



MINISTÈRE
DE L'ÉDUCATION
NATIONALE

EAE SIN 2

SESSION 2018

**AGREGATION
CONCOURS EXTERNE**

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

**Option : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR
ET INGÉNIERIE INFORMATIQUE**

**MODÉLISATION D'UN SYSTÈME, D'UN PROCÉDÉ
OU D'UNE ORGANISATION**

Durée : 6 heures

Calculatrice électronique de poche - y compris calculatrice programmable, alphanumérique ou à écran graphique – à fonctionnement autonome, non imprimante, autorisée conformément à la circulaire n° 99-186 du 16 novembre 1999.

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout autre matériel électronique est rigoureusement interdit.

Dans le cas où un(e) candidat(e) repère ce qui lui semble être une erreur d'énoncé, il (elle) le signale très lisiblement sur sa copie, propose la correction et poursuit l'épreuve en conséquence.

De même, si cela vous conduit à formuler une ou plusieurs hypothèses, il vous est demandé de la (ou les) mentionner explicitement.

NB : La copie que vous rendrez ne devra, conformément au principe d'anonymat, comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé comporte notamment la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de signer ou de l'identifier.

Tournez la page S.V.P.

A

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie.

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

Concours	Section/option	Epreuve	Matière
EAE	1417A	109	2680

Diagnostic automatique des troubles du sommeil



(photo source Cleveland Medical Devices Inc.)

Le sujet comporte une introduction au système ainsi que cinq parties indépendantes de questionnement. Les annexes comprennent six documents techniques et un document réponse.

Présentation du système étudié

Le syndrome d'apnées du sommeil a de lourdes conséquences dans le comportement et la santé des personnes atteintes. Outre le manque de sommeil directement responsable des somnolences et autres troubles de l'attention, les conséquences sont également profondes pour les organes internes tels que le cœur et le cerveau.

Cinq stades d'endormissement sont habituellement définis :

- l'éveil, dénommé « W » (Wakefulness) ;
- le sommeil paradoxal « R » (REM – Rapid Eye Movements) ;
- le sommeil lent « N » (NREM) ;
- le sommeil lent léger subdivisé en stades « N1 » (NREM 1) et « N2 » (NREM 2) ;
- le sommeil lent profond « N3 » (NREM 3).

Les symptômes caractéristiques d'une apnée du sommeil sont :

- la perturbation de la respiration impliquant un manque temporaire d'oxygénation ;
- des phases de micro-éveils pendant lesquelles certains organes sont actifs (yeux, jambes...) ;
- des cycles de sommeils déréglés avec notamment l'absence de certaines phases d'endormissement ;
- un battement cardiaque non caractéristique du sommeil.

Afin de diagnostiquer précisément l'origine de ces troubles, le sommeil du patient peut être instrumenté au cours du temps (enregistrement polysomnographique). La figure 1 est un tracé au cours du temps des signaux de plusieurs capteurs spécifiques :

- électroencéphalogrammes en plusieurs coordonnées crâniennes (C3A2, O1A2, C4A1) ;
- mouvements de l'œil gauche (LEOG), de l'œil droit (REOG) ;
- électromyographies (EMG, relevé de l'activité électrique des muscles) du menton (chin EMG) et de la jambe (leg EMG) ;
- électrocardiogramme (ECG 1 et ECG 2) ;
- débit d'air (AirFlow), efforts thoraciques en plusieurs positions (TEffort et AEffort) et ronflement (Snore) ;
- saturation de l'hémoglobine en oxygène (SPO₂).

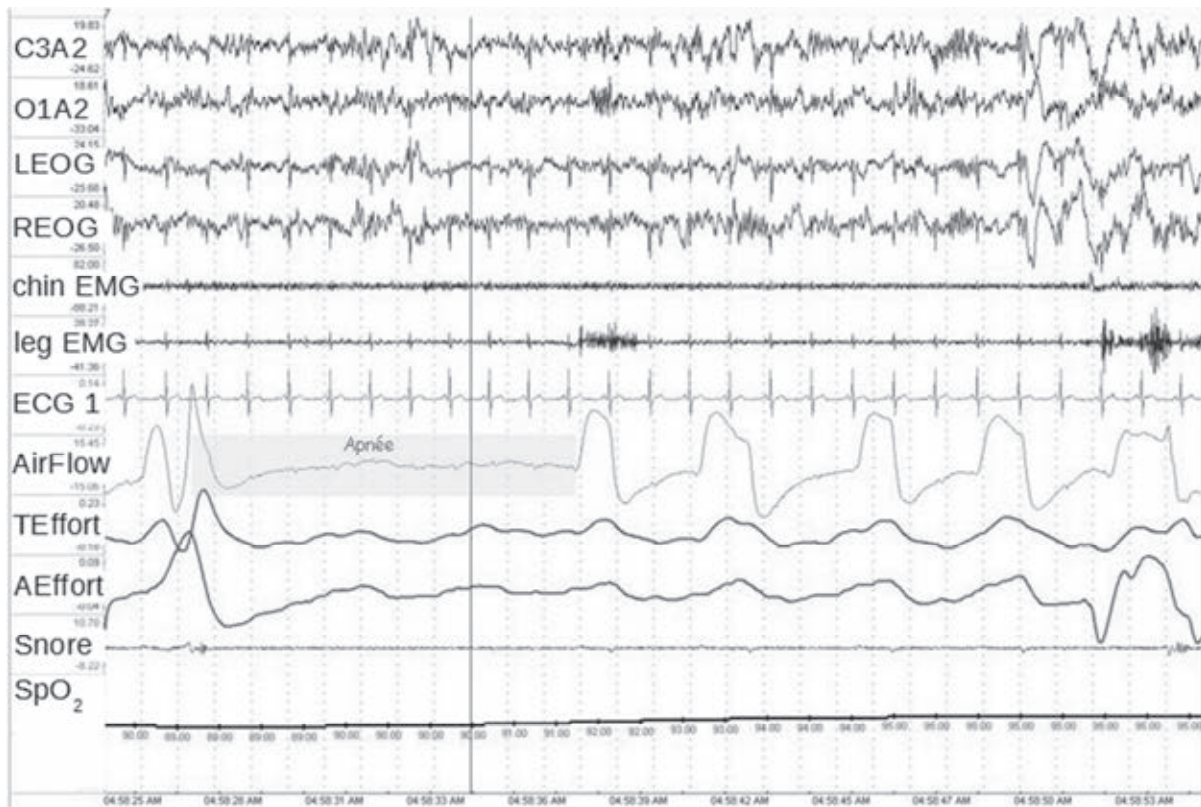


Figure 1 : relevés de quelques signaux du sommeil comportant une phase d'apnée

Actuellement, l'analyse et la corrélation des différents signaux nécessitent un examen détaillé par un ou plusieurs praticiens expérimentés. Cet examen long a un coût important mais repose sur des tâches très répétitives. Elles visent principalement à repérer dans la grande quantité de données enregistrées les signaux caractéristiques des différents troubles. L'objectif souhaité est un diagnostic automatisé repérant les différentes phases de sommeil au cours du temps.

Les méthodes d'identification des données numériques issues de différents capteurs font l'objet de plusieurs travaux de recherche dans le domaine de la fusion symbolique, une méthode de résolution de problèmes basée sur des règles de décisions. Un sous-ensemble de ces règles est listé dans le document technique DT1. Les données sont préalablement analysées afin d'extraire certaines grandeurs caractéristiques (fréquences propres, amplitudes...). Ces différentes caractéristiques sont ensuite combinées entre elles afin d'être comparées aux schémas habituels constitutifs des différentes phases d'endormissement.

Afin de déterminer les différents niveaux de sommeil, le scoring des stades de sommeil consiste à identifier, par analyse visuelle des courbes neurologiques (EEG, EOG, EMG, Menton), le stade de sommeil d'une portion de signal. L'AASM (American Academy of Sleep Medicine) demande de procéder à une analyse par fenêtres de 30 secondes,

appelées époques. Les règles de fusion s'inspirent des recommandations médicales à utiliser pour le diagnostic du syndrome d'apnées du sommeil.

À titre d'exemple, la figure 2 représente les règles pour tester le stade REM d'une époque. D'autres règles de calculs permettent de caractériser les autres stades de sommeil. Les règles de fusion symbolique sont indiquées sur le document technique DT1.

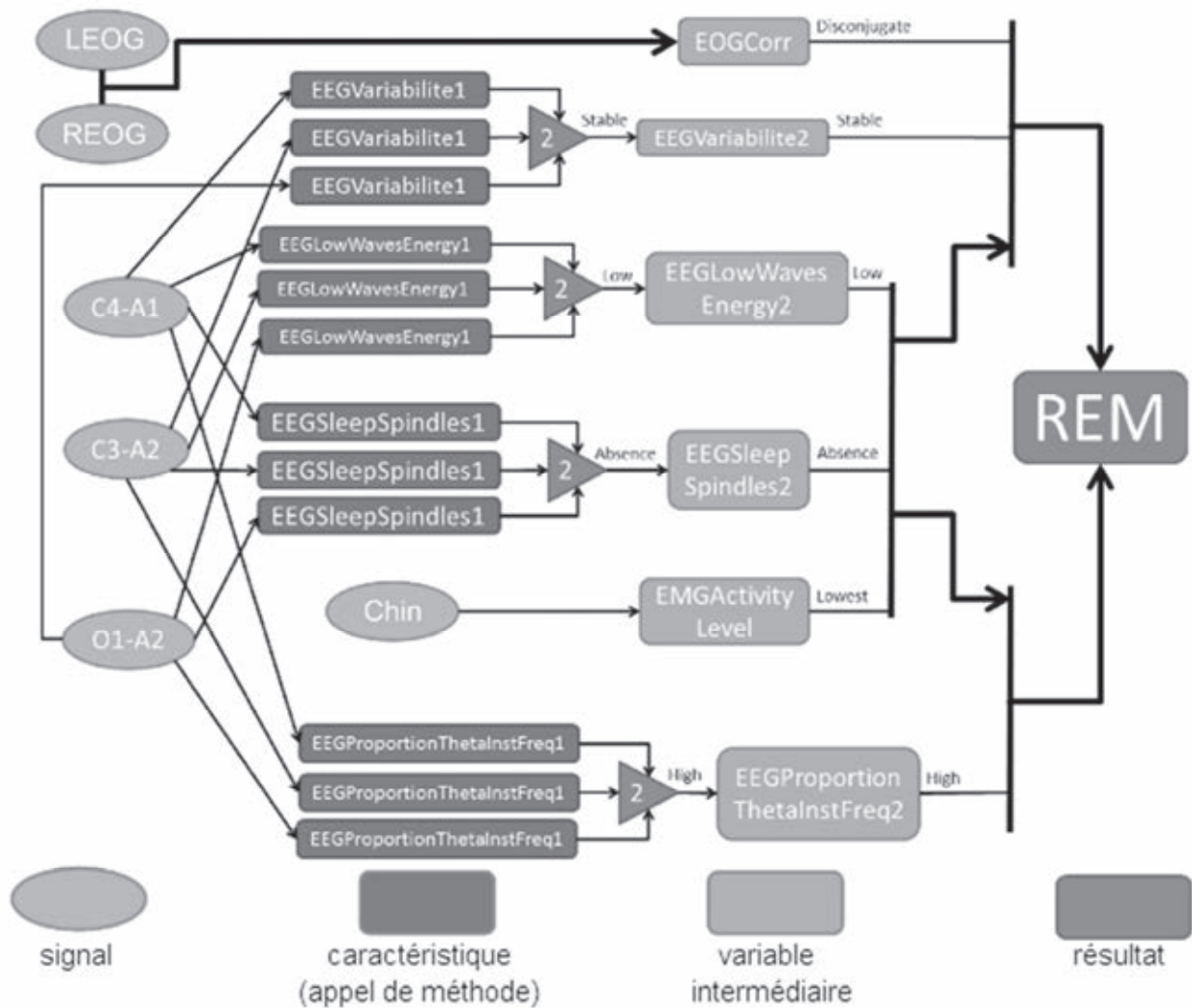


Figure 2 : diagramme issu d'un consensus médical pour valider le stade REM

L'ensemble des contraintes sur ce système sont indiquées dans le diagramme des exigences du document technique DT2. Expérimentalement, ce système est constitué de trois parties décrites dans le diagramme de définition de blocs du document technique DT3 : l'« unité patient » est un dispositif autonome relevant les signaux des différents capteurs dédiés. Cet enregistrement, généralement sur une nuit entière, est ensuite lu et analysé par l'« unité praticien » sur une station de travail automatisant le diagnostic. Enfin, le « VPS » (« Virtual Private Server ») réalise le stockage des données patient.

