وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبو بكر بلقايد– تلمسان

Université ABOUBEKR BELKAID – TLEMCEN

كلية التكنولوجيا

Faculté de TECHNOLOGIE

قسم الهندسة بيوطبية

Département Génie biomédical

MÉMOIRE

Présenté par :

*Bouguern Ikram*

*En vue de l'obtention du*

**Diplôme de MASTER**

En Génie biomédical

***Spécialité : Électronique et maintenance biomédicale***

# Développement d'un stéthoscope électronique intelligent avec classification automatique des murmures cardiaques (IA embarquée)

Soutenu le 21/06/2026, devant le jury composé de :

| | | | |
|---|---|---|---|
| **M. Kholkhal Mourad** | **MCA** | **Univ.Tlemcen** | **Encadrant** |
| **Mme. RERBAL Souhila** | **MCA** | **Univ.Tlemcen** | **Président** |
| **Mme. HEDEIL Nawal** | **Professeur** | **Univ.Tlemcen** | **Examinateur** |

**Année universitaire 2025/2026.**

# Remerciements

Avant toute chose, je rends grâce à **Allah le Tout-Puissant** pour m'avoir accordé la force, la persévérance et les capacités nécessaires afin de mener à bien ce travail. Sa bénédiction m'a accompagné tout au long de ce parcours.

Je souhaite exprimer ma profonde gratitude à **Monsieur KHOLKHAL Mourad**, mon encadrant, pour son accompagnement constant, son écoute attentive et la confiance qu'il m'a témoignée. Ses orientations éclairées, ses remarques constructives et son engagement ont constitué un soutien précieux dans l'élaboration de ce mémoire.

Mes remerciements les plus sincères s'adressent également à **Madame RERBAL Souhila**, Présidente du jury, pour l'honneur qu'elle me fait de présider ce jury. Je remercie chaleureusement **Madame HEDEIL Nawal** d'avoir accepté d'examiner ce travail et d'enrichir ma réflexion par ses précieuses observations.

J'adresse également mes remerciements à l'ensemble des enseignants et responsables pédagogiques de l'**Université Aboubakr Belkaïd – Tlemcen**. Grâce à la qualité de leur enseignement, à leur dévouement et au partage de leurs connaissances, j'ai pu acquérir les compétences nécessaires à ma formation académique et professionnelle.

Enfin, j'exprime ma reconnaissance à toutes les personnes qui, par leur soutien moral, leurs encouragements ou leur aide, ont contribué de près ou de loin à l'aboutissement de ce travail. Qu'elles trouvent ici l'expression de ma sincère gratitude.

# Dédicaces

Je dédie ce mémoire :

**À ma très chère mère**
Pour ton amour inconditionnel, tes prières incessantes et tes sacrifices sans nombre. Tu as toujours été mon refuge et ma plus grande force. Que ce travail soit la modeste preuve de mon éternelle reconnaissance.

**À mon cher père**
Pour ton soutien indéfectible, ta sagesse et ta confiance. Tu m'as appris la persévérance et l'honnêteté. Puisse ce mémoire te rendre fier.

**À mes frères bien-aimés**
Pour votre complicité, vos encouragements et votre présence rassurante à chaque étape de ma vie. Merci d'avoir toujours cru en moi.

**À mes amies et collègues**
Pour les moments de partage, l'entraide et les rires qui ont illuminé ce long parcours universitaire. Votre soutien a été un véritable réconfort.

**À mes cousines**
Pour votre tendresse, votre bienveillance et les souvenirs joyeux qui m'ont rappelé l'essentiel. Merci d'avoir été présentes à votre manière.

Ce mémoire est l'aboutissement d'un chemin exigeant, rendu possible grâce à l'amour et au soutien indéfectible de tous mes proches. Qu 'Allah vous bénisse et vous protège tous .

