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

**« Acquisition et mesure des informations de température, d’humidité et de pression »**

**CORRECTION**

**Objectif**: étudier la structure matérielle associée à la fonction « Acquisition et mesure des informations de température, d’humidité et de pression », afin d’être en mesure, ultérieurement, d’élaborer le programme permettant d’acquérir, de mesurer puis de transmettre à la passerelle, les informations de température, de pression atmosphérique et de taux d’humidité relative au voisinage de la carte End Device.

**Consignes** : avant de répondre aux questions de ce TD, lire ***attentivement*** la documentation du composant **BME280** (<https://cdn-shop.adafruit.com/datasheets/BST-BME280_DS001-10.pdf>) ***au moins deux fois*** dans son intégralité, en s’aidant du logiciel de traduction en ligne DeepL si nécessaire.

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



1. **Schéma structurel associé**

|  |
| --- |
|  |

1. **Présentation fonctionnelle du composant**
   1. Préciser, parmi les propositions qui suivent, à quel type de composant le BME280 appartient : capteur, microcontrôleur, actionneur.

**Il s’agit d’un capteur.**

* 1. Indiquer s’il s’agit d’un composant analogique ou numérique ou bien encore analogique et numérique.

**On peut dire qu’il s’agit d’un composant numérique dans la mesure où les valeurs significatives de la température, de la pression atmosphérique et du taux d’humidité relative sont transmises via une interface SPI ou I2C. Cela étant, les étages d’entrée (front end) ainsi que le multiplexeur qui suit sont de technologie analogique.**

* 1. Indiquer les paramètres physiques auxquels le composant est sensible (préciser les unités des grandeurs associées à ces paramètres physiques).

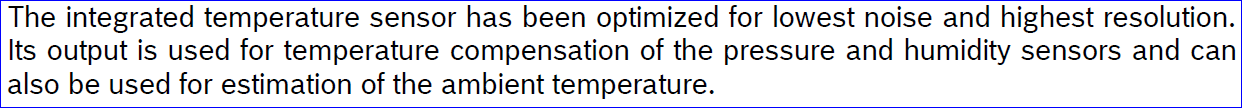
**La température en °C, la pression atmosphérique en Pa ou en mmHg et le taux d’humidité relative en %.**

**Pour information, 1 Pa = 7500.10-6 mmHg (ou 1 mmHg = 133,322 Pa).**

* 1. Indiquer la plage de variation possible de chacun de ces paramètres.
* **Taux d’humidité relative : de 0 à 100% (page 8)**
* **Pression atmosphérique : de 300 à 1100 hPa (page 8) ou de 225 à 825 mmHg.**

Le constructeur indique en titre « **Combined humidity and pressure sensor** ».

Par ailleurs, il précise page 3 :



* 1. Indiquer la raison pour laquelle le constructeur ne fait pas mention de la température dans le titre, et la raison pour laquelle il indique que le composant ne permet d’effectuer qu’une estimation de la température ambiante.

**Les valeurs de pression atmosphérique et de taux d’humidité relative mesurées par le capteur, dépendent de la température. Le capteur de température a précisément pour fonction de compenser en température les capteurs de pression atmosphérique et de taux d’humidité relative.**

**Par ailleurs, le capteur de température se trouve à l’intérieur d’un boitier métallique, donc la température qui règne à l’intérieur, compte tenu de l’échauffement thermique, est forcément supérieure à la température ambiante. Le mesure de température ne permet donc de faire qu’une une estimation (par excès) de la température ambiante.**

* 1. Expliquer à quoi correspondent les 3 modes de fonctionnement *sleep mode*, *normal mode* et *forced mode*, et proposer une traduction française « parlante » à chacun de ces modes.
* **Sleep mode : le composant est inactif 🡪 mode inactif en français**
* **Normal mode : le composant effectue des mesures périodiques 🡪 mode périodique**
* **Forced mode : le composant effectue des mesures à la demande 🡪 mode déclenché**
  1. Indiquer, en mode normal, les différentes valeurs de tstandby possibles.