Partie 1. Analyse du fonctionnement du système

Objectifs : analyser le fonctionnement du système décrit par plusieurs diagrammes SysML, déterminer ses principales caractéristiques.

Le système est décrit fonctionnellement dans les documents techniques DT2, DT3 et DT4.

Sous-partie 1.1. Transfert d'informations entre l' « unité patient » et l' « unité praticien »

Chaque unité dispose d'un stockage local des données et d'interfaces de communications. Ces deux unités fonctionnent indépendamment sauf pendant la phase de transfert, limitée dans le temps par confort pour le praticien.

Question 1. À l'aide des documents techniques, identifier les blocs réalisant le stockage interne des données et les blocs réalisant la communication entre ces unités.

L' « unité patient » réalise une acquisition pendant toute sa phase d'autonomie en énergie. Pour chaque voie d'acquisition, chaque échantillon est stocké sur 16 bits, soit 2 octets.

Question 2. Déterminer la taille des données lors de cette acquisition. En déduire les exigences supplémentaires sur les blocs de stockage (capacité et débit requis) et de communication (débit requis).

Les communications dans la bande ISM ont des caractéristiques proches des réseaux Wi-Fi classiques. Les spécifications Wi-Fi et USB sont disponibles dans le document technique DT5.

Question 3. En justifiant vos raisonnements, valider les modes Wi-Fi et USB satisfaisant les exigences du système et proposer d'autres solutions techniques répondant aux besoins. Conclure quant à la pertinence du choix de ces moyens de communication dans le développement du système.

Sous-partie 1.2. Stockage des informations du patient (sous-système VPS)

Le praticien officie dans un établissement centralisant toutes les données relatives aux consultations, examens, traitements et interventions sur les patients. Cette organisation permet à plusieurs médecins d'accéder aux informations traitées en utilisant différents terminaux (applications dédiées sur PC, clients légers sur tablettes, terminaux dédiés sur les instruments médicaux...).

Le stockage de ces informations est réalisé sur un VPS (Virtual Private Server) agréé. Le stockage et l'accès aux données sont organisés selon une architecture trois tiers :

- la couche « données » réalise l'archivage dans une unique base de données classique (type mySQL) ;
- la couche « traitement » est constituée d'un ensemble d'applications dédiées transcrivant les données dans un format compatibles avec les applications clients ;
- la couche « présentation » est spécifique à chaque type de terminal, elle affiche les données aux clients et permet leurs saisies et mises à jour.

Pour chaque patient, il est nécessaire de stocker les informations d'identité (nom, prénom, date de naissance, numéro de sécurité sociale...) et des informations médicales (allergies, organismes de rattachement...). Chaque patient peut suivre une ou plusieurs interventions, caractérisées par un nom, une date d'entrée, une date de sortie et des commentaires laissés par un ou plusieurs praticiens. Les personnels de l'établissement de soin sont identifiés par des informations nominatives (nom, prénom) et de niveau d'accès (fonction comme « administratif », « praticien »).

Question 4. À l'aide de la description précédente, proposer une modélisation sous la forme d'un diagramme de blocs internes du sous-système VPS.

Pour chaque table, les différentes entrées sont identifiées uniquement par un champ supplémentaire « Identifiant » étant un entier auto-incrémenté. Ce champ unique est utilisé pour réaliser les relations entre les tables.

Question 5. Déterminer le type de diagramme UML le plus approprié à la modélisation de la structure d'une base de données. Proposer un modèle conforme à la description textuelle de cette base.

Question 6. Indiquer le nombre de tables requis dans cette base de données. En syntaxe SQL ou approchée, réaliser la requête de création de la table « personnels » en utilisant le type de données qui vous semble approprié.

Sur son poste de travail connecté au réseau de l'établissement par une liaison Ethernet filaire, le praticien dispose d'un client léger sous la forme d'un « navigateur web » classique. Les données relatives à l'intervention sont accessibles en se connectant sur une des applications de traitement implémentant un serveur via un protocole HTTPS.

Question 7. Dans le modèle OSI composé de sept couches (physique, liaison, réseau, transport, session, présentation, application), indiquer les principales couches concernées par le protocole HTTPS. Préciser les protocoles et standards utilisés par les autres couches, à l'exception de la couche « session » afin que le navigateur puisse échanger des données avec le serveur via HTTPS.

Le praticien consulte la fiche du patient d'identifiant 135, l'adresse de cette page est : <https://xxx.yyy/FichePatient?Identifiant=135>

L'application de traitement réalise l'interface entre la base de données, consultée via SQL, et le navigateur web utilisé par le praticien, connecté en HTTPS. Le diagramme de séquence de la figure 3 schématise ces échanges de données.

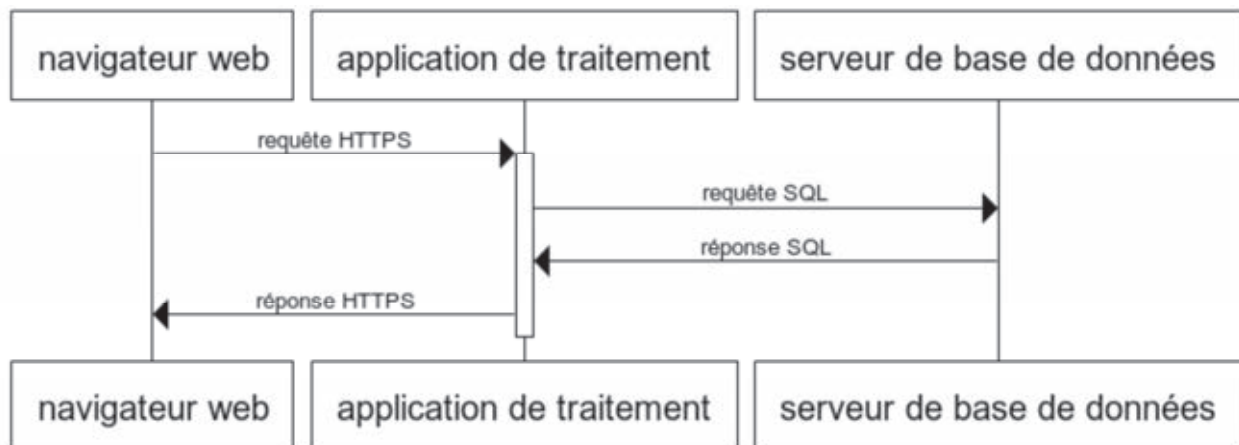


Figure 3 : diagramme de séquence suite à une requête d'un praticien

Lors de la réception d'une requête HTTPS, l'application de traitement extrait l'identifiant du patient et interroge la base de données afin de récupérer les champs « nom », « prenom », « date_naissance » et « numero_securite_sociale » correspondant. Une fois les résultats reçus, l'application de traitement crée dynamiquement le contenu de cette page.

Question 8. En syntaxe SQL ou approchée, déterminer une requête correspondant à l'interrogation de la base de données. La réponse de l'application de traitement vers le navigateur web contient les différentes informations relatives au patient, indiquer le type de langage classiquement utilisé dans cet échange.

Sous-partie 1.3. Bilan de la structure de stockage et de traitement

Question 9. Critiquer la pertinence du choix d'une architecture trois tiers pour le VPS vis-à-vis de la complexité de développement, de déploiement et de maintenance. Valider la capacité du système à satisfaire les contraintes du diagramme des exigences. Critiquer la pertinence de la séparation matérielle et logicielle de l'« unité patient » de l'« unité praticien ».

Partie 2. Modélisation des signaux

Objectifs : traiter et caractériser automatiquement les signaux issus des enregistrements, implémenter ces algorithmes dans un langage adapté, comparer différentes approches spectrales par filtrage récursif ou par transformations de Fourier.

La caractérisation des signaux est implémentée dans l' « unité praticien ». Les données issues du dispositif d'acquisition sont traitées automatiquement dans le but de déterminer les niveaux de sommeils et les éventuels symptômes typiques de troubles du sommeil.

Sous-partie 2.1. Implémentation de règles de fusion

Les traitements, tels que ceux présentés dans la figure 2 sont décrits à l'aide de règles de fusions, définies dans le document technique DT1.

Question 10. Implémenter sous forme graphique et en pseudo-code la fonction logique « majoritaire » en utilisant les fonctions « ET » et « OU » décrites dans le document technique DT1.

Question 11. Sans définir les méthodes appelées, écrire à l'aide de pseudo-code les règles de validation du stade « REM » définies dans la figure 2.

Dans cette figure 2, le résultat de la fonction `EEGLowWavesEnergy1` est une valeur d'un type énuméré à trois valeurs distinctes « low », « middle » et « high ». L'utilisation de ce type en Python est la suivante :

```
>>> from enum import Enum
>>> Ret = Enum('Ret', 'low middle high')
>>> Ret.low
<Ret.low: 1>
>>> list(Ret)
[<Ret.low: 1>, <Ret.middle: 2>, <Ret.high: 3>]
>>> a = Ret.low
>>> b = Ret.middle
>>> a == b
False
>>> a == Ret.low
True
```

Question 12. À partir de ces cas d'utilisation de la classe Python `Enum`, proposer une écriture de la fonction majoritaire dans l'état « low » dans le cas de valeurs de retour énumérées.

Les sous-parties suivantes consistent en la modélisation et l'implémentation des différentes méthodes appelées. La figure 4 représente les signaux pendant une époque caractéristique du sommeil REM.

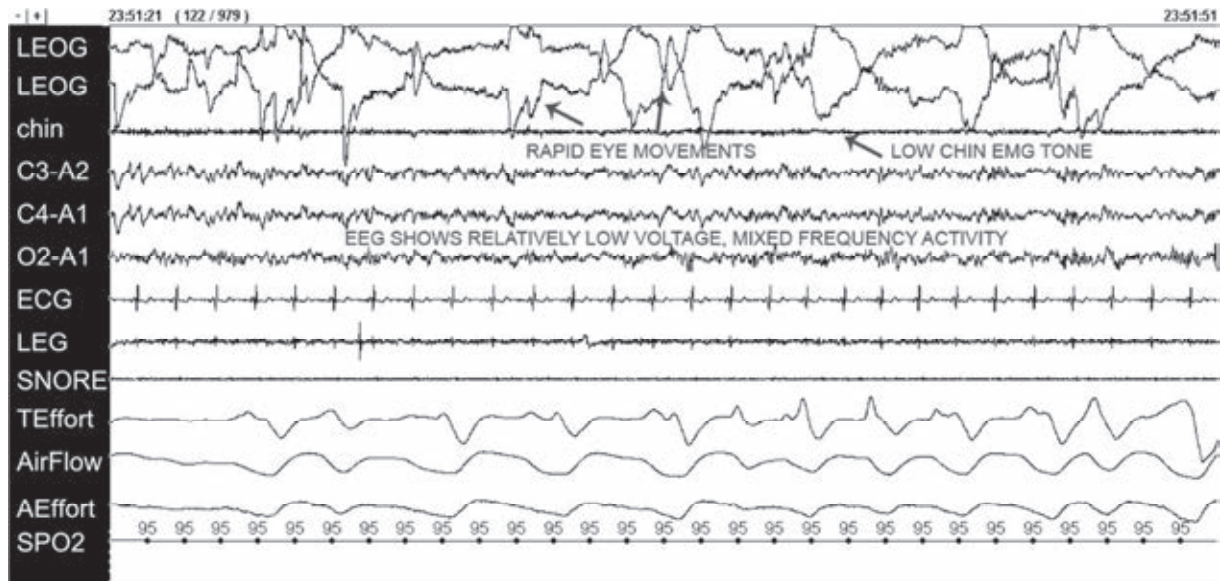


Figure 4 : signaux caractéristiques du stade de sommeil REM

Sous-partie 2.2. Implémentation de « EEGProportionThetaInstFreq1 »

La fréquence instantanée est une caractéristique des signaux. Elle consiste à déterminer grossièrement les fréquences majoritaires d'un signal EEG. La fonction « EEGProportionThetaInstFreq1 » vise à déterminer la proportion des composantes fréquences dans la bande de fréquence [4 Hz...7 Hz] d'un des signaux de l'électroencéphalogramme. Pour cela, il est nécessaire de filtrer le signal afin de conserver les fréquences intermédiaires, puis de comparer la valeur efficace sur une époque de ce signal filtré par rapport à la valeur efficace du même signal complet. Le filtrage nécessite l'utilisation de 21 coefficients dont la figure 5 présente les valeurs numériques ainsi que leur allure graphique (réponse impulsionnelle du filtre, obtenue par convolution des coefficients avec l'impulsion de Dirac).

