 à 43/119).**

**En configuration « edge-sensitive », les registres OUT\_G sont mis à jour juste après apparition du front montant appliqué sur l’entrée INT2 (figure 10, page 41/119), alors qu’en mode « level-sensitive », ils sont mis à jour durant toute la durée pendant laquelle l’impulsion appliquée sur l’entrée INT2 est au niveau haut, avec en plus mise à 1 du LSB de Nω (figure 12, page 43/119).**

**Le choix de l’une de ces deux configurations s’effectue par programmation des bits TRIG\_EN, LVLen et LVL2\_EN (registre CTL6\_C), conformément à la table 22, page 41/119.**

* 1. Expliquer comment il est possible de modifier de façon logicielle, l’orientation des axes de référence du gyromètre (roulis, tangage et lacet), par rapport à celle définie par défaut et précisée à la question 3.5. Indiquer alors les bits ainsi que le registre auquel il faut accéder pour modifier l’orientation des axes.

**L’orientation des axes de roulis, tangage et lacet, telle que définie par défaut, peut être modifiée logiciellement (pages 43 et 44/119).**

**Le changement d’orientation s’effectue par l’intermédiaire des bits Orient\_[2:0] du registre ORIENT\_CFG\_G, et le sens de rotation par rapport à l’axe (horaire ou trigonométrique), par l’intermédiaire des bits SignX\_G, SignY\_G et SignZ\_G, conformément à la table 24, page 44/119.**

***La capacité du LSM6DS3 à élaborer des requêtes d’interruption***

Nous avons vu à la question 5.14, que le composant pouvait être configuré (logiciellement) et câblé (matériellement) de façon à élaborer auprès du µC qui le pilote, une requête d’interruption à chaque fois qu’une nouvelle valeur de mesure d’accélération linéaire a ou de vitesse angulaire ω est disponible.

En réalité, le composant peut élaborer de nombreuses requêtes d’interruption supplémentaires, que nous nous proposons de lister dans ce paragraphe.

Les informations à prendre en compte pour la suite de ce TD sont données paragraphe 5 de la note d’application, aux pages 45 à 61/119.

On rappelle ci-dessous la liste des 9 modes de fonctionnement (page 45/119) auxquels peuvent être associés la génération d’une requête d’interruption :

* Free-fall ;
* Wake-up ;
* 6D/4D orientation detection ;
* Single-tap and double-tap sensing ;
* Activity/Inactivity detection ;
* Pedometer functions ;
* Significant motion ;
* Tilt ;
* Timestamp.
  1. Préciser la référence des registres dédiés à la configuration des différentes requêtes d’interruption que le composant LSM6DS3 est capable d’élaborer.

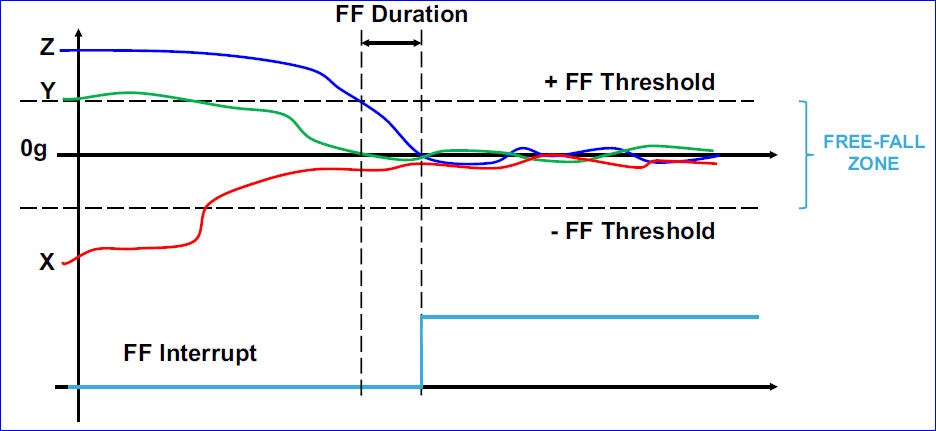
**Il s’agit des 4 registres INT1\_CTRL, INT2\_CTRL, MD1\_CFG et MD2\_CFG.**

Afin de ne pas alourdir inconsidérément ce TD, on se limitera dans ce qui suit, à décrire chacun des modes 9 modes de fonctionnement, en expliquant les évènements conduisant à l’élaboration d’une requête d’interruption, ainsi que le principe mis en œuvre pour détecter ces évènements.

* 1. Expliquer à quoi correspond la situation « Free-fall », ainsi que le principe mis en œuvre permettant de la détecter.