**0,5/10/20/62,5/125/250/500/1000 ms (page 28).**

1. **Les caractéristiques du composant BME280**

*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 VDDIO (pages 1, 7 et 11) :**

* **VDD constitue l’alimentation interne du composant**
* **VDDIO constitue l’alimentation de l’étage de sortie du composant (interfaces SPI/I2C)**
  1. Préciser les valeurs d’alimentation possibles que l’on peut appliquer sur ces broches.
* **VDD : de 1,71V à 3,6V**
* **VDDIO : de 1,2V à 3,6V**
  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 1,71V et 3,6V, on reliera les deux broches d’alimentation VDD et VDDIO du composant ensemble que l’on connectera à la tension d’alimentation du µC.**

**Par contre, si la tension d’alimentation du µC est comprise entre 1,2V et 1,71V, on connectera la broche VDD à n’importe quelle valeur de tension comprise entre 1,71V et 3,6V, et la broche VDDIO à la tension d’alimentation du µC.**

* 1. Indiquer les valeurs numériques typiques et maximales des courants IDDSL, IDDSB, IDDH, IDDT et IDDP (voir figure 4 page 13), considérés à une température ambiante de 25°C.
* **IDDSLtyp = 0,1 µA IDDSLmax = 0,3 µA (page 7)**
* **IDDSBtyp = 0,2 µA IDDSBmax = 0,5 µA (page 7)**
* **IDDHtyp = 1,8 µA IDDHmax = 2,8 µA (page 8)**
* **IDDTtyp = 1 µA IDDTmax non spécifiée (page 10)**
* **IDDPtyp = 2,8 µA IDDPmax = 4,2 µA (page 9)**
  1. Préciser ce que signifie l’indication « *1 Hz forced mode* » donnée dans les tables 2 à 4.

**Cette expression signifie que les mesures s’effectuent en mode déclenché, mais de façon répétitive, avec une fréquence de répétition de 1 Hz, soit au rythme d’une mesure toutes les secondes.**

*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 l’instant de mise sous tension (page 30) : si à l’instant de mise sous tension, la broche CSB est reliée au potentiel VDDIO, alors c’est l’interface I2C qui est active, sinon (c’est-à-dire, si la broche CSB est connectée à la masse), c’est l’interface SPI qui est validée.**

* 1. Proposer, pour chacune des deux interfaces, le schéma de câblage du composant BME280 associé à un microcontrôleur alimenté sous Vcc = 3,3V (on ne prendra pas en compte l’interface SPI 3-Wire).



**Connexion via une interface I2C**



**Connexion via une interface SPI**

**Ce qui peut être troublant, pour ce qui concerne le schéma de connexion via une interface I2C, est le fait que la broche SDO est une broche d’entrée, d’une part, et que la broche SDI constitue, lors d’une opération de lecture, une sortie, d’autre part.**

**D’autres constructeurs auraient nommé ces deux broches respectives A0/SDO et SDA/SDI.**

* 1. Préciser, en utilisant 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 111011x où x correspond au niveau logique de la tension appliquée sur la broche SDO (page 31). Le composant comporte donc 2 adresses distinctes selon le niveau logique de la tension appliquée sur la broche SDO : si cette dernière est connectée à la masse, l’adresse est égale à 1110110, soit 0x76, et si elle est connectée au potentiel VDDIO, son adresse sera égale à 1110111, soit 0x77.**

