

Projet électronique 2 - 2025

Préambule :

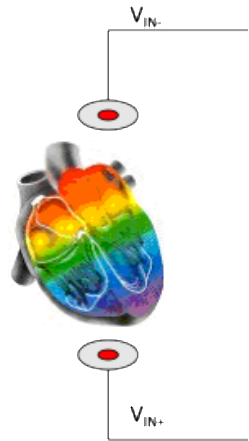
- Ce document constitue la version 1 du projet. Il n'y aura aucune modification des objectifs, mais en fonction de vos questions, il se peut que je doive le réactualiser (détails manquants, clarification, ...).
- Vous pouvez me soumettre des questions par mail jusqu'au dernier moment, mais le rendu est prévu au plus tard le dimanche 26 janvier 2025 à 23:59:59.
- Les différents modules proposés (cinq en tout) peuvent être analysés en totale indépendance, cependant dans une analyse préliminaire vous devez caractériser l'amplificateur opérationnel qui sera exploité dans la suite.
- Délivrable :
 - Outre le rapport, tous les schémas de simulation doivent être inclus dans le dossier (sous forme de fichiers).
 - Prenez soin de ne mettre que le strict nécessaire
- Recommandations rapport :
 - Dans le rapport, vous devez inclure l'image du schéma édité dans le simulateur et le résultat de la simulation afin que le commentaire soit facile à interpréter.
 - La liste complète des fichiers de simulation doit être intégrée dans le rapport en annexe.
 - Faites un plan !!! Cela permet au premier coup d'œil de comprendre ce que vous allez présenter
 - Vous pouvez vous inspirer de la structure des questions, mais vous êtes libres d'adopter le plan que vous estimez pertinent.
 - Faire une conclusion générale : Outre le bilan du projet, estimez le temps global que vous avez passé pour ce projet. Le travail global est estimé entre vingt et trente heures incluant : réflexion, lecture, calculs, simulations et rédaction. Bien mettre en évidence les problèmes que vous avez rencontrés.

Projet électronique 2 - 2025

1. Le contexte du projet

Le projet proposé est associé au domaine médical. Plus spécifiquement, il a pour objectif la détection des arythmies cardiaques.

Le cœur est un muscle dont l'activité électrique peut être mesurée

 <p>Figure 1 : le cœur</p>	<p>La tension est mesurée autour du cœur par des électrodes fixées sur la peau (au minimum deux comme dans le projet proposé)</p> <p>V_0 est la tension moyenne du corps que l'on doit éliminer et l'on va détecter des variations ΔV_{IN} entre les deux électrodes</p> <ul style="list-style-type: none">• $V_{IN-} = V_0 - \Delta V_{IN}$• $V_{IN+} = V_0 + \Delta V_{IN}$ <p>Comme d'habitude le système est sensible au bruit qu'il faudra éliminer</p> <ul style="list-style-type: none">• Basse fréquence (en principe des fréquences proches de 0, courants et tensions d'offset)• Haute fréquence (principalement la lumière des néons) supérieure à 100 Hz
---	--

Un système complet est trop complexe à étudier, aussi, le projet propose quelques modules à examiner, y compris l'écriture d'un algorithme (simplification avec nombres entiers ou pour les courageux nombres réels). Dans le schéma ci-dessous, les cinq modules qui seront examinés sont explicitement notés en **rouge** :

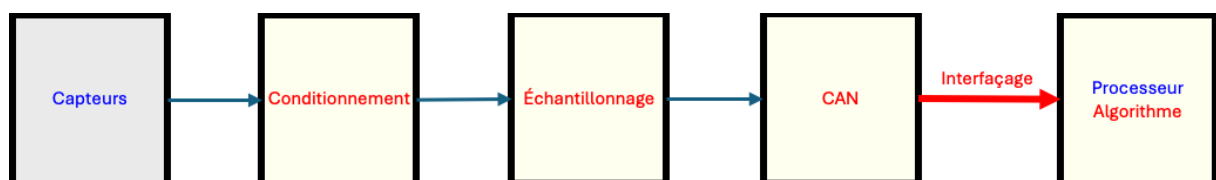


Figure 2 : Les modules étudiés

Les cinq modules étudiés représentent respectivement :