**La situation « Free-fall » correspond à l’arrêt de toute activité d’accélération sur l’ensemble des 3 axes. En toute rigueur, cette situation se produit lorsque la vitesse linéaire sur les 3 axes est constante, mais ce cas de figure étant peu probable, nous supposerons qu’il s’agit du cas particulier où la vitesse est certes constante, mais carrément nulle, d’où l’appellation « Free-fall » (chute libre) choisie par le constructeur.**

**Cette situation sera avérée lorsque les 3 valeurs de ax, ay et az seront comprises entre deux valeurs seuil programmables (free-fall zone), et ce, pendant une durée (FF duration) supérieure à un seuil lui aussi programmable (figure15, page 47/119) :**

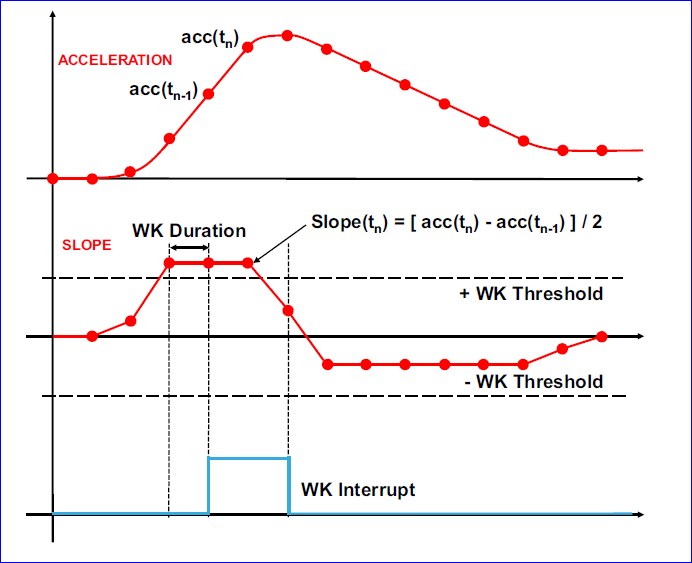


* 1. Expliquer à quoi correspond la situation « Wake-up », ainsi que le principe de fonctionnement permettant de la détecter.

**La situation « Wake-up » (réveil) correspond à la situation inverse de celle traitée juste avant (Free-fall), et permet donc de détecter la (re)mise en mouvement de l’élément sur lequel est implanté l’accéléromètre.**

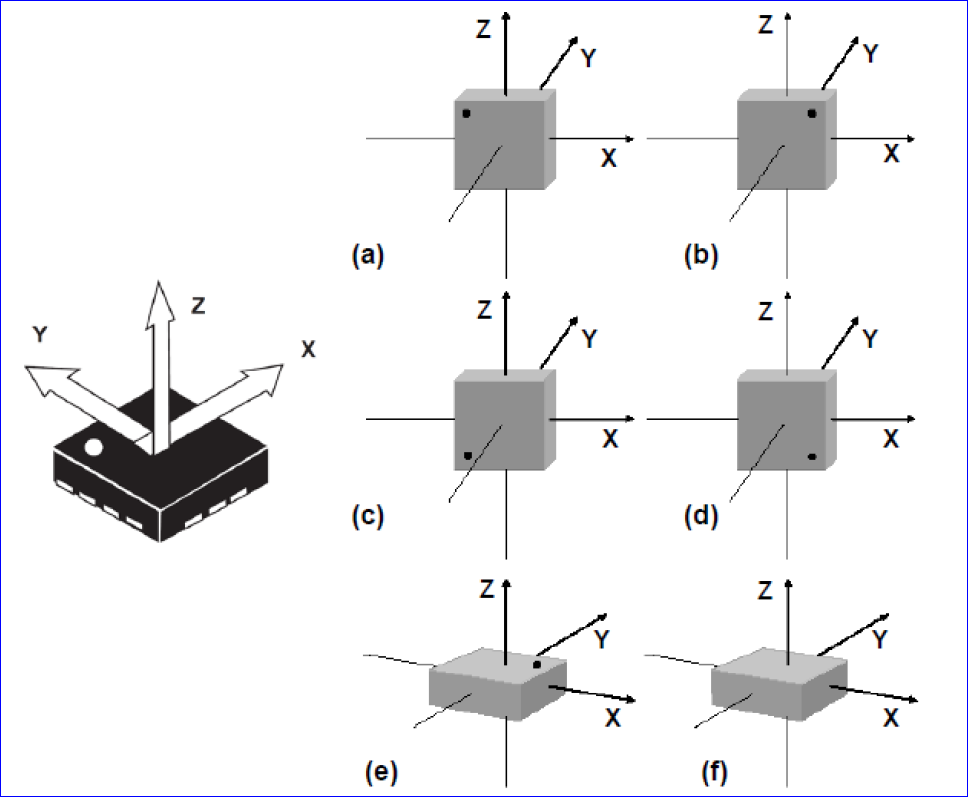
**Comme l’indique la figure 5 page 28/119, cette situation sera détectable à l’aide du filtre passe-haut ou du filtre « SLOPE ».**

**Le principe permettant de détecter une situation de réveil consiste à détecter sur n’importe lequel des axes x, y ou z, une variation supérieure à un certain seuil, c’est-à-dire une valeur de N issue de la sortie du filtre SLOPE supérieure au seuil « WK Threshold » programmable, et ce, pendant une durée supérieure au seuil « WK Duration », elle aussi programmable (figure 16, page 49/119) :**



* 1. Expliquer à quoi correspond la situation « 6D/4D orientation detection », ainsi que le principe de fonctionnement permettant de la détecter.