# Resume

Le développement d'un stéthoscope électronique intelligent capable de classifier automatiquement les souffles cardiaques représente une avancée majeure pour le diagnostic précoce en cardiologie. Ce projet de fin d'études a consisté à concevoir un dispositif portable et autonome intégrant une intelligence artificielle embarquée (TinyML). Le système repose sur un microphone à électret (module KY-038) couplé à une cloche acoustique, dont le signal est numérisé par un microcontrôleur ESP32 à 200 Hz. Un prétraitement en temps réel (centrage, filtrage médian d'ordre 5, contrôle automatique de gain) est appliqué, puis trois descripteurs temporels sont extraits par fenêtres glissantes de 100 échantillons : le taux de passage par zéro (ZCR), l'aplatissement (kurtosis) et l'asymétrie (skewness). Un modèle Random Forest de 25 arbres, pré-entraîné sur la base de référence Physio Net Challenge 2016 (exactitude de 94,2 %), a été converti en code C++ natif via l'outil EloquentTinyML et intégré dans l'ESP32. Le diagnostic binaire (normal / pathologique) est affiché sur un écran OLED et transmis par liaison série à une interface Python. Les tests réalisés sur 5 sujets (3 sains, 2 pathologiques) ont donné une exactitude de 80 %, une sensibilité de 100 % et une spécificité de 66,7 %, avec une consommation de 80 mA sous 5 V (soit un appel de courant de **127 mA** sur la batterie 3,7 V après conversion), et une autonomie estimée à **15 heures** en conditions réelles d'utilisation. Ce prototype valide la faisabilité d'un stéthoscope numérique portable, économique et doté d'une intelligence embarquée pour l'aide au diagnostic cardiaque en milieu clinique ou en télé médecine.

**Mots clés** : Stéthoscope électronique, intelligence artificielle embarquée (TinyML), Random Forest, classification de souffles cardiaques, ESP32, traitement du signal PCG

# Abstract

The development of an intelligent electronic stethoscope capable of automatically classifying heart murmurs represents a significant advance for early diagnosis in cardiology. This final year project consisted of designing a portable, standalone device integrating embedded artificial intelligence (TinyML). The system uses an electret microphone (KY-038 module) coupled with an acoustic chest piece, whose signal is digitized by an ESP32 microcontroller at 200 Hz. Real-time preprocessing (centering, 5-point median filtering, automatic gain control) is applied, then three time-domain features are extracted over sliding windows of 100 samples: zero-crossing rate (ZCR), kurtosis, and skewness. A 25-tree Random Forest model, pre-trained on the Physio Net Challenge 2016 reference database (94.2% accuracy), was converted to native C++ code using the EloquentTinyML tool and embedded into the ESP32. The binary diagnosis (normal / pathological) is displayed on an OLED screen and transmitted via serial link to a Python interface. Tests on 5 subjects (3 healthy, 2 pathological) yielded 80% accuracy, 100% sensitivity and 66.7% specificity, with a current consumption of 80 mA at 5 V (corresponding to a battery draw of **127 mA** at 3.7 V after DC-DC conversion), and an estimated battery life of **15 hours** under real operating conditions. This prototype validates the feasibility of a portable, low-cost digital stethoscope with on-board intelligence to assist cardiac diagnosis in clinical or telemedicine settings.



**Keywords**: Electronic stethoscope, embedded artificial intelligence (TinyML), Random Forest, heart murmur classification, ESP32, PCG signal processing.

## الملخص

.(TinyML) المبكر لأمراض القلب. تم في هذا المشروع تصميم جهاز محمول ومستقل يدمج ذكاءً اصطناعياً مضموماً مقترن بغطاء صوتي، حيث تتم رقمنة الإشارة بواسطة متحكم (KY-038 وحدة) يعتمد النظام على ميكروفون كهربائي بتردد 200 هرتز. تُطبق معالجة أولية في الزمن الحقيقي (توسيط، ترشيح وسيطي من الدرجة الخامسة، تحكم تلقائي ESP32 ، التفلطح(ZCR) في الكسب)، ثم يتم استخراج ثلاثة معاملات زمنية عبر نوافق انزلاقية من 100 عينة: معدل عبور الصفر المؤلف من 25 شجرة، والمدرَّب مسبقاً (Random Forest) نموذج الغابة العشوائية .(skewness) ، والالتواء(kurtosis) أصلي باستخدام C++ دقة 94.2%)، تم تحويله إلى كود) Physio Net Challenge 2016 على قاعدة البيانات المرجعية OLED يُعرض التشخيص الثنائي (طبيعي / مرضي) على شاشة .ESP32 ودمجه داخل المتحكم EloquentTinyML أداة أظهرت الاختبارات على 5 أشخاص (3 أصحاء، 2 مرضى) دقة 80%، .Python ويرسل عبر المنفذ التسلسلي إلى واجهة **مللي 127**ما يعادل سحب تيار )**مللي أمبير بجهد 5 فولت 80**وحساسية 100%، وخصوصية 66.7%، مع استهلاك تيار **ساعة** في ظروف الاستخدام الحقيقية. يثبت هذا **15أمبير** من البطارية 3.7 فولت بعد التحويل)، وعمر بطارية يقدر بـ النموذج الأولي جدوى تطوير سماعة طبية رقمية محمولة ومنخفضة التكلفة ومزودة بذكاء مضمن لدعم تشخيص أمراض .القلب في البيئة السريرية أو الطب عن بعد