- La partie conditionnement du signal différentiel. Elle intégrera à la fois une amplification du signal et un filtrage passe-bande. L'analyse impliquera des calculs et une simulation
- L'échantillonnage du signal est un module assez simple en termes de nombres de composants, cependant, une analyse fine accompagnée d'une simulation seront nécessaires pour dimensionner correctement les composants.
- La conversion analogique digitale exploitera le convertisseur à approximations successives. Compte tenu de la complexité du module, l'analyse sera limitée à sa conception logique.
- L'interface entre le convertisseur et le processeur est assez simple, et ressemble au principe de l'échantillonnage, nécessitant une analyse fine et une simulation.
- L'informatique (pour le traitement des informations) et l'électronique sont aujourd'hui indissociables. L'algorithme de détection des arythmies est bonne illustration de cette cohabitation. Une solution analogique pourra être « abordée » pour comparer les deux approches numériques et analogiques.

Projet électronique 2 - 2025

2. Mesurande

Pour le calcul de l'ECG: on ne s'intéresse qu'aux pulsations cardiaques et non au profil. On veut principalement **relever le temps** entre deux battements.

Premières contraintes :

On **détecte un battement** lorsqu'un seuil est dépassé.

Pour éviter que le seuil soit comptabilisé plusieurs fois lors d'un même cycle, on estime que l'écart entre deux mesures doit être supérieur à 100ms. En effet un temps plus court supposerait que le cœur batte plus vite que 10 Hz soit 600 battements à la minute. Pour rappel, le nombre de battements par minute se calcule avec le nombre de battements par seconde (inverse de la valeur de fréquence) soit $\frac{1}{\Delta T} * 60$

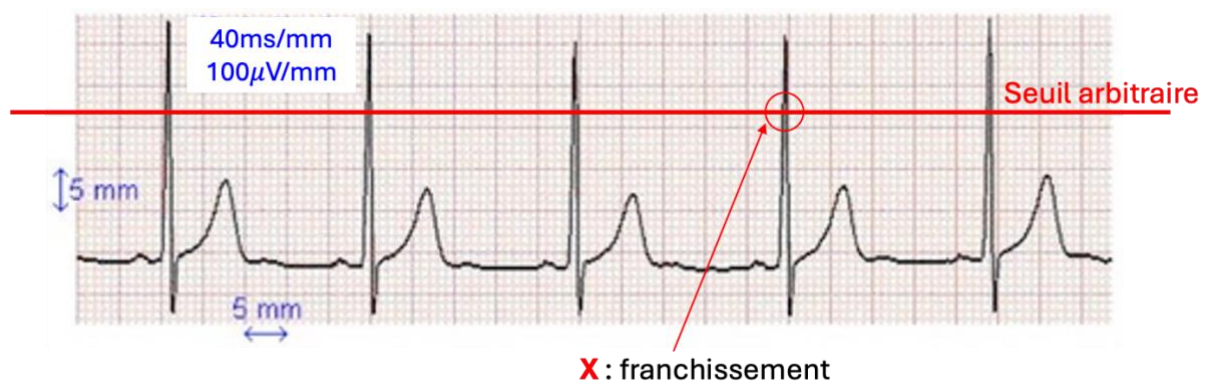


Figure 3 : profil typique d'un ECG à 60 battements par minute

X : Au franchissement du seuil, on détecte un écart de 10 ms entre les flans montant et descendant du signal. Pour respecter le **théorème de Shannon** il faudrait un échantillonnage minimum à 200 Hz pour être sûr de détecter le franchissement du seuil. Cependant, afin d'améliorer la précision on **échantillonnera à 1KHz** (voir justification avec la seconde série de contraintes).

Si $\Delta T = 1s$ (60 battements à la minute) on pourra alors détecter des écarts de 1 ms ce qui constitue une précision suffisante pour l'évaluation des arythmies.