**Le composant a la possibilité de repérer les 6 orientations (figure 17, page 52/119) rappelées ci-dessous, et de générer une requête d’interruption à chaque fois qu’il passe d’une orientation à la suivante () :**



**Contrairement à ce que l’on pourrait penser, ce n’est pas avec le gyromètre que cette situation est détectée, mais avec l’accéléromètre.**

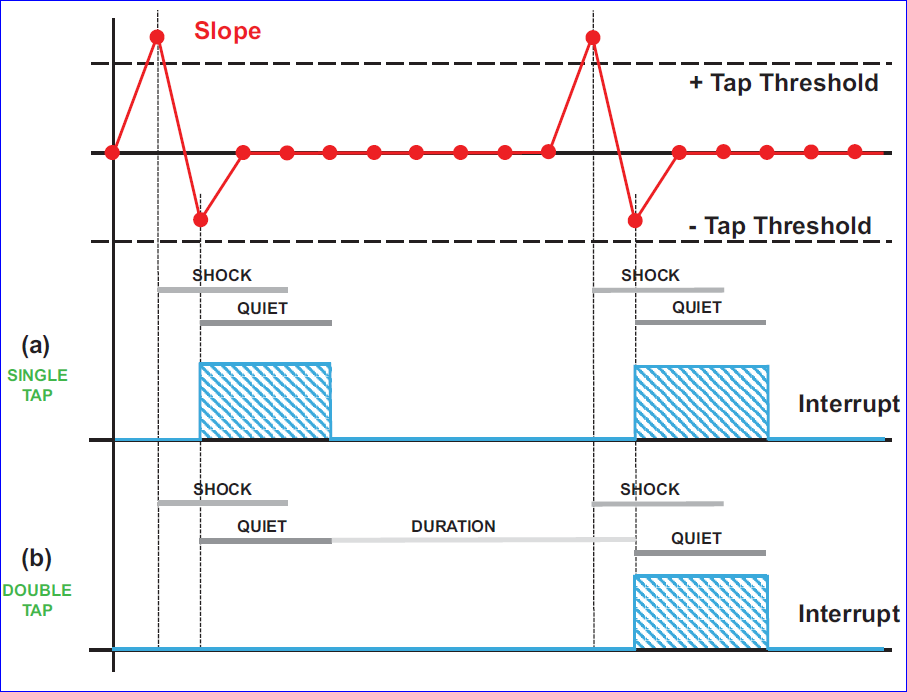
**Le principe consiste à détecter une accélération a sur l’un des axes x, y ou z supérieure à un certain seuil bas programmable, et dans le même temps une accélération a sur les deux autres axes inférieure à un seuil haut programmable, et ce, pendant une durée supérieure à (au moins deux valeurs successives de Na).**

**La détection d’orientation en 4D ne concerne que les cas a, b, c et d (voir figure page précédente) ; elle sera mise en œuvre dans le cas d’un smartphone, par exemple, pour basculer l’affichage de mode portrait à paysage ou inversement, alors que la détection d’orientation 6D, plus complète, sera utilisée pour les casques de réalité virtuelle.**

* 1. Expliquer à quoi correspondent les situations « Single-tap and double-tap sensing », ainsi que le principe de fonctionnement permettant de les détecter.

**L’expression « Single-tap and double-tap sensing » fait référence au dispositif de détection de pression manuelle (le plus souvent avec le doigt) simple ou double, que l’on trouve à titre d’exemple, systématiquement intégré aux smartphones.**

**Là encore, le système de détection utilise le filtre SLOPE, et la détection d’une pression simple ou double peut se faire selon n’importe lequel des axes x, y ou z.**



**Le principe, pour une détection de pression simple (cas a) est le suivant :**

**Si la sortie du filtre délivre au moins une ou plusieurs valeurs de N supérieure(s) au seuil programmable « Tap Treshold » pendant une durée inférieure à la durée programmable « SHOCK », alors une requête d’interruption sera élaborée pendant la durée programmable « QUIET ».**

**Pour ce qui concerne la détection par pression double (cas b), il y aura élaboration d’une requête d’interruption, dans les mêmes conditions que précédemment, à condition que la deuxième pression intervienne, après la première pression, au bout d’une durée inférieure à la valeur de la durée programmable « QUIET » et de la durée programmable « DURATION » cumulées.**

* 1. Expliquer à quoi correspondent les situations « Activity/Inactivity detection », ainsi que le principe de fonctionnement permettant de les détecter.