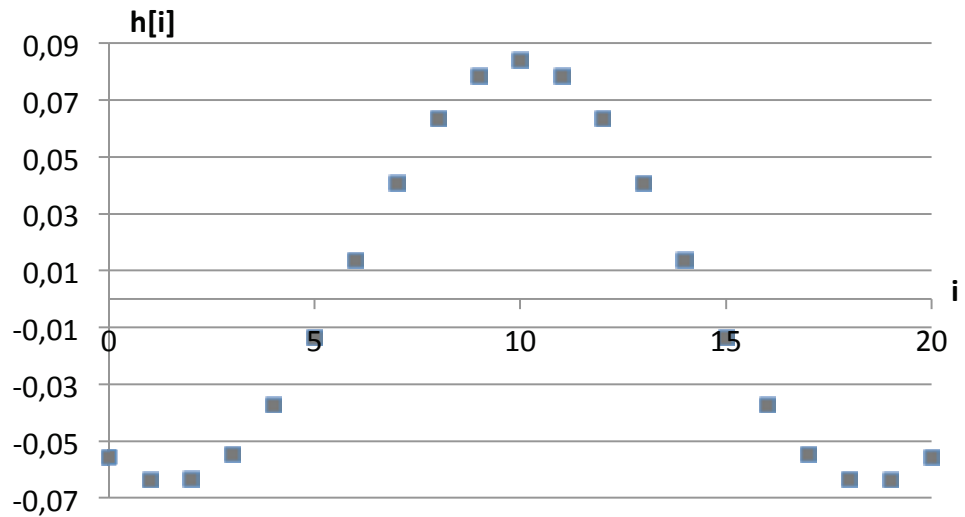


Figure 5 : coefficient du filtre passe-bande [4Hz...7Hz]

Le principe du filtrage par convolution est donné dans la figure 6. Les signaux sont échantillonnés à la fréquence F_R .

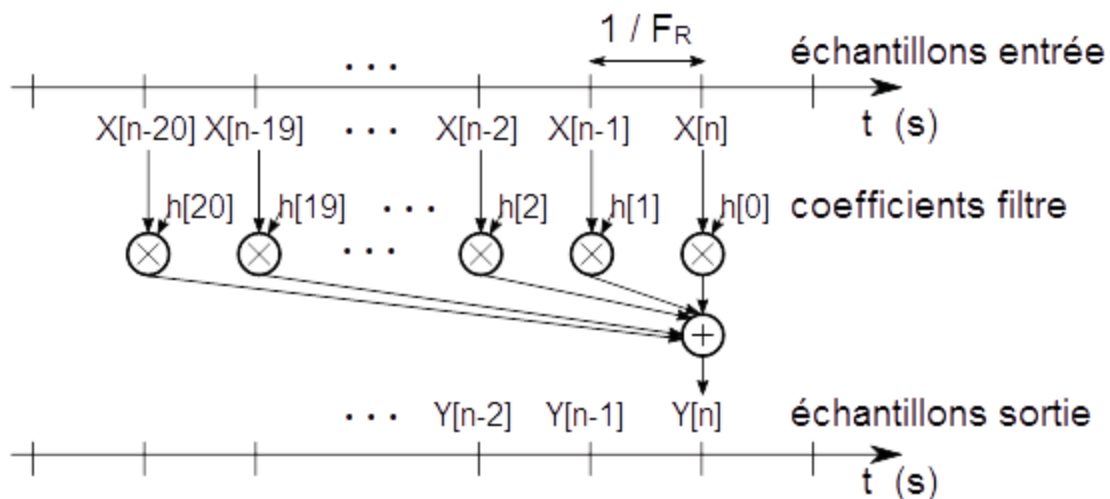


Figure 6 : principe du filtrage par convolution pour une fenêtre à 21 points

Question 13. Écrire la relation de récurrence permettant d'exprimer $Y[n]$ en fonction des $X[n - 20] \dots X[n]$ conformément au schéma de la figure 6. En déduire la fonction de transfert $F(z)$ de ce filtre définie comme $F(z) = \frac{Y(z)}{X(z)}$.

Question 14. L'observation de la figure 5 met en évidence une symétrie dans les coefficients $h[i]$. Exploiter cette particularité afin de diminuer la puissance de calcul nécessaire. Évaluer le gain en nombre de produits et d'accumulations afin d'obtenir un échantillon filtré. Indiquer le gain de complexité algorithmique de cette modification.

La figure 7 représente le diagramme en flot de données de la méthode « EEGProportionThetaInstFreq1 ». Les deux branches du flot de données représentent respectivement le traitement sur le signal filtré et les traitements sur le signal non-filtré. Les échantillons, cadencés à la fréquence F_R , sont découpés par tableaux de 3000 éléments. L'intégralité de ce tableau sera renouvelée à un cadencement $F_R / 3000$.

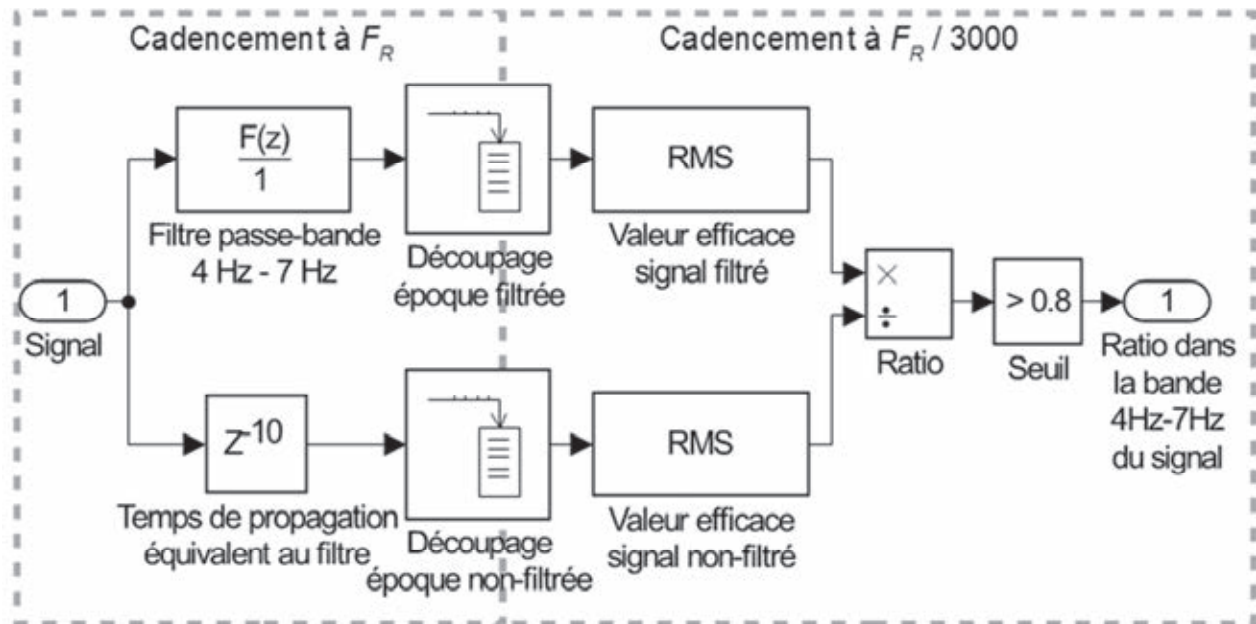


Figure 7 : diagramme flot de données de la méthode EEGProportionThetaInstFreq1

La branche inférieure du flot de données repose sur un délai équivalent suivi d'un découpage d'une partie d'un tableau. Cette opération de copie en mémoire peut être éliminée en implémentant la fonction RMS « Root Mean Square » afin qu'elle effectue ses calculs sur une sous-partie du tableau `echants` commençant à la borne `debut` passée en paramètres.

À temps continu, la valeur RMS de la fonction mathématique f calculée sur un intervalle temporel de largeur T est définie par la relation suivante : $RMS_f = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} f^2(t) dt}$ où t_0 représente le temps de début de l'intervalle d'observation de largeur T . Dans ce système spécifique, les signaux sont échantillonnés à la période $T_R = 1/F_R$ et l'intervalle d'observation correspond à une époque, soit 3000 échantillons (numérotés de 0 à 2999).

Question 15. Discrétiser la relation précédente et proposer une formule RMS adaptée au modèle à temps discret de ce système. Écrire le corps de la fonction RMS calculant la valeur efficace de 3000 échantillons (nombres flottants) stockés dans un tableau, le langage Python sera préféré.

Question 16. Indiquer les opérations mathématiques les plus coûteuses en complexité et en temps de calcul de votre algorithme de la fonction `RMS`. L'étape suivante dans le flot de données étant le ratio entre deux valeurs efficaces, suivies d'une comparaison à un seuil, proposer une réécriture du diagramme de flot de données simplifiant ces opérations coûteuses.

L'enregistrement total dure dix heures. Le tableau de nombres flottants « `Signal` » contient l'intégralité des échantillons d'un électroencéphalogramme lus depuis un enregistrement. Les coefficients du filtre sont présents dans la variable globale de type tableau `H`. La valeur de retour, stockée dans le tableau « `Ratio` » contient un élément de type booléen par époque.

Question 17. Écrire la fonction `EEGProportionThetaInstFreq1` implémentant les traitements vus dans la figure 7, le langage Python sera préféré.

Sous-partie 2.3. Implémentation de « `EEGVariabilite1` »

La fonction `EEGVariabilite1` vise à calculer la proportion des composantes fréquences supérieures à 18 Hz dans le signal traité. Sa structure est identique à celle de la fonction `EEGProportionThetaInstFreq1` (figure 7). Seuls les coefficients et la réponse du filtre numérique sont différents : il s'agit d'un filtre passe haut, cadencé à la fréquence d'échantillonnage $F_R = 100 \text{ Hz}$ conservant les fréquences supérieures à 18 Hz.

Les coefficients de ce filtre passe-haut numérique d'ordre $n = 20$ sont synthétisés selon la méthode dite de la fenêtre :

- le gabarit fréquentiel est transformé dans le domaine temporel à l'aide d'une transformation de Fourier inverse ;
- la réponse temporelle obtenue est tronquée par une fenêtre rectangulaire temporelle de largeur n/F_R centrée autour de $t = 0 \text{ s}$, cette réponse fenêtrée est notée $h_w(t)$;
- cette réponse temporelle tronquée ($h_w(t)$) est échantillonnée au cadencement F_R afin de créer le signal échantillonné $h_s(t)$;
- cette réponse temporelle tronquée et échantillonnée ($h_s(t)$) est retardée afin d'être rendue causale (signal noté $h_c(t)$) ;
- les $n + 1$ points de ce signal causal $h_c(t)$ sont conservés afin de créer le vecteur des coefficients de ce filtre, noté $h[i]$.

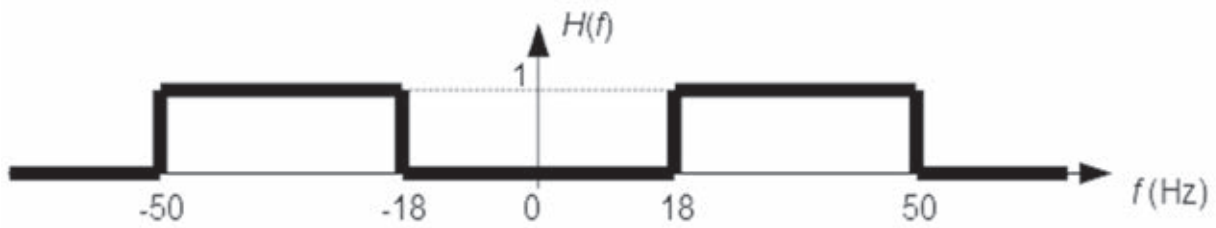


Figure 8 : gabarit de la réponse fréquentielle du filtre passe-haut échantillonné

Le gabarit fréquentiel du filtre analogique équivalent est donné dans la figure 8, la réponse est tronquée à la fréquence de Nyquist ($F_R/2$).