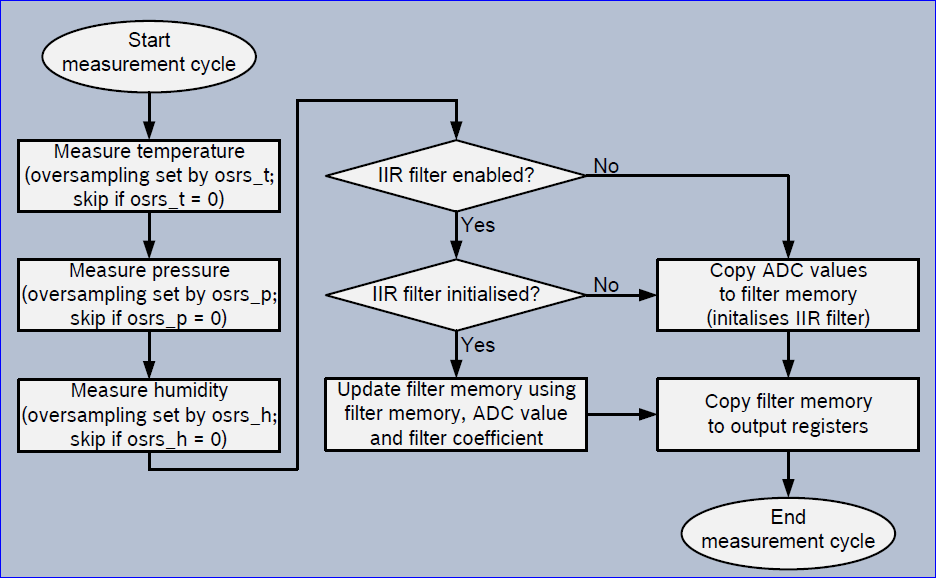
* 1. Indiquer, pour chacune des interfaces de communication, la valeur maximale de la fréquence d’horloge.

**Pour ce qui concerne l’interface SPI, fSCKmax = 10 MHz (page 36) et pour l’interface I2C, fSCKmax = 3,4 MHz (page 2).**

**Remarque personnelle : j’ai des doutes sur cette valeur de 3,4 MHz, si je considère (page 35) l’information selon laquelle la période minimale, en considérant des fronts montant et descendant instantanés, est égale à (160+210) ns, soit une fréquence maximale de 2,7 MHz.**

*Processus de mesure des paramètres physiques*

Le processus de mesure, proposé par le constructeur, est rappelé ci-dessous :



On s’attache, dans ce qui suit, à démystifier les notions d’« *oversampling* » et de « *IIR filter* ».

Le constructeur indique successivement, aux paragraphes 3.4.1 à 3.4.3 :

« *For the humidity, oversampling is possible to reduce the noise.* »,

« *For the pressure, oversampling is possible to reduce the noise.* »,

« *For the temperature, oversampling is possible to reduce the noise.* ».

Il indique, par ailleurs, sur le synoptique de la page précédente, que les différentes valeurs possibles de « *oversampling* » sont fixées par les paramètres notés respectivement *osrs\_h*, *osrs\_p* et *osrs\_t*.

* 1. Indiquer les différentes valeurs pouvant être affectées aux paramètres *osrs\_h*, *osrs\_p* et *osrs\_t*.

**Chacun des 3 paramètres osrs\_h, osrs\_p et osrs\_t peut prendre les valeurs 0, 1, 2, 3, 4 et 5 ou plus, correspondant respectivement à « pas d’oversampling », « oversampling de 1, de 2, de 4, de 8 ou de 16 (Table 20, page 26 et Tables 23 et 24, page 27).**

* 1. Préciser dans quel sens varie le bruit en fonction des valeurs de *osrs\_h*, *osrs\_p* et *osrs\_t*, et en déduire la fonction réalisée par ce dispositif d’« *oversampling* ».

**Plus la valeur de osrs\_h, de osrs\_p ou de osrs\_t augmente, plus la valeur d’« oversampling » augmente également, et plus le bruit diminue (Tables 11 et 12, page 19 et Table 14, page 20).**

* 1. Préciser dans quel sens varie le courant consommé par le composant en fonction des valeurs de *osrs\_h*, *osrs\_p* et *osrs\_t* et justifier la cohérence de la réponse.

**Plus la valeur de osrs\_h, de osrs\_p ou de osrs\_t augmente, plus la valeur d’« oversampling » augmente également, et plus le courant consommé augmente (Tables 11 et 12, page 19).**

**Puisque ces paramètres osrs\_h, osrs\_p et osrs\_t ont pour effet de diminuer le bruit, ils agissent comme un filtre passe-bas. Et si leur présence a pour effet d’augmenter la consommation de courant, cela signifie que le temps de mesure tmeasure augmente (Figure 4, page 13 et Figure 5, page 14). On peut donc supposer à ce stade, que le paramètre « oversampling » correspond au nombre de fois (2, 4, 8 ou 16 fois) où la mesure est effectuée, et que dans ces conditions, la mesure délivrée correspond à la moyenne des 2, 4, 8 ou 16 mesures effectuées.**

**Le dispositif d’« oversampling » reviendrait donc à réaliser un filtre moyenneur.**

* 1. Préciser dans quel sens varie le temps de mesure du paramètre physique en fonction des valeurs de *osrs\_h*, *osrs\_p* et *osrs\_t* et justifier la cohérence de la réponse.