Secondes contraintes :

On fixe l'étendue de mesure avec les limites suivantes :

- Cœur le plus lent : 30 battements par minute soit 0.5 Hz
- Cœur le plus rapide : 300 battements par minute soit 5 Hz
- Pour être en mesure d'analyser un profil (qui n'est pas demandé dans le projet) et un temps précis entre deux battements, on souhaite prélever au minimum 100 échantillons. Sachant que la durée la plus rapide entre deux battements est de 200ms (5Hz), on en déduit que la période d'échantillonnage pourrait être fixée à 2 ms

Conclusion, le choix de 1 KHz pour la fréquence d'échantillonnage est un bon compromis.

D'autre part, pour disposer d'une bonne précision, on fixera la mesure sur 16 bits soit 65'536 niveaux possibles

Projet électronique 2 - 2025

3. Les modules étudiés

A. Préambule : Caractérisation de l'AOP LT1356

Cet amplificateur sera l'élément de base pour le module de conditionnement qui suivra. Il faut donc le caractériser et observer ses défauts à partir de plusieurs simulations. Compte tenu de l'usage de l'AOP, tous les défauts ne sont pas forcément significatifs.

Utilisez un schéma de type **suiveur** (vu lors du cours AOP1) pour caractériser les trois premiers défauts. La présence de la résistance sur la borne V+ n'est pas nécessaire.

1. L'un des éléments évoqués lors du cours AOP1, faisait apparaître une limitation de courant de sortie de l'AOP. Imaginez l'expérience qu'il faudrait mener pour mettre en évidence cette limitation de courant et essayez de la simuler.
2. Un autre défaut significatif est observé avec les courants de "fuite" sur les entrées V+ et V-. Quantifiez ces courants de fuite en mettant une source de tension variable sur l'entrée V+.
3. Mettre sur l'entrée V+ un signal carré avec une montée de 1 ns et observez à quelle vitesse le signal de sortie réagit. Cette caractéristique s'appelle le Slew-Rate et se calcule en $V/\mu s$
4. En option, on peut aussi mesurer la bande passante de cet amplificateur. Ne connaissant pas le gain en boucle ouverte, je vous propose d'éditer sur LTSpice le schéma de l'**amplificateur inverseur** en imposant un gain de -100. Utilisez une source de tension de 50 mV d'amplitude. Faites varier la fréquence et vérifiez à la simulation l'évolution du gain. Lorsqu'il commencera à chuter, vous pourrez en déduire que vous avez atteint la limite de la bande passante pour ce gain de -100.

B. Circuit de conditionnement :

Le circuit qu'il faut concevoir est modélisé ci-dessous

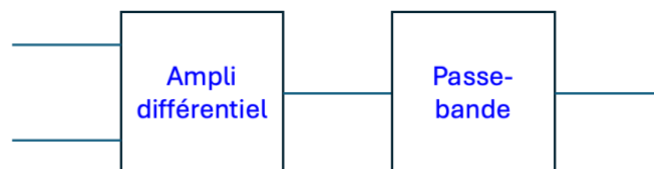


Figure 4 : Module de conditionnement

Vous êtes totalement libres de proposer votre choix qui tient compte de peu de contraintes :

- L'étendue de mesure pour le signal ΔV_{IN} (voir figure 1) représente quelques mV (un mm représente $100 \mu V$) alors qu'on veut générer des V. Il faut donc amplifier d'un facteur 200 minimum.
- Le bruit basse fréquence est assimilable à des valeurs continues d'offset
- Le bruit haute fréquence est liée à la lumière ambiante des néons dont les fréquences aujourd'hui dépassent 100 Hz

On demande donc de concevoir ce module et de dimensionner les composants.

Tous les choix doivent être commentés !!!

Simuler en mode **.Trans** pour vérifier le comportement du montage.

Projet électronique 2 - 2025

Remarque : Vous pouvez ajouter une source de tension virtuelle pour générer du bruit à haute fréquence (un sinus à 100 Hz) pour observer son atténuation.

C. Circuit d'échantillonnage/blocage :

Le modèle proposé en cours est tout à fait suffisant :

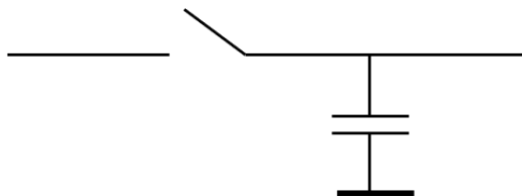


Figure 5 : module d'échantillonnage simplifié

L'interrupteur sera réalisé à l'aide d'une implémentation CMOS (NMOS et PMOS en parallèle)
On impose une valeur de 1 nF pour la capacité.