La définition de la transformation de Fourier inverse est :

$$h(t) = TF^{-1}\{H(f)\} = \int_{-\infty}^{+\infty} H(f)e^{+2\pi jft} dt$$

Question 18. Montrer que la transformée de Fourier inverse de ce gabarit peut s'écrire sous la forme $h(t) = 64 \times \cos(68 \times \pi \times t) \times \text{sinc}(32 \times \pi \times t)$.

Question 19. Adapter l'algorithme vu à la Question 17. de la fonction `EEGProportionThetaInstFreq1` afin de calculer simultanément les ratios d'ondes dans la bande [4 Hz...7 Hz] et dans la bande [18 Hz...50 Hz], le langage Python sera préféré. Évaluer le gain obtenu en mutualisant ainsi les fonctions `EEGProportionThetaInstFreq1` et `EEGVariabilite1`.

Sous-partie 2.4. Implémentation de « `EEGLowWavesEnergy1` »

La fonction `EEGLowWavesEnergy1` calcule l'énergie des composantes d'un signal d'électroencéphalogramme situées sur la bande de fréquences 0 – 2 Hz. La méthode pour déterminer ce paramètre est fournie dans le schéma bloc de la figure 9.

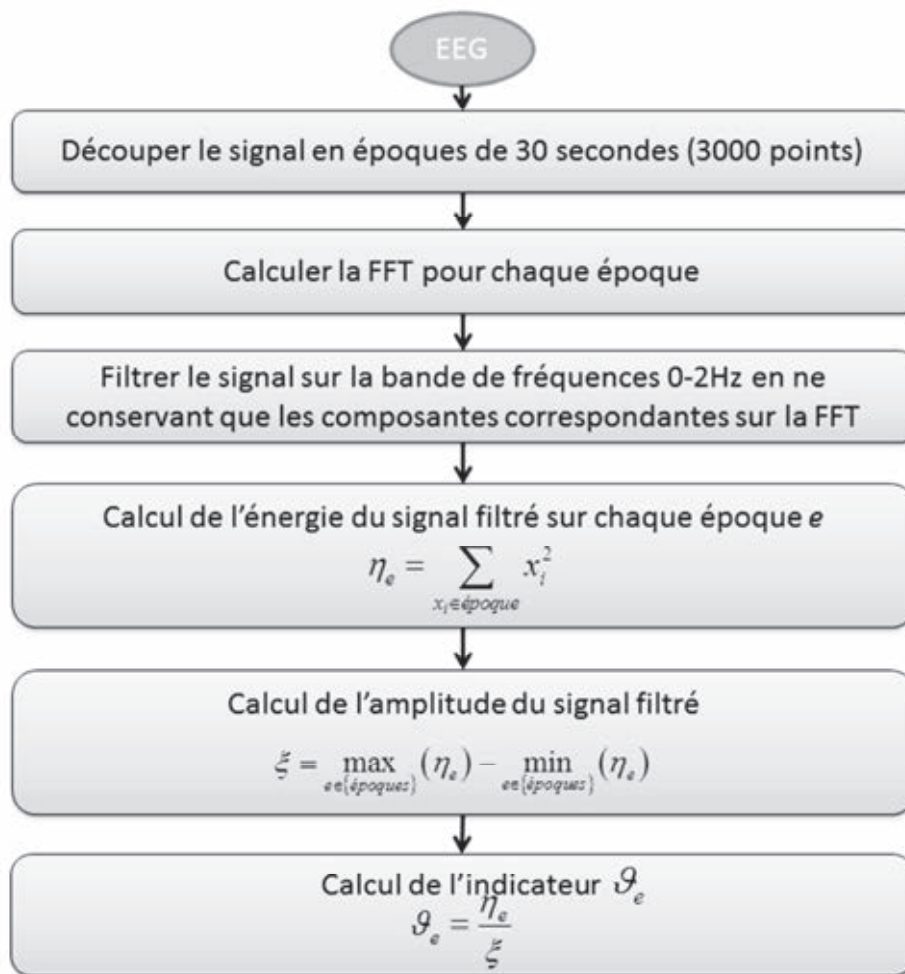


Figure 9 : schéma bloc du paramètre « Énergie des ondes lentes »

Cette fonction nécessite le calcul de la transformée de Fourier rapide (Fast Fourier Transform ou FFT). Soient x_0, x_1, \dots, x_{N-1} les N échantillons du signal X et j le nombre imaginaire, la fonction FFT renvoie un vecteur de N composantes.

La $m^{\text{ème}}$ composante ($m = 0, 1, \dots, N - 1$) de la FFT de ce signal vaut par définition :

$$FFT_{x,m} = \frac{1}{N} \times \sum_{k=0}^{N-1} x_k \times e^{-\frac{2 \cdot j \cdot \pi \cdot m \cdot k}{N}}$$

Le signal de l'électroencéphalogramme est échantillonné à la fréquence F_R , le $m^{\text{ème}}$ élément de ce vecteur correspond à la composante de fréquence $f_m = \frac{m \cdot F_R}{N}$.

Question 20. Le signal X est échantillonné et réel, indiquer les conséquences de ces deux caractéristiques sur son spectre. En déduire la relation entre les coefficients $FFT_{x,l}$ et $FFT_{x,N-l}$ où l est un nombre entier de l'intervalle $1 \dots N - 1$.

Question 21. Le signal $x(t)$ est défini par la formule, $x(t) = 10^{-5} \times \sin(0.6 \times \pi \times t)$.
En déduire une écriture des échantillons x_k de ce signal.

Question 22. Tracer en partie réelle et en partie imaginaire l'allure du spectre de ce signal échantillonné, en déduire le vecteur spectre $FFT_{x,k}$ calculé par la fonction FFT.

Seules les composantes dans la bande de fréquence 0-2 Hz sont nécessaires pour la suite du calcul. L'énergie du signal dans cette bande de fréquence est donnée par la relation :

$$\eta_e = 2 \times \sum_{k=0}^{\frac{2}{F_R} \times N} |FFT_{x,k}|^2 = 2 \times \sum_{k=0}^{60} |FFT_{x,k}|^2$$

Les correspondances des opérateurs mathématiques et matériels sont les suivantes :

- un produit entre deux valeurs réelles = une multiplication matérielle ;
- un produit entre une valeur réelle et une complexe = deux multiplications matérielles ;
- un produit entre deux valeurs complexes = quatre multiplications matérielles ;
- la somme de deux valeurs réelles = une addition matérielle ;
- la somme de deux valeurs complexes = deux additions matérielles.

Question 23. Déterminer le nombre de produits-accumulations nécessaires afin de calculer les composantes fréquentielles correspondantes ainsi que pour obtenir l'énergie du signal basse-fréquence sur une époque, notée η_e .

Sous-partie 2.5. Bilan de la caractérisation des signaux

Question 24. Proposer des optimisations envisageables dans les différentes implémentations tout en respectant les règles de diagnostics.

Question 25. À partir des traitements effectués dans les fonctions précédentes, justifier le traitement différé des données préalablement enregistrées du sommeil. Conclure sur la faisabilité de la caractérisation automatique des signaux.

Partie 3. Modélisation des optimisations du diagnostic

Objectifs : prédire les performances de l'implémentation des caractérisations des signaux en tirant partie de la parallélisation des calculs, comparer les différentes approches possibles (multithread, multiprocess).

L'application de diagnostic automatisé s'exécute sur la station de travail du praticien. Ce programme multithread nécessite une configuration matérielle standard : monoprocesseur à quatre cœurs équipé de 8 Go de RAM, système d'exploitation 64 bits « grand public ».

Sous-partie 3.1. Temps d'exécution

La détermination des stades de sommeil nécessite des calculs répétitifs sur les différentes époques. Dans le cadre du stade REM, les fonctions `EEGSleepSpindles1`, `EEGVariabilite1`, `EEGLowWavesEnergy1` et `EEGProportionThetaInstFreq1` s'exécutent sur des signaux distincts C4-A1, C3-A2 et O1-A2. Afin de réduire le temps d'attente du praticien avant la présentation des résultats, les traitements sont parallélisés sur les différents cœurs du processeur de la station de travail. La structure du programme est présentée dans la figure 10 :

- l'appui sur un bouton spécifique de l'interface graphique provoque la création de trois threads de traitement, chacun paramétré avec le nom du signal de l'électroencéphalogramme à traiter ;
- chaque thread exécute les cinq étapes de traitements à la suite (figure 10) ;
- les messages de l'interface graphique de l'application sont continuellement traités en parallèle par le thread principal du programme, ce traitement évite l'interface figée en répondant aux différents événements de l'utilisateur et du système ;
- les résultats des différentes fonctions de caractérisations (indicateurs Low/Middle/High/Absence/Présence/Stable/Instable) sont fusionnés conformément aux règles logiques précisées dans la figure 2 ;
- enfin les résultats sont affichés dans l'interface graphique.

Les temps moyens constatés pour les différentes étapes sont les suivants :

- mise à jour de l'affichage, $t_A = 5 \text{ ms}$ par message ;
- création d'un thread par le système d'exploitation, $t_C = 20 \text{ ms}$ par thread ;
- envoi d'un signal inter-thread, $t_S = 0.5 \text{ ms}$ par signal ;
- destruction et synchronisation d'un thread, $t_D = 20 \text{ ms}$ par thread ;
- chargement d'un EEG, $t_L = 10 \text{ ms}$;

- traitement EEGLowWavesEnergy1, $t_1 = 384 \text{ ms}$;
- traitement EEGProportionThetalnstFreq1, $t_2 = 28 \text{ ms}$;
- traitement EEGSleepSpinless1, $t_3 = 428 \text{ ms}$;
- traitement EEGVariabilite1, $t_4 = 28 \text{ ms}$.

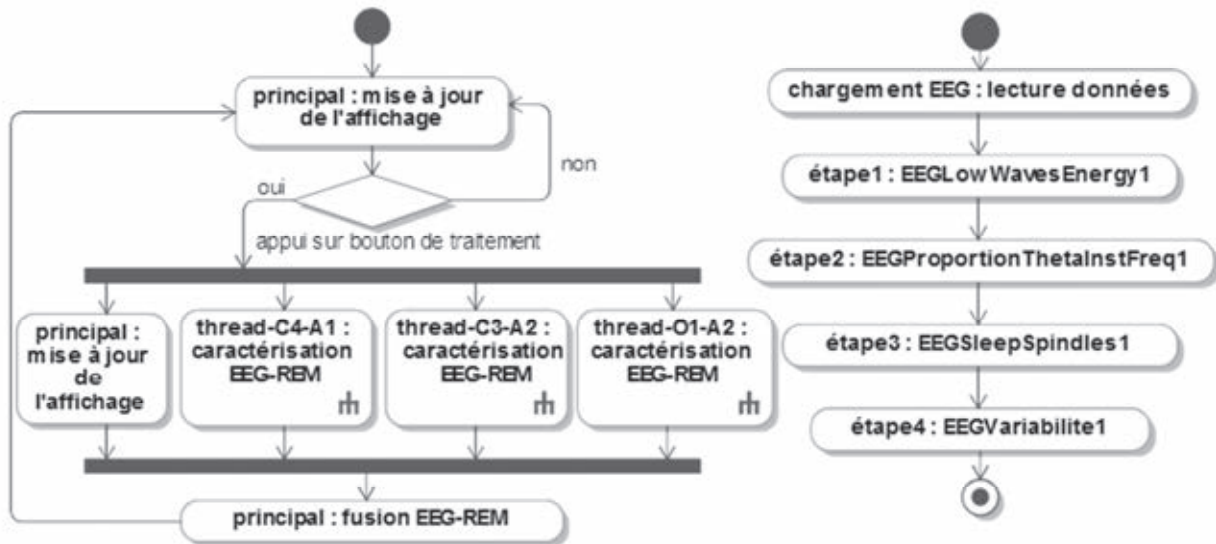


Figure 10 : diagramme du programme complet (à gauche) et détails pour les trois threads de caractérisation (à droite)

Question 26. À partir de la description des traitements, déduire le diagramme de séquence mettant en œuvre le thread principal et les trois threads de traitement de l'application.