**Les formules données paragraphe 9.1 page 51 montrent clairement que le temps de mesure tmeasure (voir Figure 4, page 13 et Figure 5, page 14) augmente avec la valeur d’« oversampling », ce qui vient confirmer ce que l’on a supposé à la question précédente.**

**Il n’y a plus de doute possible, ce que le constructeur appelle très maladroitement (selon moi) « oversampling » et qu’on traduit en français par suréchantillonnage, est tout simplement un filtre moyenneur dont on peut programmer le nombre de mesures successives (2, 4, 8 ou 16) à partir desquelles est calculée la valeur moyenne.**

Le constructeur indique la possibilité d’insérer, en sortie du DAC, un filtre numérique de type IIR.

* 1. Rappeler ce qu’est un filtre IIR et donner son appellation en français.

**IIR signifie Infinite Impulse Response. Il s’agit donc d’un filtre à Réponse Impulsionnelle Infinie (filtre RII).**

* 1. Donner, en anglais et en français, l’appellation de l’« autre type » de filtre numérique.

**L’autre type de filtre est le filtre FIR (Finite Impulse Response) ou Filtre à Réponse Impulsionnelle Finie (filtre RIF).**

* 1. Préciser l’équation de récurrence du filtre IIR en donnant les différentes valeurs de « *filter\_coefficient* » possibles dont il est fait mention dans la documentation.

**(page 15), où « filter\_coefficient » peut prendre les valeurs 2, 4, 8 ou 16 (Table 6, page 16).**

* 1. Préciser dans quel sens varie le bruit associé au paramètre pression en fonction des valeurs du « *filter\_coefficient* », et en déduire la fonction réalisée par ce dispositif.

**Plus la valeur de filter\_coefficient augmente et plus le bruit sur la mesure de pression atmosphérique diminue (Table 12, page 19). Ce dispositif joue donc bien un rôle de filtre.**

* 1. Ecrire l’équation de récurrence du filtre sous forme sa canonique (les échantillons seront notés e(n) et s(n) et les coefficients a pour e(n) et b pour s(n-1)). Préciser alors s’il s’agit d’un filtre récursif ou non récursif et indiquer les différentes valeurs possibles des coefficients a et b.

**avec**

**Il s’agit d’un filtre récursif pour lequel les coefficients a et b peuvent, en fonction des valeurs de « filter\_coefficient » possibles, prendre les 4 couples de valeurs suivants :**

**(a = ½ et b = ½), (a = 1/4 et b = ¾), (a = 1/8 et b = 7/8), (a = 1/16 et b = 15/16).**

* 1. Retrouver, en considérant le graphe de la figure 7, le type de filtre réalisé (passe-bas, passe-haut, …) et en déduire son ordre.

**Le graphe de la figure 7 représente la réponse du filtre à un échelon pour les différentes valeurs de « filter\_coefficient » possibles. On constate que plus la valeur de « filter\_coefficient » augmente, plus la constante de temps τ du filtre augmente également.**

**Dans tous les cas, on obtient la réponse d’un filtre passe-bas du 1er ordre.**

Le constructeur définit la grandeur *ODR* (Appendix B).

* 1. Indiquer ce que signifie l’abréviation ODR, justifier les valeurs proposées dans la documentation et proposer une traduction en français.