Sachant que l'échantillonnage est fixé à 1 ms, il faut prévoir :

- Un signal de commande de fermeture de l'interrupteur très court (10 μs maximal) permettant de disposer de 990 μs pour convertir le signal analogique en numérique.
- Il faut dimensionner correctement les transistors NMOS et PMOS de l'interrupteur pour que la valeur analogique puisse être disponible sur la capacité.

Tous les choix doivent être commentés !!!

La simulation de type **.Trans** doit prouver la pertinence des choix. Placez une source de tension à l'entrée du circuit avec une sinusoïde de quelques volts d'amplitude et une résistance de sortie de 10 ohms. Cette source symbolise la sortie de l'étage précédent)

Prélevez lors de la simulation une dizaine d'échantillons de la valeur sinusoïdale (vérifiez les niveaux de tension stables qui doivent durer 990 μs).

D. Circuit de conversion :

Le modèle proposé est celui des approximations successives présenté en cours.

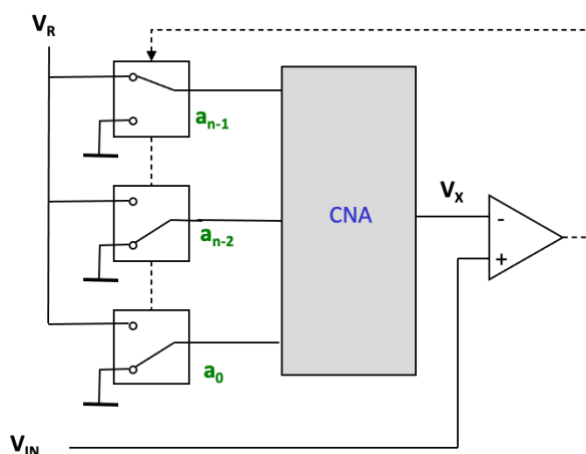


Figure 6 : convertisseur AN à approximations successives

Projet électronique 2 - 2025

Une analyse électrique complète du montage est impossible dans le cadre d'un mini-projet. L'analyse ici s'apparente davantage à de la conception de portes logiques.

Rappel du fonctionnement.

- A l'initialisation du système, tous les switches sont sur la position 0.
- Un jeton est injecté dans la cellule de poids le plus élevé pour indiquer que le poids le plus élevé est analysé. Le switch associé à ce poids est positionné sur l'entrée V_R .
- Si la valeur issue du convertisseur CNA est plus petite que le signal V_{IN} (signal analogique à convertir) le switch est maintenu à la position V_R . Dans le cas contraire on remet le switch sur la position 0.
- On se déplace sur la cellule de poids suivant et le processus recommence de cellule en cellule. Remarque : On doit donc disposer d'un registre à décalage permettant d'identifier la cellule analysée.

On demande de concevoir les blocs fonctionnels du montage avec des portes logiques sans simulation. Il faut évidemment justifier vos choix.

E. Circuit d'interfaçage entre convertisseur et processeur :

Le modèle proposé est très simplifié.

La sortie du convertisseur est supposée être une valeur numérique avec des niveaux de tensions comparables à ceux utilisés dans le processeur. Par exemple 0V pour le 0 logique et 5V pour le 1 logique.

L'objectif est de forcer la valeur numérique du convertisseur dans le port d'entrée du processeur.