**الكلمات المفتاحية**: سماعة طبية إلكترونية، (TinyML) ذكاء اصطناعي مضمن، غابة عشوائية، تصنيف نفخات القلب، PCG. ، معالجة إشاراتESP32

## TABLE DES MATIÈRES

**Remerciements** ..... i
**Dédicaces** ..... ii
**Résumé** ..... iii
**Abstract** ..... iv
**الملخص** ..... v
**Liste des figures** ..... vi
**Liste des tableaux** ..... vii
**Liste des abréviations** ..... x

**Introduction générale** .....

**Chapitre 1 – Anatomie et fonctionnement du cœur** .....
1.1. Définition du cœur ..... 1
1.1.1. Les oreillettes ..... 1
1.1.2. Les ventricules ..... 2
1.1.3. Les valves cardiaques ..... 3
1.2. Concept de la circulation systémique et pulmonaire ..... 5
1.2.1. La circulation pulmonaire (petite circulation) ..... 5
1.2.2. La circulation systémique (grande circulation) ..... 5
1.2.3. Rôle des valves dans le flux unidirectionnel ..... 6
1.3. Cycle cardiaque ..... 6
1.4. Les bruits cardiaques ..... 6
1.4.1. Le phonocardiogramme (PCG) ..... 7
1.4.2. Le premier bruit cardiaque (B1) ..... 7
1.4.3. Le deuxième bruit cardiaque (B2) ..... 8
1.4.4. Le troisième bruit cardiaque (S3 ou B3) ..... 8
1.4.5. Le quatrième bruit cardiaque (S4 ou B4) ..... 8
1.5. Souffle cardiaque ..... 9
1.5.1. Définition du souffle cardiaque ..... 9
1.5.2. Types de souffle ..... 10
1.5.3. Causes pathologiques fréquentes ..... 10
1.6. Structure et fonctionnement des stéthoscopes ..... 11
1.6.1. Stéthoscope classique (conventionnel) ..... 11
1.6.2. Stéthoscope électronique ..... 13
1.6.2.1. Principes et fonctionnement ..... 13
1.6.2.2. Chaîne de traitement du signal numérique ..... 14
1.6.3. Comparaison entre les systèmes conventionnel et électronique ..... 14
1.7. Problématique de l'interprétation humaine et nécessité de l'IA ..... 15
**Conclusion du chapitre 1** ..... 15

**Chapitre 2 – Conception matérielle du stéthoscope électronique** ..... 16
2.1. Introduction ..... 16
2.2. Architecture globale du système ..... 16
2.3. Alimentation électrique (batterie + convertisseur élévateur) ..... 17
2.3.1. Batterie rechargeable (Lithium-ion 18650) ..... 17
2.3.2. Unité de gestion d'énergie : module de charge et convertisseur step-up ..... 19
2.4. Bloc de transduction (microphone) ..... 22
2.4.1. Étude comparative des technologies de captation ..... 22
2.4.2. Mise en œuvre, connectique et réglage du gain ..... 26
2.4.3. Couplage acoustique (cloche) ..... 27
2.5. Bloc de numérisation par l'ESP32 ..... 28
2.5.1. Présentation de la carte ESP32-WROOM-32 ..... 29
2.5.2. Convertisseur analogique-numérique (ADC) intégré ..... 29
2.5.3. Résolution 12 bits et pas de quantification ..... 29
2.5.4. Choix de la fréquence d'échantillonnage (200 Hz) ..... 30
2.6. Interface utilisateur locale (écran OLED) ..... 30
2.6.1. Écran OLED 0,96 pouce soudé nativement ..... 31
2.6.2. Architecture physique du protocole I²C ..... 32
2.6.3. Stratégie de rafraîchissement graphique (Screen Buffering) ..... 33
2.7. Communication série avec un ordinateur (interface UART) ..... 33
2.7.1. Paramètres physiques de la liaison ..... 33
2.7.2. Protocole de trame textuelle (VAL, FEAT, PRED) ..... 33
2.7.3. Exploitation sur ordinateur (script Python) ..... 35
2.8. Constitution de la base de données et protocole expérimental ..... 35
**Conclusion du chapitre 2** .....35