**ODR signifie, selon le constructeur, Output Data Rate.**

**Le constructeur précise, selon le mode utilisé (mode déclenché ou mode périodique), que (mode déclenché) ou (mode périodique), sans préciser que tmeasure et tstandby sont exprimés en ms (page 51).**

**Une fois qu’on a remarqué cela, on en déduit que cette grandeur correspond au nombre de mesures par seconde (c’est bien une unité de débit (rate) comme l’indique le constructeur).**

**Le constructeur préfère exprimer cette grandeur en Hz, ce qui n’est pas faux, et ceci vient conforter la réponse à la question 4.5.**

1. **La programmation du composant BME280**

La documentation indique, Table 18, page 25, que le composant présente 56 registres internes.

* 1. Indiquer la fonction du registre à lecture seule « *id* ».

**Le registre « id » d‘adresse 0xD0 contient l’identifiant du composant, égal à 0X60 (page 25).**

* 1. Indiquer la fonction du registre à écriture seule « *reset* ».

**Le registre « reset » d’adresse 0xE0 permet, lorsqu’on l’initialise avec la valeur 0xB6, de réinitialiser le composant (page 25).**

* 1. Identifier les 3 registres de configuration du composant, et indiquer pour chacun d’entre eux et par une phrase simple, la fonction de chacun des bits ou groupes de bits associés.

**Il s’agit des registres (page 25) « ctrl\_hum » (adresse 0xF2), « ctrl\_meas » (adresse 0xF4) et « config » (adresse 0xF5).**

**Registre « ctrl\_hum » : les 3 bits de poids faibles (b0 à b2), notés osrs\_h[2:0] permettent de configurer le nombre de mesures de taux d’humidité relative effectuées successivement et à partir desquelles est calculée la valeur moyenne du taux d’humidité relative (page 26).**

**Registre « ctrl\_meas » :**

* **les bits b0 et b1, notés mode[1:0], permettent de configurer le composant en mode inactif, déclenché ou périodique (page 27) ;**
* **les bits b2 à b5, notés osrs\_p[2:0] permettent de configurer le nombre de mesures de pression atmosphérique effectuées successivement et à partir desquelles est calculée la valeur moyenne de la pression atmosphérique (page 27) ;**
* **les bits b6 à b7, notés osrs\_t[2:0] permettent de configurer le nombre de mesures de température effectuées successivement et à partir desquelles est calculée la valeur moyenne de la température (page 27).**

**Registre « config » :**

* **le bit b0, noté spi3w\_en[0] (page 28) permet de sélectionner le type d’interface SPI (version classique à 4 fils ou variante à 3 fils) ;**
* **les bits b2 à b4, notés filter[2:0], permettent de fixer la constante de temps du filtre numérique IIR passe-bas du 1er ordre (Table 28, page 28) ;**
* **les bits b5 à b7, notés t\_sb[2:0], permettent de fixer la valeur de la durée tstandby lorsque le composant est configuré en mode périodique (Table 27, page 28).**
  1. Identifier le registre d’état du composant et indiquer par une phrase simple, la fonction des deux bits d’état associés.

**Le registre d’état noté « status », d’adresse 0xF3, comporte deux bits accessibles en lecture :**

* **le bit b0, noté im\_update[0], permet d’informer le programmeur si une mise à jour des registres de sortie est en cours ou non (page 26) ;**
* **le bit b3, noté measuring[0], permet d’informer le programmeur si une mesure est en cours ou non (donc terminée) (page 26).**
  1. Identifier les 3 registres de sortie contenant l’information significative de la pression, et préciser sur combien de bits cette dernière est codée.

**Il s’agit des registres (p 25) « press\_msb » (adresse 0xF7), « press\_lsb » (adresse 0xF8) et « press\_xlsb » (adresse 0xF9), qui contiennent l’information significative de la pression atmosphérique mesurée (page 29).**

**Cette information est codée sur 20 bits (seuls les bits b4 à b7 du registre « press\_xlsb » sont significatifs).**

* 1. Identifier les 3 registres de sortie contenant l’information significative de la température, et préciser sur combien de bits cette dernière est codée.

**Il s’agit des registres (p 25) « temp\_msb » (adresse 0xFA), « temp\_lsb » (adresse 0xFB) et « temp\_xlsb » (adresse 0xFC), qui contiennent l’information significative de la température mesurée (page 29).**

**Cette information est codée sur 20 bits (seuls les bits b4 à b7 du registre « temp\_xlsb » sont significatifs).**

* 1. Identifier les 2 registres de sortie contenant l’information significative de l’humidité, et préciser sur combien de bits cette dernière est codée.