Question 27. Déterminer le temps d'exécution du traitement présenté dans la figure 10. Préciser les critères aboutissant à une répétabilité de ce temps de calcul. Comparer ce temps au temps équivalent obtenu si le thread principal effectuait tous les traitements.

Sous-partie 3.2. Gestion de la mémoire

Question 28. Dans un langage compilé et exécuté, tel que le C ou le C++, indiquer si l'espace mémoire occupé par les variables locales des fonctions est alloué sur le tas ou sur la pile. Préciser le mécanisme mis en œuvre par le système d'exploitation afin d'assurer l'indépendance des appels en parallèle des mêmes fonctions par plusieurs threads.

Pour les langages de haut-niveau, tels que Python, toutes les variables sont allouées sur le tas, leur durée de vie est déterminée par des compteurs de références avec

désallocation immédiate, lorsque le compteur de référence revient à zéro, ou après passage d'un ramasse-miettes (*garbage collector*).

Question 29. Dans ce cas, proposer un mécanisme mis en œuvre par le compilateur et/ou l'interpréteur afin de garantir l'indépendance des appels en parallèle des mêmes fonctions par plusieurs threads.

Les systèmes d'exploitation proposent différents mécanismes parallèles d'exécution :

- les processus sont les programmes en cours d'exécution, chacun dispose d'un tas mémoire propre et de ressources lui permettant de communiquer (fichiers, connexions réseaux, mémoires partagées avec d'autres programmes...), un processus est composé d'un ou plusieurs threads ;
- les threads sont des fils d'exécution, chacun dispose d'une pile mémoire propre et ne peut accéder uniquement qu'au tas mémoire et aux ressources du processus courant,

Les communications et synchronisations inter-processus sont lentes, elles nécessitent systématiquement des appels au noyau du système d'exploitation afin d'échanger des données en mémoires, dans un fichier ou par le biais d'une communication réseau. Les communications et synchronisations inter-thread sont rapides puisque les échanges sont directement possibles dans le tas mémoire partagé par tous les threads à l'intérieur d'un même programme.

Question 30. À partir du diagramme et de la description du programme, lister les paramètres envoyés depuis le thread principal vers les threads de traitement avant le lancement de ceux-ci, lister également les valeurs retournées par les threads de traitement vers le thread principal une fois leur traitement achevé.

Sous-partie 3.3. Synthèse des optimisations

Question 31. Indiquer les modifications à prévoir pour répartir l'exécution d'un traitement sur plusieurs programmes s'exécutant sur plusieurs stations de travail distinctes. Conclure sur la pertinence du choix d'une structure multithread pour cette application de caractérisation automatique des signaux.

Partie 4. Modélisation de la chaîne d'acquisition

Objectifs : acquérir les signaux pendant le sommeil du patient tout en limitant en temps réel l'influence du bruit, implémenter efficacement un filtre analogique et un filtre numérique.

L'interface d'acquisition « unité patient » est un dispositif autonome réalisant l'instrumentation médicale pendant l'examen. Elle est basée autour d'un processeur embarqué de traitement de signal (DSP) et sauvegarde les signaux relevés dans une carte mémoire.

Les signaux analogiques présentés dans la figure 1 ont des fréquences caractéristiques faibles : la grille horizontale correspond à une seconde, ainsi 30 secondes sont représentées dans cette figure, les fréquences caractéristiques des signaux sont de l'ordre de 0,2 à 15 Hz. Les échantillons des signaux sont enregistrés à une fréquence $F_R = 100 \text{ Hz}$. L'échelle des signaux de l'électroencéphalogramme est en μV , ceux de l'électrocardiogramme est en mV. Ces signaux peuvent être facilement perturbés par l'environnement électromagnétique lors de la mesure. Une attention particulière est portée au filtrage fréquentiel des échantillons.

Dans le but de limiter le repliement fréquentiel, un filtrage à deux étages est réalisé (figure 11). Un premier filtre anti-repliement analogique précède le convertisseur analogique-numérique cadencé à une fréquence d'échantillonnage $F_E = 32 \times 100 = 3,2 \text{ kHz}$. Un second filtre numérique réalise précisément la réjection des fréquences indésirables. Enfin le signal est décimé afin de conserver un seul échantillon à la fréquence F_R .

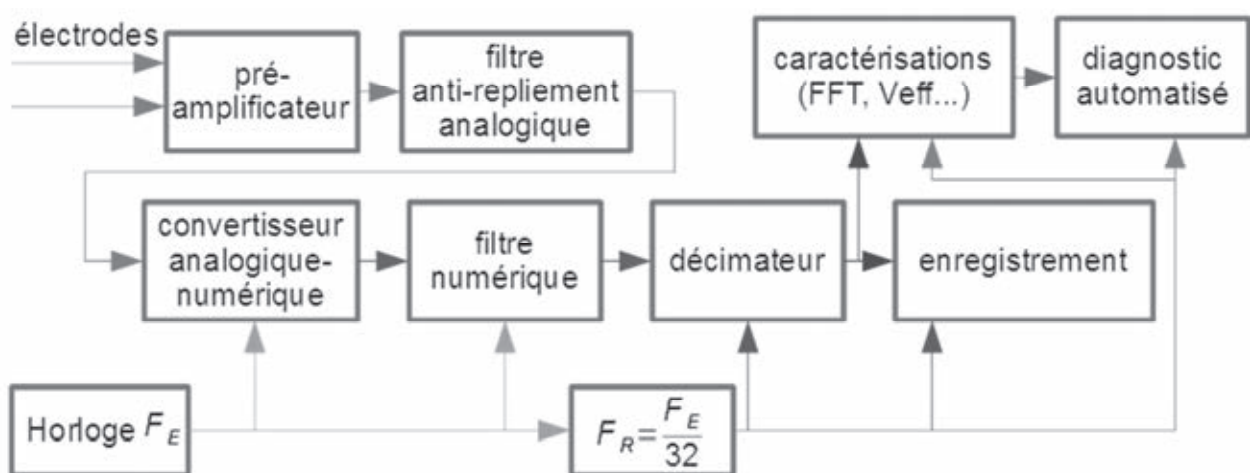


Figure 11 : synoptique complet de la chaîne d'acquisition et de traitement

Toute la chaîne d'amplification et de traitement est alimentée par une tension asymétrique $V_{DD} = 3,3 V$ et $V_{SS} = 0 V$.

Tous les blocs fonctionnels en aval du convertisseur analogique-numérique sont implémentés dans un processeur de traitement de signal (DSP).

Sous-partie 4.1. Signaux bruités et filtrage analogique anti-repliement

Dans cette sous-partie traitant de l'amplification des signaux électriques, les puissances sont exprimées en V^2 : pour chaque signal, elles correspondent à la puissance équivalente dissipée par une résistance de 1Ω alimentée par ce signal. La mesure du signal de l'électroencéphalogramme est bruitée. La puissance du signal à l'entrée du préamplificateur est notée $P_{SE} = 6,4 \times 10^{-10} V^2$, la puissance du bruit à l'entrée $P_{BE} = 8,2 \times 10^{-10} V^2$ et le rapport signal à bruit en entrée $RSB_E = P_{SE}/P_{BE}$.

À l'entrée du préamplificateur, les allures des densités spectrales de puissance du bruit (P_{BE}) et du signal (P_{SE}) sont représentées dans la courbe A du document réponse DR1.

La densité spectrale de puissance de ce bruit à l'entrée de l'amplificateur, notée η_{BE} , est supposé constante (hypothèse de bruit blanc) dans la bande de fréquence $[0 \dots 10 MHz]$:

$$\eta_{BE} = \frac{P_{BE}}{10 MHz}$$

La figure 12 représente les modèles équivalents de l'amplificateur en considérant uniquement le signal (en haut) et uniquement le bruit (en bas). Chaque étage de l'amplificateur a un gain de 100 et possède une bande passante limitée $[0 \dots F_{MAX}]$. Les caractéristiques suivantes seront utilisées : $P_{BM} = 4.5 \times 10^{-11} V^2$ et $F_{MAX} = 500 kHz$.

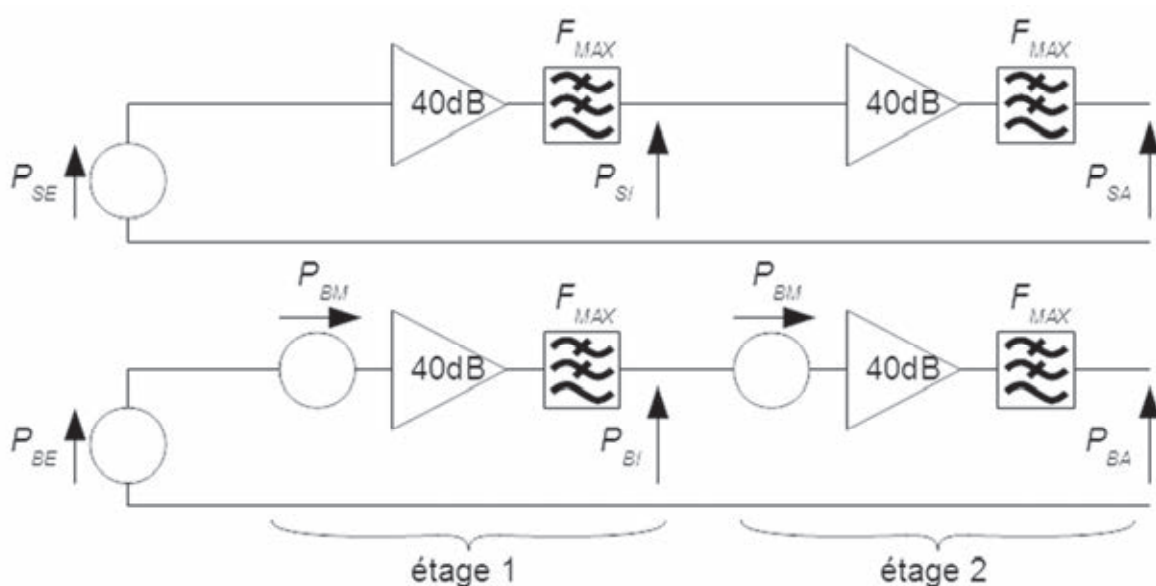


Figure 12 : modèle du préamplificateur pour le signal (en haut) et pour le bruit (en bas)

Question 32. Tracer l'allure du spectre du signal et du bruit préamplifiés dans la courbe B du document réponse DR1.

Question 33. Calculer les puissances des signaux P_{SI} et P_{SA} en fonction de P_{SE} . Calculer les puissances des bruits P_{BI} et P_{BA} en fonction de P_{BE} . En déduire le rapport signal à bruit RSB_A en sortie de l'amplificateur.

Le préamplificateur est connecté au filtre anti-repliement analogique. Ce filtre d'ordre 1 a une fréquence de coupure $F_C = 30 \text{ Hz}$. Sa réponse en fréquence est de la forme :

$$F_A(p) = \frac{1}{1 + \frac{p}{2 \times \pi \times F_C}}$$

Question 34. À partir des caractéristiques du filtre anti-repliement analogique, tracer l'allure du spectre du signal filtré dans la courbe C du document réponse DR1.

Le rapport signal à bruit d'un convertisseur analogique-numérique sur N_{bit} est pour un signal « pleine-échelle » à son entrée : $RSB \approx 1.76 + 6.02 \times N_{bit}$. La tension nominale de sortie du pré-amplificateur est $V_{App} = 2 \text{ V}$

Question 35. Déterminer la puissance de signal P_{SC} et la puissance du bruit P_{BC} présents à l'entrée du convertisseur analogique-numérique. En déduire le nombre de bits significatifs atteignables lors de la conversion.

Question 36. Conclure sur la précision réelle de la mesure réalisée. Proposer des modifications possibles de la chaîne d'acquisition permettant d'augmenter la précision de la mesure.