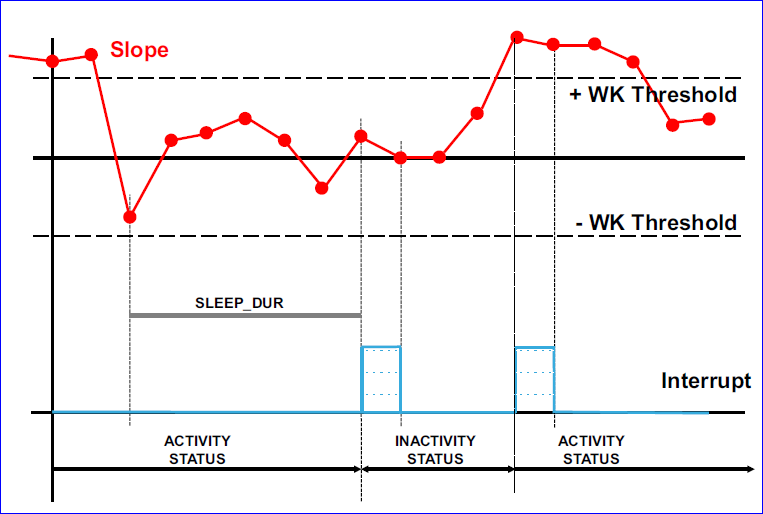
**La situation « Activity/Inactivity detection » a pour objet de forcer automatiquement la valeur d’ODR de l’accéléromètre à 12,5 Hz (la valeur la plus faible) lorsque le composant ne subit aucun mouvement (situation d’inactivité), et de rétablir la valeur d’ODR à sa valeur initialement programmée lorsque le composant est à nouveau le siège de mouvements (situation d’activité déclenchée par l’évènement « Wake-up »).**

**L’objectif consiste, évidemment, à réduire la consommation du composant en cas d’inactivité (d’après la table 6, page 21/119, le courant consommé avec ODR = 12,5 Hz est de 24 µA, contre 240 µA pour des valeurs d’ODR allant jusqu’à 833 Hz, qui constitue la valeur maximale au-delà de laquelle la fonctionnalité détection d’activité/inactivité ne peut être implantée).**

**La détection d’activité ou d’inactivité s’effectue à l’aide du filtre SLOPE.**

**Si l’amplitude des échantillons de sortie du filtre SLOPE est inférieure au seuil programmable WK Threshold, et ce, pendant une durée supérieure à la durée programmable SLEEP\_DUR, alors le composant délivre une requête d’interruption significative de la détection d’inactivité.**

**Ultérieurement, lorsque le composant sera à nouveau le siège de mouvements, la détection de situation d’activité s’effectuera conformément à la réponse à la question 5.21.**



* 1. Expliquer à quoi correspond l’application « Pedometer functions », ainsi que le principe de fonctionnement permettant de la mettre en œuvre.

**La fonction podomètre (pedometer) consiste à comptabiliser le nombre de pas, ce qui suppose que le composant soit « embarqué » sur un humain (poche, sac, …).**

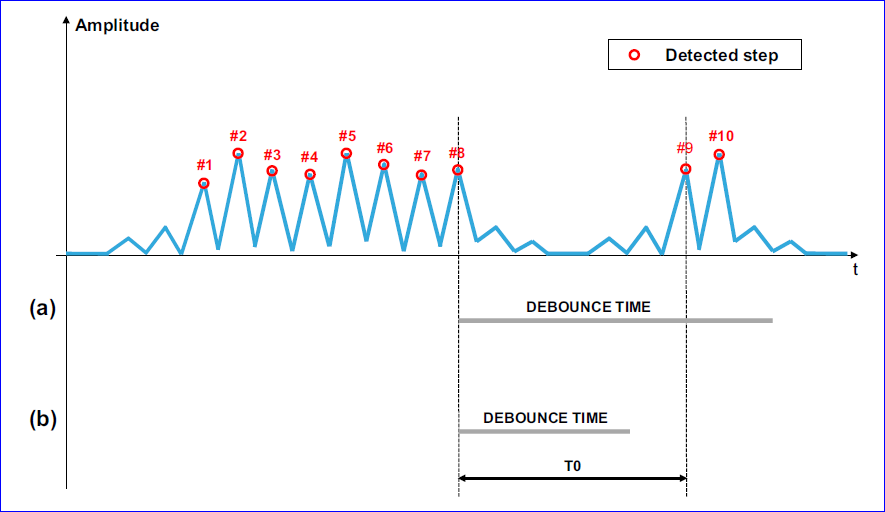
**Sa mise en œuvre suppose que l’accéléromètre soit configuré pour la gamme ±2 g.**