**Il s’agit des registres (p 25) « hum\_msb » (adresse 0xFD) et « hum\_lsb » (adresse 0xFE), qui contiennent l’information significative du taux d’humidité relative mesuré (page 29).**

**Cette information est codée sur 16 bits.**

Une fois la mesure effectuée, les 8 registres compris aux adresses allant de 0xF7 à 0xFE contiennent les informations significatives de la pression, de la température et de l’humidité.

Ces informations « brutes » doivent être « travaillées » par logiciel, afin d’extraire les valeurs de pression en Pa, de température en °C et d’humidité relative en %.

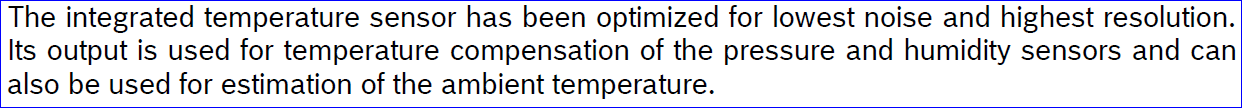
Cette phase calculatoire, que l’on n’étudiera pas, est donnée par le constructeur au paragraphe 4.2.3.

Pour cela, le code accède à des valeurs notées *dig\_T*, *dig\_P* *et dig\_H*.

* 1. Indiquer à quel endroit sont stockées ces valeurs.

**Ces valeurs sont stockées dans les 42 registres de calibration positionnés aux adresses allant de 0x88 à 0xA1 puis de 0xE1 à 0xF0 (page 25).**

On revient à l’indication récupérée de la documentation donnée en page 1 de ce TD :



* 1. Expliquer ce que signifie « *temperature compensation of the pressure and humidity sensors* » et repérer l’élément de programme du paragraphe 4.2.3 permettant d’illustrer le phénomène décrit ci-dessus en anglais.

**Il s’agit de compenser en température, les capteurs de pression et d’humidité.**

**Dans le code proposé pages 23 et 24, on note la présence d’une variable globale notée t\_fine, qui contient l’information significative de la température mesurée. On note également que cette variable t\_fine est utilisée par :**

* **la fonction BME280\_S32\_t BME280\_compensate\_T\_int32(BME280\_S32\_t adc\_T) permettant de calculer la température mesurée et exprimée en °C ;**
* **la fonction BME280\_U32\_t BME280\_compensate\_P\_int64(BME280\_S32\_t adc\_P) permettant de calculer la pression atmosphérique mesurée et exprimée en Pa ;**
* **la fonction BME280\_U32\_t bme280\_compensate\_H\_int32(BME280\_S32\_t adc\_H) permettant de calculer le taux d’humidité relative mesuré et exprimé en %.**

**On voit donc que cette valeur t\_fine est utilisée pour le calcul de la pression atmosphérique et du taux d’humidité relative, ce qui confirme que le capteur de pression atmosphérique et de taux d’humidité relative BME280 est *compensé en température*.**

* 1. Indiquer dans quelles variables seront finalement mémorisées les valeurs de température, de pression atmosphérique et de taux d’humidité relative.

**La température mesurée sera mémorisée dans la variable T, la pression atmosphérique mesurée dans la variable p, et le taux d’humidité relative dans la variable spécifiée par l’instruction d’appel de la fonction bme280\_compensate\_H\_int32(BME280\_S32\_t adc\_H).**

* 1. Indiquer, pour chacun des 3 paramètres, la relation numérique liant la valeur mémorisée à la valeur exprimée en °C, en Pa et en %.

**La température exprimée en °C sera égale au contenu de la variable T divisé par 100.**

**La pression atmosphérique en Pa sera égale au contenu de la variable p divisé par 256.**

**Le taux d’humidité relative en % sera égal à la valeur retournée la fonction divisée par 1024.**

Application

On souhaite effectuer une mesure de paramètres selon la configuration « Gaming » (table 10).