**Chapitre 3 – Traitement numérique et classification par IA embarquée** ..... 37
3.1. Introduction ..... 37
3.2. Prétraitement numérique du signal PCG ..... 37
3.2.1. Centrage du signal et étalonnage initial (noiseFloor) ..... 38
3.2.2. Contrôle Automatique de Gain (AGC) adaptatif et lissé ..... 38
3.2.3. Segmentation en fenêtres glissantes ..... 38
3.2.4. Filtrage médian pour la réduction des pics impulsionnels ..... 39
3.3. Extraction des caractéristiques (features) ..... 39
3.3.1. Taux de passage par zéro (ZCR) ..... 39
3.3.2. Kurtosis (aplatissement) .....40
3.3.3. Skewness (asymétrie) .....40
3.3.4. Récapitulatif des caractéristiques .....40
3.4. Classification par Random Forest .....41
3.4.1. Principe du Random Forest .....41

3.4.2. Entraînement et validation croisée sous Python .....42
3.4.3. Génération du fichier HeartClassifier_Balanced.h (TinyML) ..... 43
3.5. Implémentation sur ESP32 .....44
3.5.1. Acquisition ADC et exécution du filtrage numérique .....44
3.5.2. Calcul des caractéristiques en temps réel .....45
3.5.3. Inférence du modèle Random Forest .....46
3.5.4. Optimisation mémoire et temps de calcul .....46
3.6. Interface homme-machine externe – Logiciel Python ..... 47
3.6.1. Protocole de communication et réception des trames ..... 48
3.6.2. Visualisation temps réel ..... 49
3.6.3. Enregistrement et sauvegarde (DAT, HDR, WAV) ..... 49
3.6.4. Enregistrement audio par le microphone de l'ordinateur ..... 50
3.6.5. Génération de rapport PDF ..... 50
3.6.6. Organisation du code et dépendances ..... 50
**Conclusion du chapitre 3** .....51

**Chapitre 4 – Validation et évaluation du système** ..... 52
4.1. Introduction .....52
4.2. Validation de la chaîne d'acquisition .....52
4.2.1. Visualisation des signaux PCG acquis (avec et sans pavillon) .....52
4.2.2. Vérification de la qualité du signal après filtrage .....53
4.2.3. Influence du positionnement du capteur .....54
4.3. Évaluation du modèle de classification Random Forest (validation croisée) ..... 54
4.3.1. Performances obtenues lors de la validation croisée .....55
4.3.2. Matrice de confusion (validation croisée) .....55
4.3.3. Calcul des indicateurs de performance .....56
4.4. Validation sur les enregistrements locaux (prototype) ..... 56
4.4.1. Description du protocole expérimental .....56
4.4.2. Résultats individuels .....56
4.4.3. Matrice de confusion et indicateurs sur le prototype .....57
4.5. Évaluation des performances embarquées de l'ESP32 .....58
4.5.1. Temps d'exécution de l'inférence .....58
4.5.2. Occupation mémoire du programme .....58
4.5.3. Consommation énergétique approximative .....58
4.6. Fonctionnement de l'interface Python .....59
4.6.1. Affichage en temps réel des signaux .....59
4.6.2. Sauvegarde des données aux formats .dat et .hdr .....59
4.6.3. Exemple d'enregistrement complet .....59
4.7. Discussion des résultats .....59
4.7.1. Points forts du système proposé .....59
4.7.2. Limites rencontrées .....60
4.7.3. Perspectives d'amélioration .....60
**Conclusion du chapitre 4** .....60