**Le principe consiste à générer une requête d’interruption à chaque fois qu’un pas est détecté (la fonction d’interruption associée se chargera simplement d’incrémenter d’une unité une variable destinée à comptabiliser le cumul de pas détectés).**

**Le seul réglage, pour cette application, porte sur la durée « DEBOUNCE TIME » qui suit la détection d’un arrêt des pas.**

**Dans le cas a), les deux pas isolés 9 et 10 seront comptabilisés car ils se produisent durant la fenêtre temporelle DEBOUNCE, alors que dans le cas b) ils ne seront pas pris en compte.**

**Ce dispositif permet de ne considérer que les séances de marche significatives (suites de pas) et non des pas isolés qui peuvent s’avérer être en effet des pas, mais aussi des aléas.**



* 1. Expliquer à quoi correspond l’application « Significant motion », ainsi que le principe de fonctionnement permettant de la mettre en œuvre.

**Cette application est un cas particulier de la fonction podomètre, pour laquelle on ne comptabilise pas des pas, mais des mouvements « significatifs » tels que le déplacement d’un humain.**

**Elle élabore une requête d’interruption à chaque fois que le mouvement significatif est détecté.**

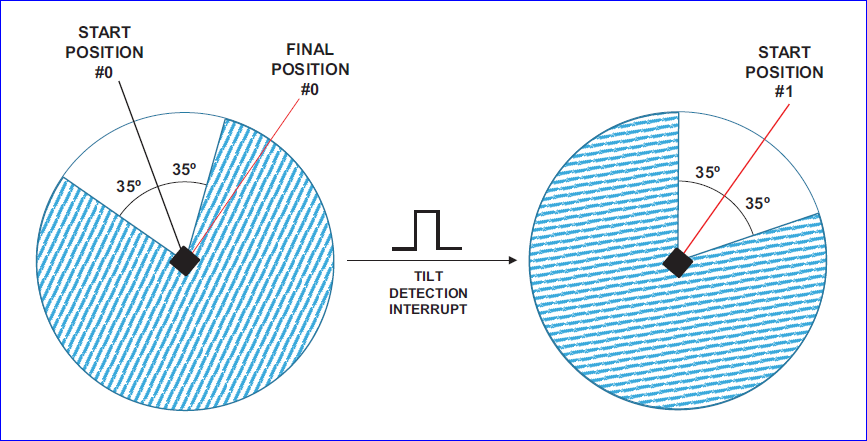
* 1. Expliquer à quoi correspond l’application « Tilt », ainsi que le principe de fonctionnement permettant de la mettre en œuvre.

**Cette application consiste à détecter une variation d’inclinaison (tilt) supérieure à ±35°. Lorsque cet évènement est détecté, le composant élabore une requête d’interruption.**

**L’exemple type donné par le constructeur consiste à détecter et comptabiliser le nombre de fois qu’une personne ayant son smartphone en poche se lève et s’assoit.**

**Dans cet exemple, il y aura élaboration d’une requête d’interruption lorsque « l’inclinaison » de la personne se stoppera en position finale 0.**

**Après une durée programmable, l’inclinaison actuelle (0) constituera l’inclinaison de départ (1) suivante.**



* 1. Expliquer à quoi correspond l’application « Timestamp », ainsi que le principe de fonctionnement permettant de la mettre en œuvre.

**L’application « Timestamp » (horodatage) consiste à mettre en œuvre un temporisateur programmable 24 bits qui s’incrémente d’une unité toutes les 6,4 ms en mode basse résolution ou toutes les 25 µs en mode haute résolution. Ce temporisateur programmable peut être réinitialisé à 0 à tout instant.**

**La durée que met le temporisateur pour compter depuis sa valeur minimale (0) jusqu’à sa valeur maximale (224-1) est donc égale à :**

* **6,4.10-3x224 soit 107374,1824 s en mode basse résolution ;**
* **25.10-6x224 soit 419,4304 s en mode basse résolution.**

**Le dispositif élabore par ailleurs une requête d’interruption 1,638 seconde avant que le temporisateur n’ait atteint sa valeur maximale, donc au bout d’une durée égale à 107 372,5444 s (soit un peu moins de 30 heures) en mode basse résolution et égale à 417,7924 s (soit un peu moins de 7 min) en mode haute résolution.**

**Je ne vois pas trop en quoi cela permet de faire de l’horodatage (timestamp) …**

**On remarquera, à titre de synthèse, que les 10 situations et applications que nous venons de décrire ne concernent que l’accéléromètre intégré au composant LSM6DS3, et en aucune façon le gyromètre.**

***La lecture des valeurs de Na et Nω en temps réel ou par paquets en temps différé***

Nous avons vu que le composant LSM6DS3 avait la capacité d’informer le µC, par élaboration d’une requête d’interruption, de la mise à disposition dans les registres de sortie OUT\_XL ou OUT\_G, d’une nouvelle valeur de Nax, de Nay, de Naz, de Nωx, de Nωy ou de Nωz.