Le schéma suivant donne une vision assez réaliste

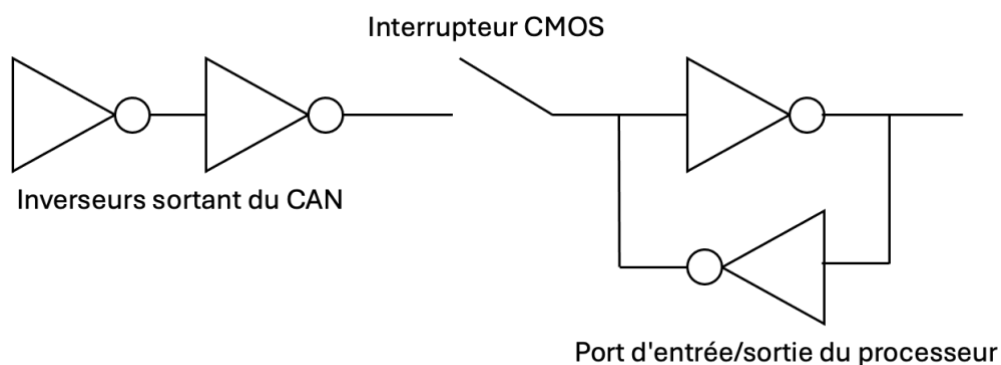


Figure 7 : interfaçage entre le CAN et le processeur

Le circuit analysé est composé de trois blocs :

- L'**interrupteur CMOS** : Il doit permettre de passer l'information numérique du convertisseur vers le processeur. Le principe est le même que pour l'échantillonnage/blocage. La commande de l'interrupteur est identique, mais le dimensionnement « pourrait » être différent.
- Le **port d'entrée/sortie** du processeur est composé de deux inverseurs bouclés. Ce principe très largement utilisé dans les RAM statiques permet de maintenir un niveau

Projet électronique 2 - 2025

logique tant que le circuit est alimenté. On propose de prendre des tailles fixes pour les deux inverseurs :

- NMOS avec $W = 100$ nm et $L = 20$ nm
- PMOS avec $W = 300$ nm et $L = 20$ nm
- Les **inverseurs** sortant du convertisseur : Ils ne sont pas forcément limités à deux inverseurs et on vous demande d'en mettre autant que nécessaire pour garantir le basculement du port d'entrée/sortie. A noter que les transistors des inverseurs sont de plus en plus gros (un facteur 3 est généralement imposé à chaque couche d'inversion). On propose de prendre pour la première couche les mêmes tailles que pour le port d'entrée/sortie.

Essayez de justifier :

- Pourquoi le PMOS est trois fois plus large que le NMOS (idem question C.)
- Pourquoi la taille augment d'un facteur trois entre les différentes couches d'inversion

Proposez une solution qu'il faudra impérativement simuler en mode **.Trans**

Bien entendu, il faudra vérifier lors de la simulation que le port d'entrée/sortie bascule du niveau haut vers le niveau bas et inversement.

Remarque : Si vous avez le temps et pour avoir une vision plus réaliste, vous pouvez ajouter des capacités de part et d'autre de l'interrupteur. Ces capacités sont associées aux lignes qui sont souvent assez longues.

F. Algorithme de détection du niveau d'arythmie :

1. Principes de calculs (ECG)

L'objectif est de **fournir en temps réel** une évaluation de l'arythmie cardiaque sous forme **d'écart type** qui est un indicateur très fiable.

Rappel calcul écart-type :

La détection des arythmies cardiaques est basée sur un calcul d'écart type dont la formulation est la suivante :

$$\sqrt{\sum_{i=0}^{N-1} \frac{1}{N} (X_i - \bar{X})^2}$$

- \bar{X} : Cette valeur représente la moyenne des écarts entre impulsions. Cette valeur est toujours calculée durant la série précédente des 32 battements et elle est exploitée dans la série courante pour calculer l'écart type.
- X_i : Cette valeur représente l'écart courant entre deux impulsions

Une solution précise : elle consisterait à calculer à la fois la valeur moyenne et l'écart type de la série courante. Mais ces deux calculs ne pourraient s'effectuer qu'en fin de série, générant un temps mort important avant de relancer une nouvelle série. Vous pourriez évaluer la durée de ce temps mort. Le nombre de registres augmenterait aussi car il faudrait mémoriser tous les écarts X_i pour les reproduire lors du calcul de la valeur moyenne et de l'écart type.