**Conclusion du general** .....62

**Bibliographie** ..... 64

# Listes des figures

Figure 1- 1: Schéma du coeur( Ventricule et oreillette gauches ) ..... 2
Figure 1- 2: Schéma du cœur(Atrium et ventricule droits du cœur (légendé)) ..... 3
Figure 1- 3: Schéma Valves cardiaques ..... 4
Figure 1- 4: Schéma de la circulation du cœur ..... 5
Figure 1- 5: Le signal phono cardiogramme pendant un cycle cardiaque. ..... 7
Figure 1- 6: Relation entre les bruits cardiaques et la contraction cardiaque. [b.3] ..... 9
Figure 1- 7: Stéthoscope Laënnec : 1) stéthoscope assemblé, 2) et 3) deux segments du stéthoscope en coupe longitudinale, 4) pièce pectorale amovible, 5) pointe dévissée, 6) coupe transversale (Roguin, 2006) 6) coupe transversale (Roguin, 2006) ..... 11
Figure 1- 8: Composition d'un stéthoscope classique (Nelson, 2015). ..... 12
Figure 1- 9: Composition d'un stéthoscope électronique (Nelson, 2015). ..... 13

Figure 2- 1: Schéma bloc de l'architecture matérielle ..... 17
Figure 2- 2: Module de charge type c . ..... 19
Figure 2- 3: détail synoptique et physique pour des modules d'alimentation ..... 21
Figure 2- 4: Un microphone piézoélectrique. ..... 22
Figure 2- 5: Le microphone dynamique ..... 23
Figure 2- 6: photo du capteur ECM ..... 24
Figure 2- 7: Module microphone détecteur de son KY-038 ..... 25
Figure 2- 8: Schéma des valves cardiaques et de leurs foyers d'auscultation ..... 28
Figure 2- 9: photos d'écran Oled intégrer dans Esp32 ..... 31

Figure 3- 1: Schéma block de principe du Random Forest (forêt aléatoire) ..... 42
Figure 3- 2 :Organigramme de principe d'interface python……..48

Figure 3- 3 : Interface graphique de l'application CardioTrack AI……..49.

# Listes des tableaux

Tableau 1.1 - Comparaison des stéthoscopes conventionnel et électronique……..14

Le Tableau 2.2- synthétise les critères d'évaluation des trois technologies de captation en vue de leur intégration dans notre stéthoscope numérique portatif…….25

Tableau 2.3 - Connexion entre le module KY-038 et l'ESP32……....26

Tableau 2.5 - Spécifications de l'interface physique de l'écran OLED intégré……....32

Le Tableau 2.6 - récapitule les caractéristiques structurelles et les bandes passantes théoriques de ces trames……..34

Le tableau 3.1 synthétise les trois descripteurs, ordonnés selon l'agencement exact requis par le vecteur d'entrée du micrologiciel, ainsi que leurs valeurs typiques pour un signal cardiaque normal……41

Le tableau 3.2 résume les spécifications finales du modèle de classification embarqué……...44

Le tableau 3.3 récapitule l'estimation des ressources consommées par l'application sur l'ESP32……..48

Tableau 4.1 –

Tableau 4.1 – Matrice de confusion de la validation croisée (10 plis) ..... 54

Tableau 4.3 – Résultats individuels des enregistrements locaux (5 sujets) .....55