Le programme est donc interrompu … très souvent, et ce, d’autant plus souvent que la valeur d’ODR est élevée (si la valeur d’ODR programmée est maximale et égale à 6664 Hz pour l’accéléromètre et 1666 Hz pour le gyromètre, le µC sera interrompu 3\*6664 + 3\*1666 ≈ 25000 fois par seconde).

Si l’on a besoin des valeurs de Na ou Nω en temps réel, on n’a pas le choix.

Mais si l’application ne nécessite pas un fonctionnement en temps réel (ce qui sera souvent le cas), le constructeur a prévu un tampon dans lequel seront mémorisées les valeurs de Na et Nω successives, et que le µC pourra venir lire quand le programme le jugera utile. Dans ce cas, les valeurs de Na et Nω seront lues et traitées, non pas les unes après les autres, au fur et à mesure qu’elles arrivent, mais par paquets.

Ce tampon est appelé « FIFO buffer » (voir explications pages 88 à 113/119).

* 1. Indiquer ce que signifie l’abréviation FIFO et donner l’abréviation de l’autre type de tampon (complémentaire) qu’il faut connaitre. Justifier la raison pour laquelle le constructeur a intégré une FIFO plutôt que l’autre type de buffer.

**FIFO signifie « First-In, First-Out », ce qui signifie que lorsqu’on vient lire le tampon (qu’on appelle bien souvent « buffer »), c’est la première donnée qui a été mémorisée dans le buffer qu’on lit, donc la plus ancienne.**

**A l’inverse, il existe des buffers LIFO (Last-In, First-Out) pour lesquels la donnée que l’on lit correspond à la dernière qui vient d’entrer (la plus récente).**

**Les buffers FIFO et LIFO sont des buffers série.**

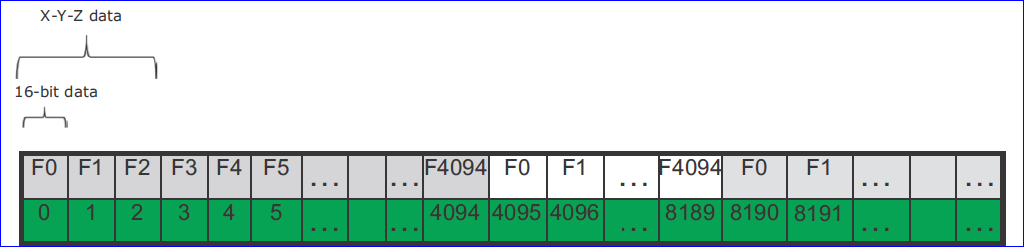
**Dans notre exemple, le constructeur a choisi une FIFO de façon que, lorsque le µC vient lire son contenu, il accède aux valeurs successives de Na et Nω des plus anciennes aux plus récentes, c’est-à-dire dans l’ordre chronologique. Si l’objectif consiste à caractériser l’évolution temporelle des valeurs de N, il est tout à fait naturel d’accéder à ces valeurs dans l’ordre chronologique.**

* 1. Expliquer de façon synthétique à quoi correspondent les modes de fonctionnement « Bypass », « FIFO », « Continuous », « Continuous-to-FIFO » et « Bypass-to-Continuous » auxquels le constructeur fait référence.

**Le constructeur précise que la taille de la FIFO est de 8 koctets, soit 8192 octets.**

**Les valeurs de Na et Nω sont codées sur 16 bits, donc sur deux octets consécutifs ; par conséquent, la mémorisation des 6 données Nax, Nay, Naz, Nωx, Nωy et Nωz occupera 12 cases mémoires successives de la FIFO.**

**Le constructeur précise l’organisation de la FIFO suivante :**



**En gris : l’adresse de chacune des 4095 cases mémoire de 16 bits de la FIFO**

**En vert : l’indice des échantillons N de 16 bits qui arrivent successivement dans la FIFO**

**On remarquera que la capacité de stockage de la FIFO est de 4095 mots de 16 bits, donc de 8190 octets et non de 8k octets.**