* 1. Proposer l’ordinogramme de l’application.

**On peut proposer l’ordinogramme suivant :**



**La fonction *init()* a pour rôle d’initialiser l’interface SPI ou I2C (selon le choix) du µC, et d’initialiser le composant BME280 conformément au cahier des charges « gaming » proposé (Table 10, page 18), c’est-à-dire, d’initialiser :**

* **le registre ctrl\_hum avec la valeur 0x00 (pas de mesure d’humidité) ;**
* **le registre ctrl\_meas avec la valeur 0x4F (mode périodique / tempx1 / pressionx4) ;**
* **le registre config avec la valeur 0x91 (tstandby = 0,5 ms), filter\_coefficient = 16) lorsqu’on choisit de piloter le composant BME280 à l’aide d’une interface SPI à 3 fils, sinon la valeur 0x90.**

**Une fois les grandeurs significatives des informations de température, de pression atmosphérique et de taux d’humidité relative récupérées (fonction *lecture\_TPH()*), le programme exécute la fonction *compensation()* permettant de calculer la valeur de la pression atmosphérique (en mode gaming, on n’a pas besoin de connaitre la valeur du taux d’humidité relative), qui est ensuite transférée à la fonction *application()*.**

**Si la durée d’exécution des fonctions [*lecture\_TPH()* + *compensation()* + *application()*]** **est supérieure à la durée (tmeasure + tstandby), alors l’application « ratera » des mesures, sinon, elle lira plusieurs fois la même mesure (il n’y a pas de mécanisme d’interruption permettant de connaitre l’instant où une nouvelle mesure vient de se produire).**

**A chaque fois qu’une série de 3 mesures (T, P et H) est effectuée, les résultats de mesure sont automatiquement transférés dans des registres fantômes (page 21). Puis, lorsque le programme termine la lecture des registres de sortie, c’est-à-dire, dans notre exemple, à la fin de l’exécution de la fonction *lecture\_TPH()*, le composant BME280 transfère automatiquement le contenu des registres fantômes dans les registres de sortie auxquels le programme aura accès lors de la prochaine exécution de la fonction *lecture\_TPH()*.**

**Ce mécanisme permet de gérer le caractère asynchrone de la lecture des résultats de mesure par rapport aux mesures proprement dites.**

1. **Le câblage du composant BME280 sur la carte End Device (schéma page 2 sur 5)**
   1. Justifier le câblage des broches d’alimentation VDD et VDDIO du composant BME280.

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

* 1. Préciser, lorsque le cavalier J15 est fermé, à l’aide de quel type d’interface du µC ESP32 le concepteur de la carte End Device a décidé de piloter le composant BME280.

**Le concepteur a choisi de piloter le composant BME280 à l’aide d’une interface SPI « classique » (4 fils).**

* 1. Justifier la présence de la résistance R31.

**Pour valider l’interface SPI du BME280, il faut que pendant la phase de mise sous tension de ce dernier, la broche CSB soit portée au niveau bas : c’est précisément la fonction de la résistance de pull-down R31.**

**Par contre, une fois l’initialisation du composant terminée, la broche CSB est positionnée au niveau bas (lorsqu’il y a communication entre le composant et le µC) ou au niveau haut (lorsqu’il n’y a pas communication entre le composant et le µC) par la broche de sortie /SS de l’interface SPI du µC.**

* 1. Justifier la présence de J15.

**L’interface SPI du µC associée aux 4 broches GPIO5, GPIO18, GPIO19 et GPIO23 pilote le composant BME280. Les signaux SCK, MOSI, MISO et /SS associés à cette interface SPI sont également transmis au connecteur d’extension J6, et il est donc possible d’équiper la carte fille positionnée sur les connecteurs J6 à J9 d’un composant doté d’une interface SPI.**

**Mais dans ce cas, pour éviter tout conflit, il faut pouvoir déconnecter le composant BME280 du bus SPI, ce qui sera fait en ouvrant le cavalier J15.**

**On retiendra qu’on ne pourra pas, dans une application, rajouter un composant équipé d’une interface SPI si on utilise par ailleurs le composant BME280.**