Tableau 4.4 – Matrice de confusion obtenue sur le prototype .....57

# Liste des abréviations

- **ADC** : Analog-to-Digital Converter (Convertisseur analogique-numérique)
- **AGC** : Automatic Gain Control (Contrôle automatique de gain)
- **BPM**: Beats Per Minute (Battements par minute)
- **CAN**: Convertisseur Analogique-Numérique
- **CPU** : Central Processing Unit (Unité centrale de traitement)
- **CSV** : Comma-Separated Values (Valeurs séparées par des virgules)
- **DC**: Direct Current (Courant continu)
- **ECM** : Electret Condenser Microphone (Microphone à électret)
- **ESP32**: Microcontrôleur d'Espressif Systems
- **FFT** : Fast Fourier Transform (Transformée de Fourier rapide)
- **FIR** : Finite Impulse Response (Réponse impulsionnelle finie)
- **GPIO**: General Purpose Input/Output (Entrée/Sortie polyvalente)
- **I²C**: Inter-Integrated Circuit (Bus de communication série)
- **IA**: Intelligence Artificielle
- **IDE** : Integrated Development Environment (Environnement de développement intégré)
- **JFET** : Junction Field-Effect Transistor (Transistor à effet de champ à jonction)
- **LCD** : Liquid Crystal Display (Afficheur à cristaux liquides)
- **LDO** : Low Drop-Out (Régulateur linéaire à faible chute de tension)
- **LED**: Light-Emitting Diode (Diode électroluminescente)
- **LSB**: Least Significant Bit (Bit de poids faible)
- **MOSFET**: Metal-Oxyde-Semiconductor Field-Effect Transistor
- **OLED**: Organic Light-Emitting Diode (Diode électroluminescente organique)
- **PC** : Personal Computer (Ordinateur personnel)
- **PCB**: Printed Circuit Board (Circuit imprimé)
- **PCG**: Phono cardiogramme
- **PCM** : Protection Circuit Module (Module de protection du circuit)
- **PDF**: Portable Document Format
- **PSRR** : Power Supply Rejection Ratio (Taux de réjection de l'alimentation)
- **PVC** : Polyvinyl Chloride (Polychlorure de vinyle)
- **RAM**: Random Access Memory (Mémoire vive)
- **SAR** : Successive Approximation Register (Registre à approximations successives)
- **SCL**: Serial Clock Line (Ligne d'horloge série)
- **SDA** : Serial Data Line (Ligne de données série)
- **SNR**: Signal-to-Noise Ratio (Rapport signal sur bruit)
- **SRAM**: Static Random Access Memory (Mémoire vive statique)
- **TinyML** : Tiny Machine Learning (Apprentissage automatique sur microcontrôleurs)
- **TTL**: Transistor-Transistor Logic
- **UART** : Universal Asynchronous Receiver-Transmitter (Émetteur-récepteur asynchrone universel)
- **USB**: Universal Serial Bus
- **WAV**: Waveform Audio File Format (Format audio)
- **ZCR** : Zero-Crossing Rate (Taux de passage par zéro)
-

# Introduction Générale

L'auscultation cardiaque est l'un des gestes médicaux les plus anciens et les plus pratiqués en cardiologie, permettant d'apprécier les bruits normaux du cœur ainsi que les souffles évocateurs de pathologies valvulaires. Malgré les progrès considérables réalisés en imagerie cardiaque et en échocardiographie, l'examen clinique au stéthoscope reste un outil de première intention, irremplaçable par sa rapidité, son faible coût et sa disponibilité. Cependant, son interprétation demeure une tâche complexe, fortement tributaire de l'expérience du clinicien et sensible aux bruits environnementaux. Cette difficulté est particulièrement marquée chez l'enfant ou chez les patients en surpoids, où les signaux phono cardiographiques (PCG) peuvent être atténués ou masqués par des artefacts. Cette incertitude diagnostique peut avoir des conséquences cliniques importantes, allant de la prescription d'examens complémentaires inutiles (et coûteux) à des retards de prise en charge de pathologies potentiellement graves.

Face à cette problématique diagnostique cruciale et aux enjeux sanitaires qui en découlent, une question centrale se pose : **Comment l'intégration de l'Intelligence Artificielle (IA) embarquée, et plus spécifiquement des approches de Machine Learning (ML) adaptées aux microcontrôleurs (TinyML), peut-elle améliorer significativement la précision et la rapidité du diagnostic cardiaque à partir du signal PCG, tout en garantissant la portabilité et l'autonomie du dispositif ?**

Dans ce contexte, l'émergence des microcontrôleurs low-power (tels que l'ESP32) et des techniques d'apprentissage automatique embarqué offre des perspectives prometteuses pour transformer les pratiques auscultatoires. Les modèles de Machine Learning, en particulier les forêts aléatoires (Random Forest), se distinguent par leur capacité à apprendre à partir de caractéristiques temporelles extraites du signal PCG (taux de passage par zéro, aplatissement, asymétrie) et à fournir une classification binaire (normal / pathologique) en temps réel, sans nécessiter de connexion à un serveur externe. Leur intégration directement au sein du stéthoscope électronique vise à améliorer la précision diagnostique, à assister le clinicien dans sa décision, et, in fine, à réduire le nombre d'examens complémentaires inutiles tout en accélérant la prise en charge des patients.