Projet électronique 2 - 2025

Solution retenue : Si on veut éviter les temps morts, le calcul ne peut pas s'effectuer au rythme de la réception des valeurs de l'ECG car l'écart-type est issu d'une analyse statistique sur N échantillons. Il faut donc trouver un bon compromis entre le nombre d'échantillons analysés et le temps de production d'un résultat

On propose d'effectuer une première série de 32 échantillons qui ne permet pas de calculer l'écart type, simplement de fournir la première valeur moyenne. Cependant, on doit admettre que le résultat ne serait pas aussi précis, la valeur moyenne pouvant significativement changer lors de la seconde série (mouvements, émotion, ...) !!! C'est le prix à payer pour avoir une détection d'arythmie en temps réel.

Le calcul de l'arythmie s'effectuera en continu, tous les 32 échantillons. Pourquoi 32 échantillons ?

- On peut estimer que des séries de 32 battements permettent déjà de faire un bon diagnostic (32 battements correspondent à 64 secondes pour un cœur très lent battant à 30 battements par minute ou 6.4 secondes pour une cadence extrême à 300 battements par minute).
- La valeur de 32 battements est une puissance de 2, ce qui simplifie le calcul de la moyenne ($Moyenne = \frac{\text{Somme des écarts}}{32}$) or diviser par 32 revient à faire un décalage de 5 positions ce qui nous affranchit de l'usage d'un diviseur ou de l'émulation algorithmique de la division.
- Si on veut obtenir au minimum 100 échantillons entre deux battements, cela signifie que pour le cœur le plus rapide (battant à 5Hz), il faut travailler à une fréquence d'échantillonnage de 500 Hz. Or on a choisi par option de simplicité, une fréquence de 1 kHz.

On propose donc d'adopter le protocole suivant :

Tout calcul d'écart type est précédé d'une phase de calibration de **32 battements**. Sur ces 32 battements, la valeur moyenne est facile et rapide à calculer. Cette valeur moyenne est utilisée pour calculer l'écart type lors de la phase suivante. A chaque nouvelle phase on recalcule une nouvelle valeur moyenne exploitée dans la phase suivante pour le calcul de l'écart type comme l'illustre la figure suivante :

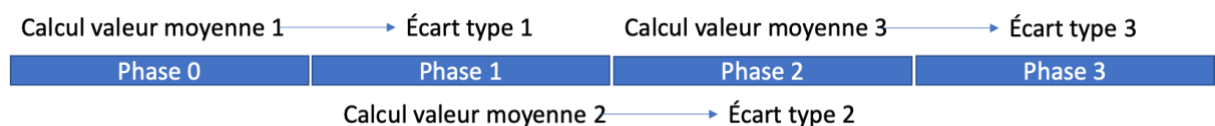


Figure 8 : alternance calcul écart-type et calcul de valeur moyenne

Le calcul de l'écart-type est **très gourmand en ressources** car il faut des opérations de multiplication, division et surtout de racine carrée. On part du principe que vos algorithmes doivent être proches du matériel disponible dans un processeur pour pouvoir évaluer correctement le temps de calcul.

Ainsi, vous devez réaliser tous les calculs avec des opérateurs d'addition, soustraction, décalage (gauche, droite). Si vous êtes pris par le temps vous pouvez admettre qu'il existe des opérateurs de multiplication et de division, mais qui prennent plus de cycles que les

Projet électronique 2 - 2025

opérateurs de base. Par exemple pour une multiplication sur N bits, il faudra N cycles, alors que les opérateurs de bases ne nécessitent qu'un seul cycle.

Il existe de nombreuses méthodes algorithmiques pour calculer la racine carrée (consulter internet).

Quelques remarques si vous avez un peu de temps.

Remarque 1 : Vous pouvez calculer l'écart type en parallèle avec la moyenne qui sera exploitée dans la phase suivante. Il faut donc penser comment entrelacer les calculs (chose simple en multithreading ou en calculs parallèles)

Remarque 2 : Si vous avez le temps, il serait intéressant de fournir un ordre de grandeur en termes de cycles de calculs de l'écart-type.

Remarque 3 : quand vous avez des tests (principalement des comparaisons), le temps de calcul est comparable à une opération de base. En effet, $A < B$ s'obtient en calculant $A - B$ puis en analysant le signe du résultat. S'il y a plusieurs conditions dans un test il faut autant de cycles que de conditions (Ex : $(A < B) \ \&\& \ (C >= D)$)