Sous-partie 4.2. Numérisation des signaux analogiques issus du patient

Le convertisseur analogique-numérique réalise la conversion des tensions de l'intervalle $[GND \dots V_{DD}]$ vers des nombres non signés sur 10 bits à la fréquence d'échantillonnage $F_E = 3.2 \text{ kHz}$. La tension à l'entrée du convertisseur analogique-numérique est notée V_{CAN} et le mot numérisé N_{CAN} . En fonctionnement normal, la tension V_{CAN} a une valeur moyenne de 1,65 V et une amplitude maximale de $\pm 1 \text{ V}$.

Le bruit de quantification est négligé en première approximation.

Question 37. Déterminer les coefficients de la relation affine $N_{CAN} = a_3 \times V_{CAN} + b_3$. Déterminer les valeurs minimales et maximales correspondantes pour le mot N_{CAN} .

Question 38. Tracer l'allure du spectre échantillonné dans la courbe D du document réponse DR1.

Sous-partie 4.3. Filtrage numérique de réjection

Le programme de traitement de signal est écrit en C++. Les caractéristiques techniques de l'architecture matérielle et du compilateur utilisé sont précisées dans le document technique DT6. Le filtre numérique, cadencé à F_E , implémente une réponse elliptique dont le cahier des charges est le suivant :

- bande passante de 0 à 45 Hz avec une ondulation acceptable de 1 dB ;
- bande coupée à partir de 50 Hz avec une atténuation minimale de 40 dB.

Ce cahier des charges est satisfait par l'utilisation d'un filtre d'ordre 6 sous la forme :

$$F_N(z) = \frac{B[0]z^{-0} + B[1]z^{-1} + B[2]z^{-2} + B[3]z^{-3} + B[4]z^{-4} + B[5]z^{-5} + B[6]z^{-6}}{A[0]z^{-0} + A[1]z^{-1} + A[2]z^{-2} + A[3]z^{-3} + A[4]z^{-4} + A[5]z^{-5} + A[6]z^{-6}}$$

La figure 13 représente la réponse en fréquence de ce filtre.

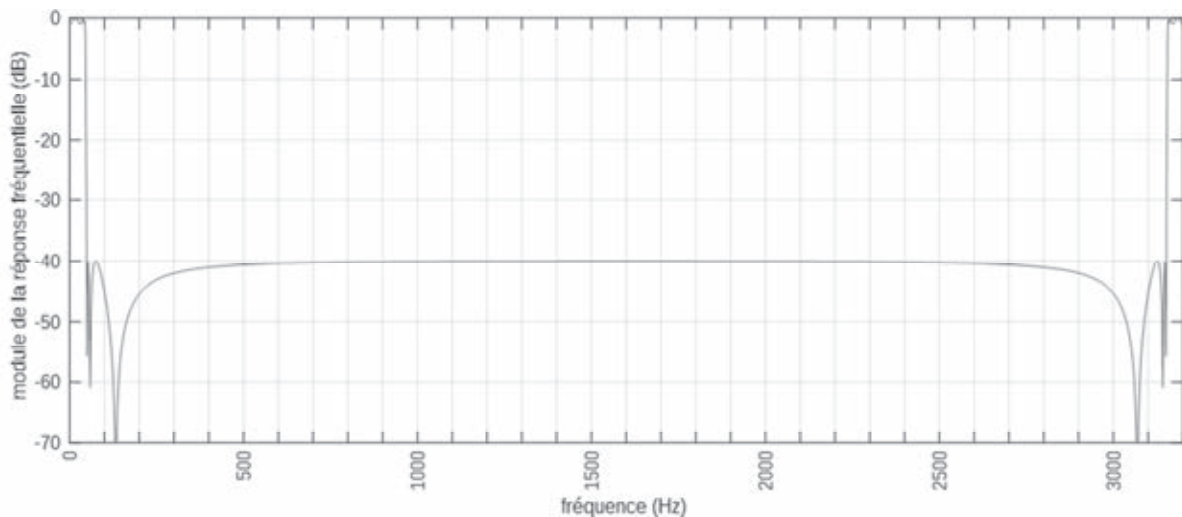


Figure 13 : réponse en fréquence du filtre elliptique

Les coefficients obtenus sont listés dans la table ci-dessous :

Coefficients du numérateur		Coefficients du dénominateur	
B[0]	0,00978992786258	A[0]	1.00000000
B[1]	-0.0578541383147	A[1]	-5.90502738953
B[2]	0.143323630095	A[2]	14.5434751511
B[3]	-0.190518766642	A[3]	-19.1224727631
B[4]	0.143323630095	A[4]	14.1569328308
B[5]	-0.0578541383147	A[5]	-5.59520769119
B[6]	0,00978992786258	A[6]	0.922300040722

Le filtre implémente l'équation récurrente suivante sous la forme de nombres flottants codés sur 64 bits : $Y_n = \sum_{i=0}^6 B[i] \times X_{n-i} - \sum_{i=1}^6 A[i] \times Y_{n-i}$

Question 39. Dans le cas de l'acquisition de trois signaux d'électroencéphalogramme, déterminer la puissance de calcul nécessaire en produits par seconde, évaluer l'occupation mémoire nécessaire à l'implémentation des filtres numériques. Tracer le spectre du signal filtré dans la courbe E du document réponse DR1.

Ce filtre elliptique a une réponse en fréquence très dépendante des erreurs d'arrondi dans les coefficients et les calculs. Ces erreurs ont des conséquences dans le placement des pôles et des zéros de la fonction de transfert aboutissant à une instabilité ou à une réponse en fréquence fortement dégradée.

En l'état, ce filtre nécessite des nombres flottants « IEEE 64 bits » afin d'être implémenté correctement. Or l'architecture matérielle prévue ne dispose pas d'unité de calcul dédiée à ce format.

Question 40. Expliquer les mécanismes d'émulation mis en œuvre par une chaîne de compilation pour proposer le support des nombres flottants codés sur 64 bits sur une architecture matérielle dépourvue de coprocesseur flottant à double précision. Indiquer les conséquences de ce type d'exécution sur les performances du système. Conclure quant à la pertinence de cette solution pour ce système.

Une solution classique à ce problème est d'implémenter un filtre récursif d'ordre élevé par la mise en série de filtres d'ordres plus faibles. Les contraintes de précisions sur les filtres d'ordres plus faibles sont généralement moins strictes.

Dans ce système, le filtre est réalisé par la mise en série de trois cellules biquadratiques (filtre récursif d'ordre 2) notées α , β et γ . Les coefficients de ces trois cellules sont :

Coefficients des numérateurs (5 chiffres significatifs)					
$B\alpha[0]$	0.21392	$B\beta[0]$	0.21392	$B\gamma[0]$	0.21392
$B\alpha[1]$	-0.41342	$B\beta[1]$	-0.42500	$B\gamma[1]$	-0.42578
$B\alpha[2]$	0.21392	$B\beta[2]$	0.21392	$B\gamma[2]$	0.21392
Coefficients des dénominateurs (5 chiffres significatifs)					
$A\alpha[0]$	1.0000	$A\beta[0]$	1.0000	$A\gamma[0]$	1.0000
$A\alpha[1]$	-1.9437	$A\beta[1]$	-1.9733	$A\gamma[1]$	-1.9880
$A\alpha[2]$	0.94575	$A\beta[2]$	0.97931	$A\gamma[2]$	0.99581

Le code est identique entre les trois cellules d'ordre 2, cette situation se traduit par la création d'une classe C++ `filter2` qui sera instanciée pour chaque cellule. La déclaration de cette classe et de la classe de `filtre` générique est la suivante :

```
// classe abstraite de filtrage
class filtre
{
    public:
        // exécution du filtre (méthode abstraite)
        // x=valeur entrante du filtre
        // retour=valeur calculée à l'instant n
        virtual float execute(float x) = 0;
};

class filtre2 : public filtre
{
    private:
        float B[3]; // coefficients du numérateur
        float A[3]; // coefficients du dénominateur
        float X[3]; // X[0]=XN-0, X[1]=XN-1, X[2]=XN-2
        float Y[3]; // Y[0]=YN-0, Y[1]=YN-1, Y[2]=YN-2
    public:
        // constructeur d'instance
        // coeffsB=pointeur vers un tableau
        // de 3 coeffs numérateur
        // coeffsA=pointeur vers un tableau
        // de 3 coeffs dénominateur
        filtre2(float *coeffsB, float *coeffsA);
        // exécution du filtre
        // x=valeur entrante du filtre
        // retour=valeur calculée à l'instant n
        virtual float execute(float x);
};
```

Question 41. Indiquer le rôle du constructeur d'instance. Proposer une implémentation du constructeur `filtre2`.

Question 42. Rappeler l'équation récurrente d'un filtre récursif d'ordre 2 et proposer une implémentation de cette équation dans la fonction `execute`.

Ce filtre d'ordre 6 est implémenté également dans une classe dont l'implémentation est la suivante :

```
class filtre6 : public filtre
{
    private:
        filtre2 alpha; // étage 1
        filtre2 beta; // étage 2
        filtre2 gamma; // étage 3
    public:
        // constructeur, Balpha, Aalpha, Bbeta, Abeta, Bgamma
        // et Agamma sont 6 tableaux de constantes déclarés
        filtre6():
            // construction de l'attribut alpha
            alpha(Balpha,Aalpha),
            // construction de l'attribut beta
            beta(Bbeta,Abeta),
            // construction de l'attribut gamma
            gamma(Bgamma,Agamma)
        {};
        // exécution du filtre
        virtual float execute(float X)
        {

            return (gamma.execute(beta.execute(alpha.execute(X))));
        };
};
```

Question 43. Indiquer le lien existant entre les classes `filtre2` et `filtre`. Indiquer le lien entre les classes `filtre6` et `filtre2`. Justifier si la destruction d'une instance de la classe `filtre6` implique la destruction des trois instances de `filtre2` automatiquement créées.

Question 44. Tracer le diagramme des classes UML décrivant les classes `filtre`, `filtre2` et `filtre6` ainsi que leurs relations.

Sous-partie 4.4. Décimation et facteur d'échelle

Seul un échantillon sur 32 est enregistré. Cette décimation doit néanmoins respecter le critère de Shannon.

Question 45. Vérifier que les caractéristiques fréquentielles du signal autorisent un échantillonnage à la fréquence F_R . Tracer l'allure du spectre du signal décimé dans la courbe F du document réponse DR1.

Dans le code suivant, écrit en C, la variable entière `Ncan` représente le mot issu du convertisseur analogique-numérique et la variable entière `Vcan` la tension estimée à l'entrée entre les électrodes en μV . Le type « `uint16` » est un entier non-signé sur 16 bits, le type « `int16` » est un entier signé sur 16 bits.

```
uint16 Ncan, Vcan;
Ncan = acquisitionCAN();
Vcan = ((int16)Ncan - 512) * 3.22265625e-7f;
```

Question 46. Indiquer précisément pour la dernière ligne de ce code toutes les opérations de calcul ainsi que les conversions implicites et explicites réalisées pour l'obtention du résultat. Préciser pour chaque opération les types des opérandes et des résultats.

Plusieurs méthodes d'arrondi sont classiquement définies dans les bibliothèques standards :

- arrondi vers 0 (implémenté par la fonction standard `trunc`), défini comme $\forall x \in \mathbb{R}, \forall n \in \mathbb{N}, |x| \in [n \dots n + 1[\Rightarrow \text{trunc}(x) = n \times \text{sign}(x)$;
- arrondi vers $-\infty$ (implémenté par la fonction standard `floor`), défini comme $\forall x \in \mathbb{R}, \forall n \in \mathbb{Z}, x \in [n \dots n + 1[\Rightarrow \text{floor}(x) = n$;
- arrondi vers $+\infty$ (implémenté par la fonction standard `ceil`), défini comme $\forall x \in \mathbb{R}, \forall n \in \mathbb{Z}, x \in]n - 1 \dots n] \Rightarrow \text{ceil}(x) = n$.

Question 47. Préciser, en donnant des exemples numériques, l'opération d'arrondi réalisée implicitement par le compilateur lors de l'affectation d'une valeur à virgule flottante dans une variable entière.