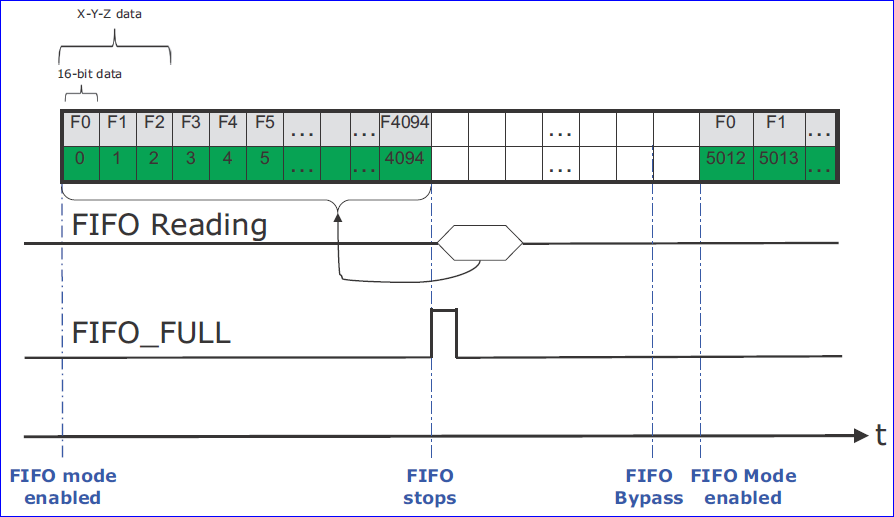
1. **Mode « Bypass » (page 96/119)**

**La FIFO n’est pas mise en œuvre, et son contenu est automatiquement initialisé à 0.**

1. **Mode « FIFO » (pages 96 et 97/119)**

**Dans ce mode, une fois la FIFO pleine, son « remplissage » cesse et les valeurs de N suivantes ne sont pas mémorisées (donc perdues).**

**Le bit indicateur « FIFO\_FULL » peut déclencher une requête d’interruption, et la fonction d’interruption pourra alors lire l’intégralité des 4095 dernières valeurs de N qui viennent d’être mesurées.**



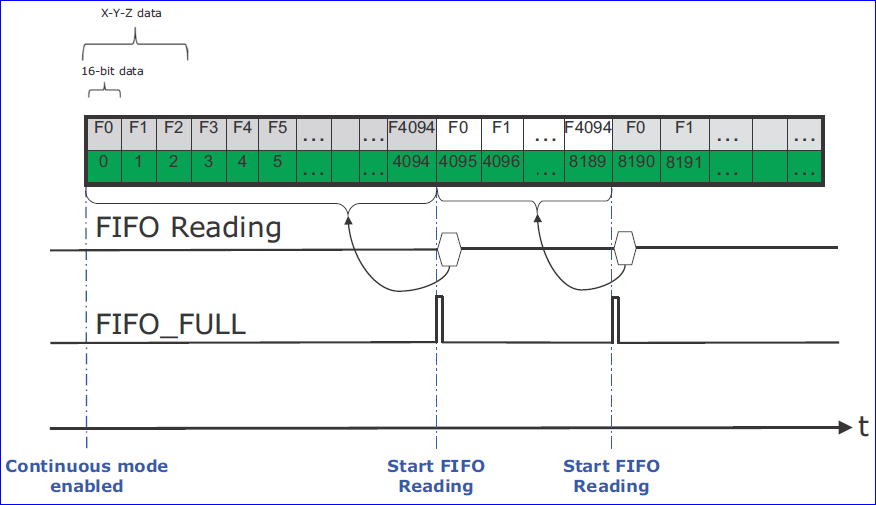
**Pour relancer la mémorisation de 4095 nouvelles valeurs de N successives, il faut au préalable effectuer une réinitialisation en configurant la FIFO en mode Bypass, puis basculer à nouveau en mode FIFO.**

**Ce mode pourrait se traduire en français, par mode déclenché.**

1. **Mode « Continuous » (pages 97 et 98/119)**

**Dans ce mode de fonctionnement, la mémorisation des valeurs successives de N dans la FIFO est permanente, et lorsque cette dernière est pleine, les valeurs de N suivantes écrasent les plus anciennes.**

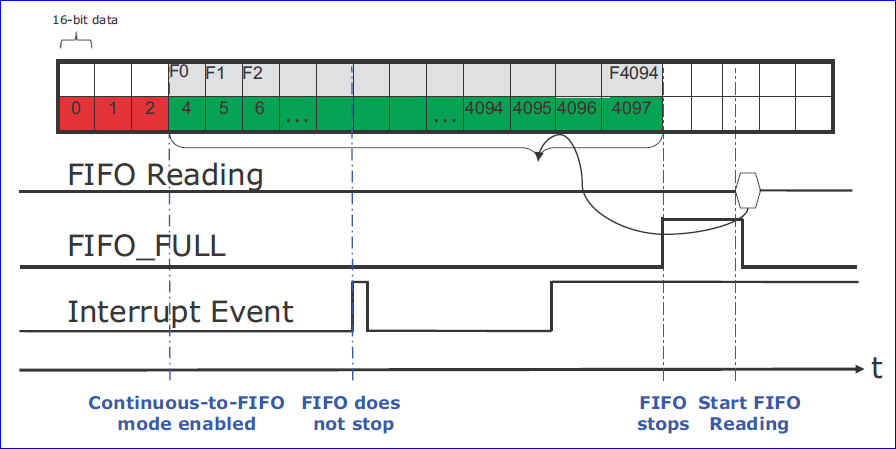
**Il est donc impératif, dans ce mode de fonctionnement, si l’on ne souhaite pas « perdre » de valeurs de N, de faire en sorte que le µC lise la totalité des 8190 octets en une durée inférieure à la durée séparant 2 mémorisations de N successives, c’est-à-dire inférieure à .**