Dans cette optique d'innovation et d'interdisciplinarité, l'objectif principal de ce projet de fin d'études est de concevoir, réaliser et valider un **stéthoscope électronique intelligent** embarquant un modèle de classification Random Forest exécuté sur un microcontrôleur ESP32. Plus spécifiquement, ce travail vise à :

- Comprendre et synthétiser en profondeur les aspects anatomiques, physiologiques et acoustiques des bruits cardiaques, ainsi que les défis diagnostiques liés aux souffles pathologiques ;
- Concevoir une chaîne d'acquisition matérielle optimisée (capteur à électret KY-038, cloche acoustique, conditionnement, numérisation 12 bits à 200 Hz) alimentée par une batterie lithium-ion, garantissant autonomie et portabilité ;
- Développer une chaîne de traitement numérique embarquée (centrage, filtre médian, contrôle automatique de gain, extraction des trois caractéristiques temporelles) et intégrer un modèle Random Forest pré-entraîné converti en C++ natif via l'outil EloquentTinyML ;

- Démontrer l'efficacité et la robustesse du système en le validant sur une base de données publique de référence (Physio Net Challenge 2016) ainsi que sur des enregistrements locaux réalisés avec le prototype sur des sujets sains et pathologiques.

Pour atteindre ces objectifs, ce projet se structure de manière logique en quatre chapitres complémentaires :

- **Chapitre 1** : rappel des bases anatomiques et physiologiques du cœur, description des bruits cardiaques normaux (B1, B2) et des souffles, ainsi qu'un état de l'art des stéthoscopes acoustiques et électroniques. Ce chapitre met en lumière la complexité du diagnostic auscultatoire et justifie la nécessité d'une aide algorithmique.
- **Chapitre 2** : présentation détaillée de l'architecture matérielle du prototype : alimentation (batterie 18650 + convertisseur élévateur), transduction acoustique (microphone KY-038 couplé à une cloche médicale), numérisation par l'ESP32 (ADC 12 bits, 200 Hz), interface utilisateur locale (écran OLED) et communication série vers un PC. Les choix technologiques et les critères de validation sont explicités.
- **Chapitre 3** : description complète de la chaîne de traitement numérique embarquée sur l'ESP32 : prétraitement (centrage, filtrage médian, AGC), extraction des caractéristiques (ZCR, kurtosis, skewness), et classification par un modèle Random Forest de 25 arbres. Ce chapitre détaille également l'entraînement du modèle sous Python, sa conversion en code C++ natif (TinyML), ainsi que le développement de l'interface graphique Python (CardioTrack AI) pour la visualisation, l'enregistrement et l'export des données.
- **Chapitre 4** : validation expérimentale complète du système. Il évalue d'abord la chaîne d'acquisition (influence du pavillon, qualité du signal après filtrage, rapport signal sur bruit). Il présente ensuite les performances du modèle Random Forest en validation croisée (94,2 % d'exactitude) puis sur des enregistrements locaux réalisés avec le prototype (5 sujets, exactitude de 80 %, sensibilité de 100 %). Enfin, il analyse les performances embarquées (temps d'inférence, occupation mémoire, consommation énergétique) et discute les limites ainsi que les perspectives d'amélioration.

Ce travail démontre la faisabilité d'un stéthoscope électronique portable, économique, doté d'une intelligence artificielle embarquée capable d'assister le clinicien dans le diagnostic précoce des anomalies cardiaques. Les résultats encourageants ouvrent la voie à de nombreuses améliorations (classification multi-classes, connectivité sans fil, calcul du BPM en temps réel) et positionnent ce prototype comme une base solide pour des applications de télémédecine ou d'enseignement en cardiologie.

# Chapitre 1

# Anatomie et fonctionnement du cœur

## 1.1-Définition du cœur

Le cœur est un organe musculaire qui pompe le sang à travers le corps en le faisant circuler à travers le système circulatoire/vasculaire. Il se trouve au milieu du médiastin, enveloppé d'un sac séreux constitué de deux couches, appelé le péricarde. Le cœur est en forme de pyramide à base carrée orientée comme si elle était tombée sur l'une de ses faces, sa base faisant ainsi face à la paroi thoracique postérieure, et son sommet pointant vers la paroi thoracique antérieure.