Dans le but de réduire les erreurs d'arrondi, l'opérateur d'arrondi « mathématique » $\text{round}_{\text{math}} : \mathbb{R} \rightarrow \mathbb{Z}$ est formellement défini selon la méthode suivante :

- $\forall x \in \mathbb{R}, x \geq 0.5, \forall n \in \mathbb{Z}, n \geq 1, x \in [n - 0.5 \dots n + 0.5[\Rightarrow \text{round}_{\text{math}}(x) = n$;
- $\forall x \in \mathbb{R}, -0.5 < x < 0.5, \text{round}_{\text{math}}(x) = 0$;
- $\forall x \in \mathbb{R}, x \leq -0.5, \forall n \in \mathbb{Z}, n \geq 1, x \in]n - 0.5 \dots n + 0.5] \Rightarrow \text{round}_{\text{math}}(x) = n$.

Question 48. En se limitant aux appels des trois fonctions standards `trunc`, `floor` et/ou `ceil`, proposer une méthode permettant d'obtenir un arrondi « mathématique » lors de l'affectation d'une valeur à virgule flottante, positive ou négative, dans la variable `Vcan`.

Sous-partie 4.5. Synthèse de la chaîne d'acquisition

Question 49. Indiquer les méthodes permettant d'optimiser un code de traitement de signal sur une architecture de type DSP. Conclure sur la faisabilité du traitement de signal dans ce système.

Question 50. Critiquer la pertinence du sur-échantillonnage dans l'acquisition des données. Conclure sur la validité de l'enregistrement des signaux physiologiques du patient avant de réaliser automatiquement la caractérisation sur un poste séparé.

Partie 5. Validation et limites du système proposé

Objectifs : valider la structure du système, déterminer les limites de la solution technique proposée.

Ce système est constitué de trois unités distinctes :

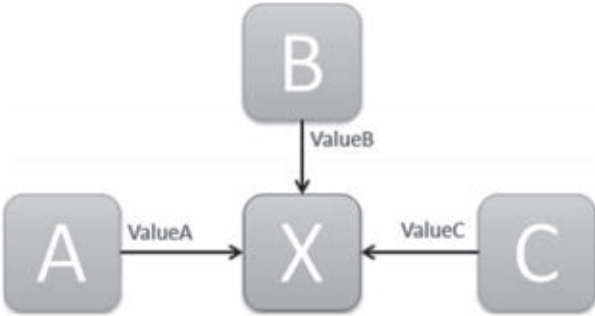
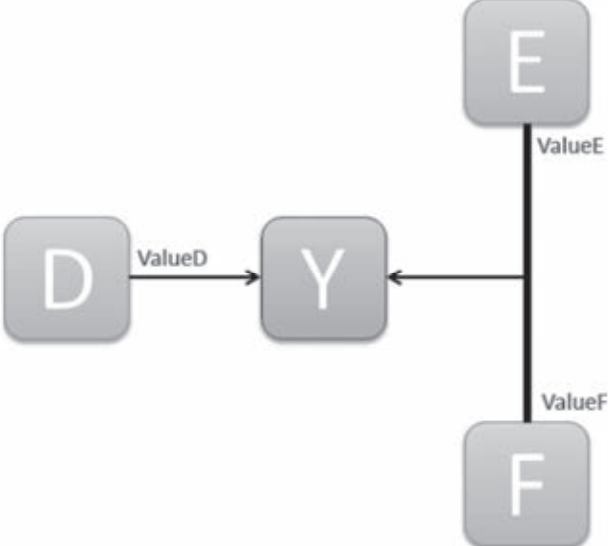
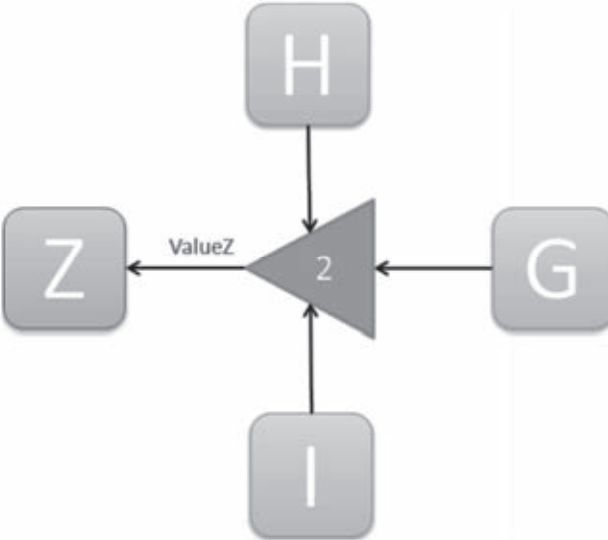
- l' « unité patient » réalise l'instrumentation des signaux lors de l'examen médical, pouvant durer une nuit entière ;
- l' « unité praticien » automatise la caractérisation de ces signaux permettant au médecin d'apporter rapidement son diagnostic ;
- à l'échelle de l'établissement de santé, les informations du patient sont stockées dans un système centralisé accessible depuis plusieurs terminaux.

Question 51. Conclure de manière synthétique sur la capacité du système à satisfaire les exigences du cahier des charges.

Question 52. Proposer trois améliorations permettant de réduire le temps de traitement ou de faciliter l'accès aux données.

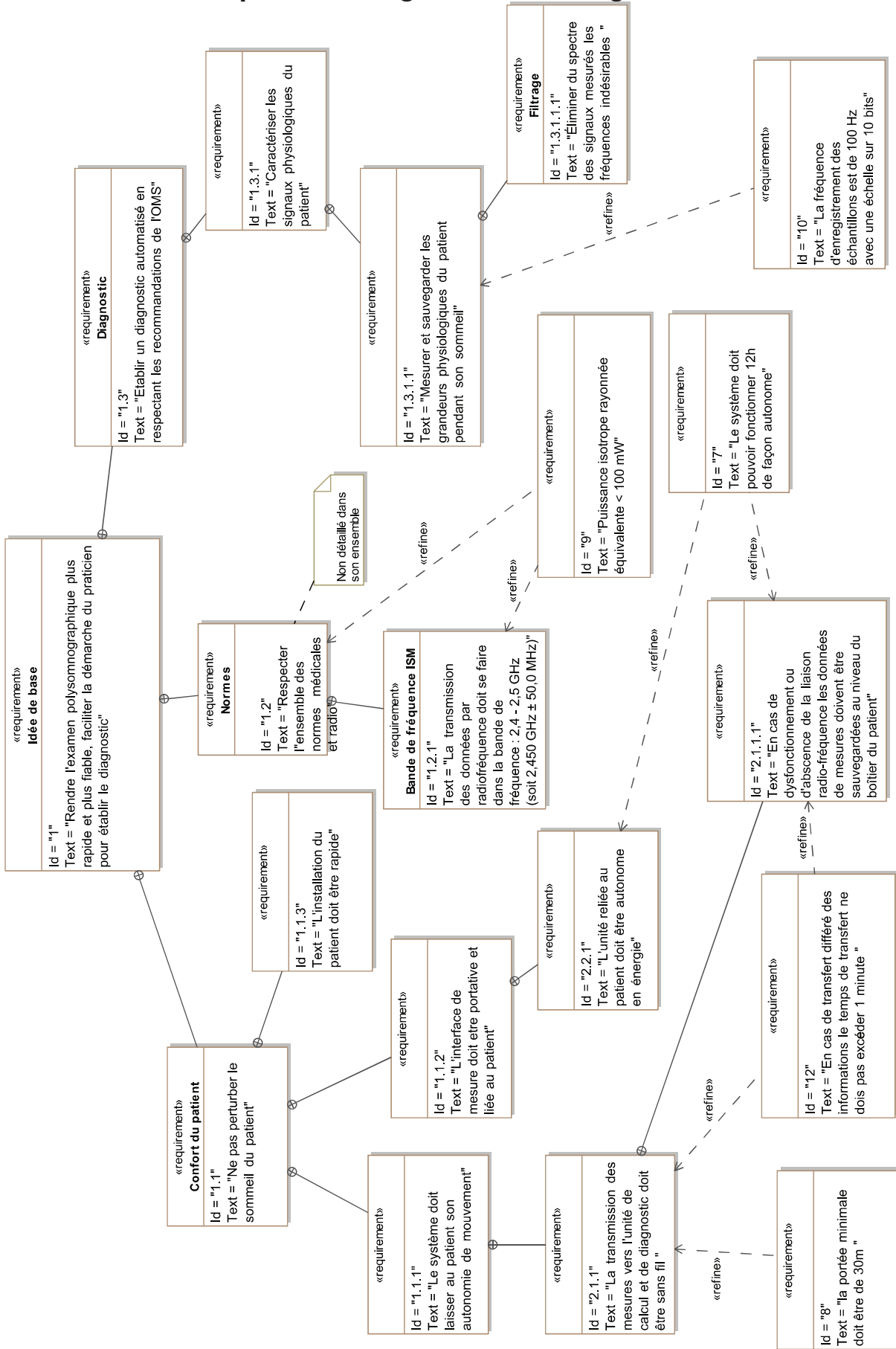
Ce sujet s'inspire librement et reprend avec adaptations les figures 1 à 3, le document technique DT1 ainsi que les méthodes développées dans le manuscrit de thèse de doctorat en spécialité informatique intitulé « Fusion Symbolique et Données Polysomnographiques » présenté par Adrien Ugon à l'Université Pierre et Marie Curie (Paris 6) en 2013. Références NNT : 2013PA066187 et tel-01160772.

Document technique DT1. Règles de fusion symbolique

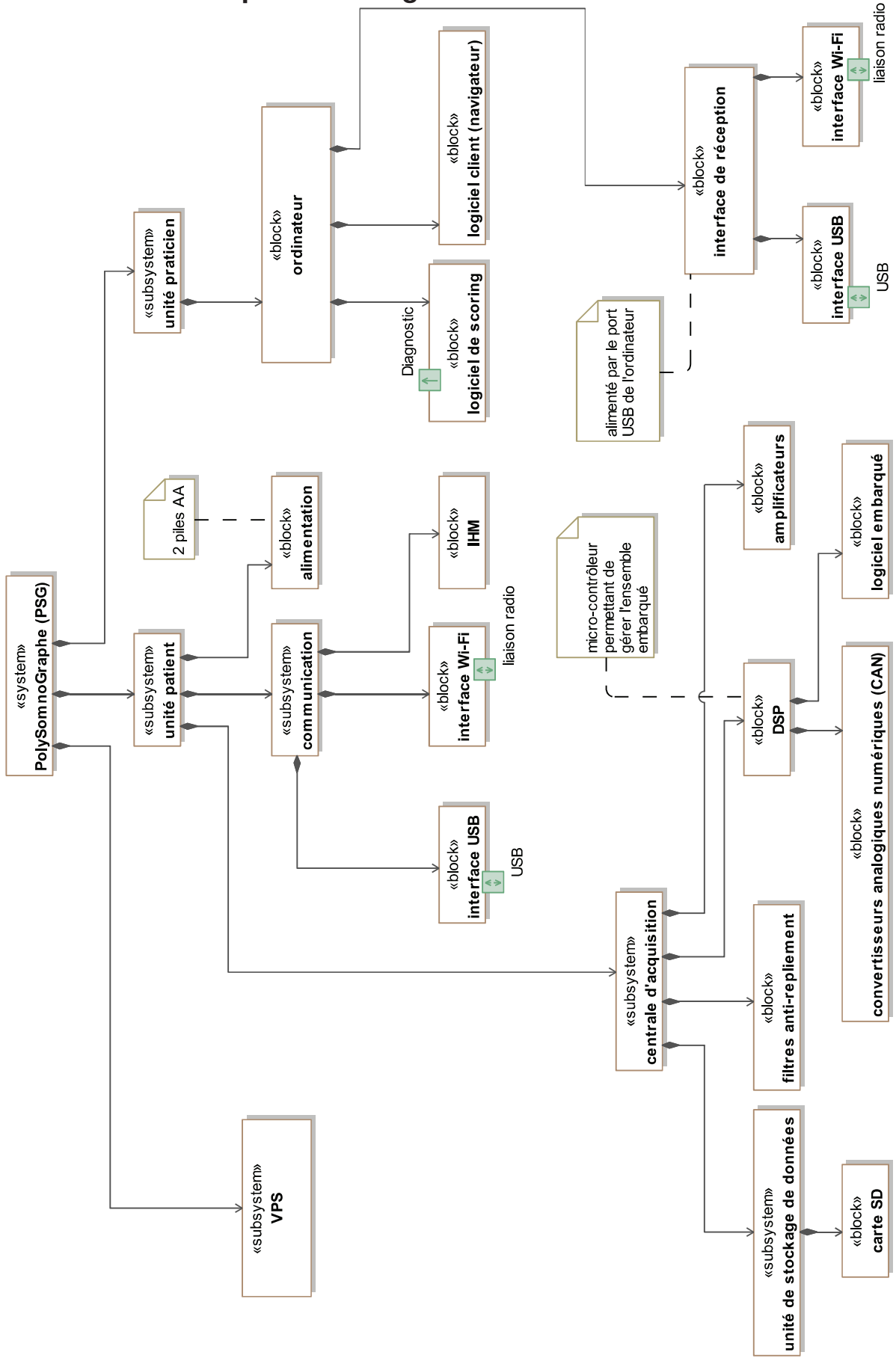
<p style="text-align: center;">OU</p> <p>La classe X est vraie si (A = « ValueA ») OU (B = « ValueB ») OU (C = « ValueC »).</p>	 <pre>graph TD; A[A] -- ValueA --> X[X]; B[B] -- ValueB --> X; C[C] -- ValueC --> X;</pre>
<p style="text-align: center;">ET</p> <p>La classe Y est vraie si (D = « ValueD ») OU ((E = « ValueE ») ET (F = « ValueF »)).</p>	 <pre>graph TD; D[D] -- ValueD --> Y[Y]; E[E] -- ValueE --> J(()); F[F] -- ValueF --> J; J --> Y;</pre>
<p style="text-align: center;">Majoritaire</p> <p>La classe Z est vraie si il existe au moins 2 variables de valeur « ValueZ » parmi les paramètres G, H, I.</p>	 <pre>graph TD; H[H] --> T((2)); G[G] --> T; I[I] --> T; T -- ValueZ --> Z[Z];</pre>