**Ce mode pourrait se traduire en français par mode continu.**

1. **Mode « Continuous-to-FIFO » (pages 99 et 100/119)**

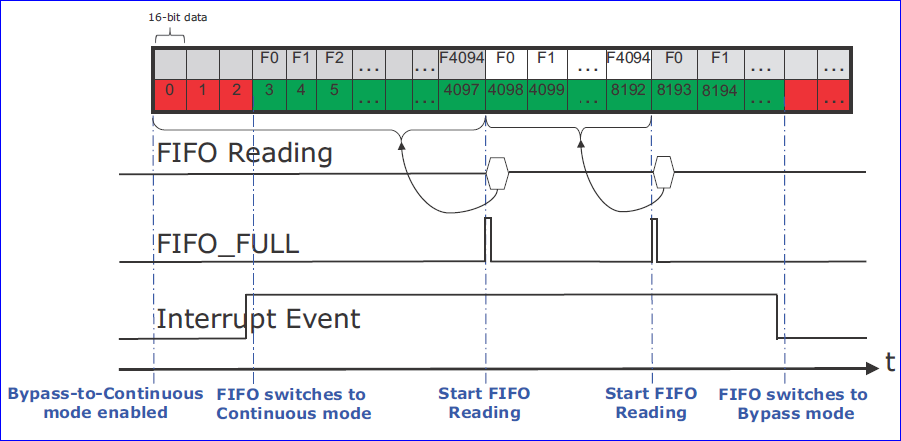
**Lorsqu’on sélectionne ce mode de fonctionnement, le composant commence par fonctionner en mode continu, et dès qu’un évènement déclenchant intervient, il passe automatiquement en mode FIFO ; cette dernière continue à se remplir jusqu’à ce qu’elle soit pleine, auquel cas le bit indicateur FIFO\_FULL signale que la FIFO est pleine :**



**La liste des évènements déclenchant est longue : significant motion, tilt, step detection, single tap, double tap, free-fall, wake-up ou 6D.**

1. **Mode « Bypass-to-Continuous » (pages 101 et 102/119)**

**Lorsqu’on sélectionne ce mode de fonctionnement, le composant commence par fonctionner en mode Bypass, et dès qu’un évènement déclenchant intervient, il passe automatiquement en mode Continuous, et y demeure aussi longtemps que l’évènement déclenchant est actif ; lorsque celui-ci n’est plus actif, le composant rebascule automatiquement en mode Bypass :**



1. **Le câblage du composant LSM6DS3 sur la carte End Device (schéma page 2 sur 9)**
   1. Justifier le câblage des broches d’alimentation Vdd et Vdd\_IO du composant LSM6DS3.

**Puisque la tension d’alimentation du µC est de 3,3 V, on peut connecter les deux broches Vdd et Vdd\_IO à ce potentiel (voir réponse à la question 4.3).**

* 1. Préciser à l’aide de quel type d’interface du µC ESP32 le concepteur de la carte End Device a décidé de piloter le composant LSM6DS.

**Le concepteur a choisi de piloter le composant LSM6DS3 à l’aide d’une interface I2C.**

* 1. Justifier le câblage des broches CS et SA0.

**La broche CS est connectée au niveau logique haut, ce qui permet de sélectionner l’interface de communication I2C (voir réponse à la question 4.4).**

**La broche SA0 est connectée au niveau logique haut, ce qui permet d’attribuer au composant, l’adresse 1101010 = 0x6A (voir réponse à la question 4.6).**

* 1. Justifier l’information « Adr I2C $6A » mentionnée sur le schéma structurel.

**Cette information indique l’adresse du composant (il s’agit bien de l’adresse 0x6A).**

* 1. Justifier la présence et le rôle des résistances R46 et R47.

**Les résistances R46 et R47 sont des résistances de Pull-Up (ou résistances de tirage au niveau haut, en français) dont le rôle consiste à forcer les lignes SDA et SCL au niveau haut en l’absence de communication.**

* 1. Indiquer, en considérant la présence de la résistance R45, à quel type de front (montant ou descendant) l’entrée d’interruption du µC GPIO33 (IRQ\_I2C) devra être sensible. Préciser alors la conséquence sur la configuration du composant LSM6DS3.

**Si le cavalier J45 est ouvert, la broche d’entrée d’interruption GPIO33 est forcée au niveau haut par R45, ce qui signifie qu’on ne pourra signifier au µC la présence d’une requête d’interruption, qu’en lui transmettant un front descendant.**

**Par conséquent, si l’on souhaite que le composant LSM6DS3 puisse élaborer une requête d’interruption auprès du µC, il faudra configurer la sortie d’interruption INT du composant pour qu’elle soit active sur un front descendant.**