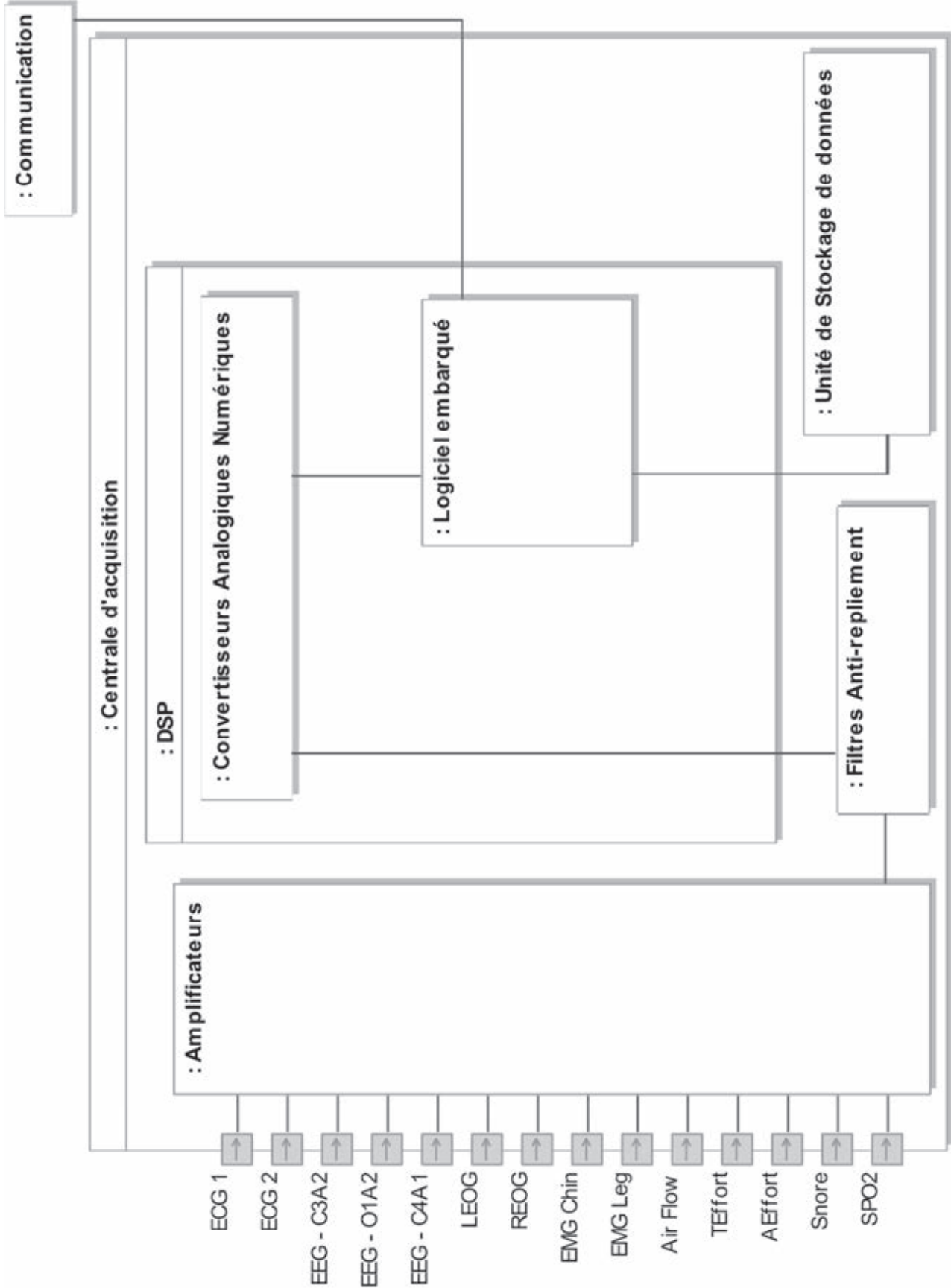
Document technique DT2. Diagramme des exigences



Document technique DT3. Diagramme de définition de blocks



Document technique DT4. Diagramme de bloc interne « unité patient »



Document technique DT5. Spécifications Wi-Fi et USB

La norme IEEE 802.11-2012 définit les débits suivants pour le standard Wi-Fi :

Table 9-5—Non-HT reference rate

Modulation	Coding rate (R)	Non-HT reference rate (Mb/s)
BPSK	1/2	6
BPSK	3/4	9
QPSK	1/2	12
QPSK	3/4	18
16-QAM	1/2	24
16-QAM	3/4	36
64-QAM	1/2	48
64-QAM	2/3	48
64-QAM	3/4	54
64-QAM	5/6	54

La spécification USB 3.1-r1.0 indique les débits nominaux suivants :

1.1 Background

The original Universal Serial Bus (USB) was driven by the need to provide a user-friendly plug-and-play way to attach external peripherals to a Personal Computer (PC). USB has gone beyond just being a way to connect peripherals to PCs. Printers use USB to interface directly to cameras. Mobile devices use USB connected keyboards and mice. USB technology commonly finds itself in automobiles, televisions, and set-top boxes. USB, as a protocol, is also being picked up and used in many nontraditional applications such as industrial automation. And USB as a source of power has become the mobile device charging solution endorsed by international communities across the globe.

Initially, USB provided two speeds (12 Mbps and 1.5 Mbps) that peripherals could use. As PCs became increasingly powerful and able to process larger amounts of data, users needed to get more and more data into and out of their PCs. This led to the definition of the USB 2.0 specification in 2000 to provide a third transfer rate of 480 Mbps while retaining backward compatibility. By 2006, two things in the environment happened: the transfer rates of HDDs exceeded 100MB/s, far outstripping USB 2.0's ~32MB/s bandwidth and the amount of digital content users were creating was an ever increasing pace. USB 3.0 was the USB community's response and provided users with the ability to move data at rates up to 450MB/s while retaining backward compatibility with USB 2.0.

Now, with the continued trend for more bandwidth driven by larger and faster storage solutions, higher resolution video, and broader use of USB as an external expansion/docking solution, USB 3.1 extends the performance range of USB up to 1GB/s by doubling the SuperSpeed USB clock rate to 10Gbps and enhancing data encoding efficiency.

Document technique DT6. Caractéristiques du DSP et du compilateur

TMS320C28346, TMS320C28345, TMS320C28344
TMS320C28343, TMS320C28342, TMS320C28341



SFRS516D—MARCH 2009—REVISED AUGUST 2012

www.ti.com

Delfino Microcontrôles

Check for Samples: TMS320C28346, TMS320C28345, TMS320C28344, TMS320C28343, TMS320C28342, TMS320C28341

1 TMS320C2834x (Delfino™) MCUs

1.1 Overview

The TMS320C2834x (C2834x) Delfino™ microcontroller (MCU) devices build on TI's existing F2833x high-performance floating-point microcontrollers. The C2834x delivers up to 300 MHz of floating-point performance, and has up to 516KB of on-chip RAM. Designed for real-time control applications, the C2834x is based on the C28x™ core, making it code-compatible with all C28x microcontrollers. The on-chip peripherals and low-latency core make the C2834x an excellent solution for performance-hungry real-time control applications.

1.2 Features

- High-Performance Static CMOS Technology
 - Up to 300 MHz (3.33-ns Cycle Time)
 - 1.1-V/1.2-V Core, 3.3-V I/O, 1.8-V PLL/Oscillator Design
- High-Performance 32-Bit CPU (TMS320C28x)
 - IEEE-754 Single-Precision Floating-Point Unit (FPU)
 - 16 x 16 and 32 x 32 MAC Operations
 - 16 x 16 Dual MAC
 - Harvard Bus Architecture
 - Fast Interrupt Response and Processing
 - Code-Efficient (in C/C++ and Assembly)

Table 6-1. TMS320C28x C/C++ Data Types

Type	Size	Representation	Range	
			Minimum	Maximum
char, signed char	16 bits	ASCII	-32 768	32 767
unsigned char, _Bool	16 bits	ASCII	0	65 535
short	16 bits	2s complement	-32 768	32 767
unsigned short	16 bits	Binary	0	65 535
int, signed int	16 bits	2s complement	-32 768	32 767
unsigned int	16 bits	Binary	0	65 535
long, signed long	32 bits	2s complement	-2 147 483 648	2 147 483 647
unsigned long	32 bits	Binary	0	4 294 967 295
long long, signed long long	64 bits	2s complement	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
unsigned long long	64 bits	Binary	0	18 446 744 073 709 551 615
enum ⁽¹⁾	16 bits	2s complement	-32 768	32 767
float	32 bits	IEEE 32-bit	1.19 209 290e-38 ⁽²⁾	3.40 282 35e+38
double	32 bits	IEEE 32-bit	1.19 209 290e-38 ⁽²⁾	3.40 282 35e+38
long double	64 bits	IEEE 64-bit	2.22 507 385e-308 ⁽²⁾	1.79 789 313e+308
pointers ⁽³⁾	32 bits	Binary	0	0xFFFFFFFF

⁽¹⁾ For details about the size of an enum type, see Section 6.4.1.

⁽²⁾ Figures are minimum precision.

⁽³⁾ Even though pointers are 32-bits, the compiler assumes that the addresses of global variables and functions are within 22-bits.

Nom de famille :

(Suivi, s'il y a lieu, du nom d'usage)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



Prénom(s) :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Numéro
Inscription :**

--	--	--	--	--	--	--	--	--	--	--	--	--	--

Né(e) le :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

(Remplir cette partie à l'aide de la notice)

Concours / Examen : **Section/Sécialité/Série :**

Epreuve : **Matière :** **Session :**

CONSIGNES

- Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.
- Ne pas signer la composition et ne pas y apporter de signe distinctif pouvant indiquer sa provenance.
- Numéroter chaque PAGE (cadre en bas à droite de la page) et placer les feuilles dans le bon sens et dans l'ordre.
- Rédiger avec un stylo à encre foncée (bleue ou noire) et ne pas utiliser de stylo plume à encre claire.
- N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.

EAE SIN 2

Document réponse

NE RIEN ECRIRE DANS CE CADRE

Document réponse DR1. Étude spectrale de la chaîne d'acquisition