Les grands vaisseaux provenant du cœur envoient des branches vers la tête et le cou, le thorax et l'abdomen, ainsi que les membres supérieurs et inférieurs. [1]

Le cœur possède cinq faces : la face basale (ou postérieure), la face diaphragmatique (ou inférieure), la face Stern costale (ou antérieure) et les surfaces pulmonaires gauche et droite. Il a également plusieurs bords : droit, gauche, supérieur et inférieur :

- Le bord droit est une petite zone de l'oreillette droite qui s'étend de la veine cave supérieure à la veine cave inférieure.
- Le bord gauche est formé par le ventricule gauche et l'oreillette gauche.
- Le bord supérieur, sur le côté antérieur, est constitué des deux oreillettes et de leur auricule.
- Le bord inférieur est marqué par le ventricule droit.

À l'intérieur, le cœur est divisé en quatre chambres : deux oreillettes (droite et gauche) et deux ventricules (droit et gauche). [1]

### 1.1.1-Les oreillettes :

Représentent les cavités supérieures du cœur, caractérisées par leurs parois minces et leur développement musculaire limité. L'oreillette droite reçoit du sang désoxygéné provenant de la circulation systémique (via la veine cave), tandis que l'oreillette gauche reçoit du sang oxygéné provenant de la circulation pulmonaire (par les veines pulmonaires).

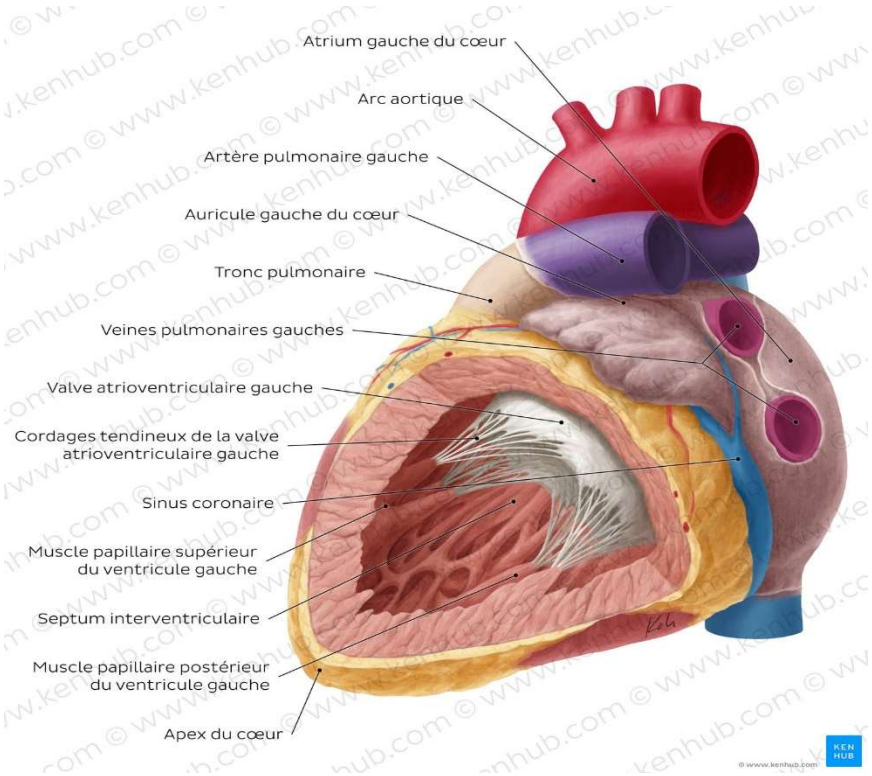



**Figure 1.1:** Schéma du cœur : Ventricule et oreillette gauches

**1.1.2-Les ventricules :** Sont les cavités inférieures, qui se distinguent par leurs parois plus épaisses et leur plus grande musculature par rapport aux oreillettes. Le ventricule droit est chargé de propulser le sang vers les poumons via l'artère pulmonaire, tandis que le ventricule gauche est chargé de distribuer le sang oxygéné dans tout le corps par l'aorte.

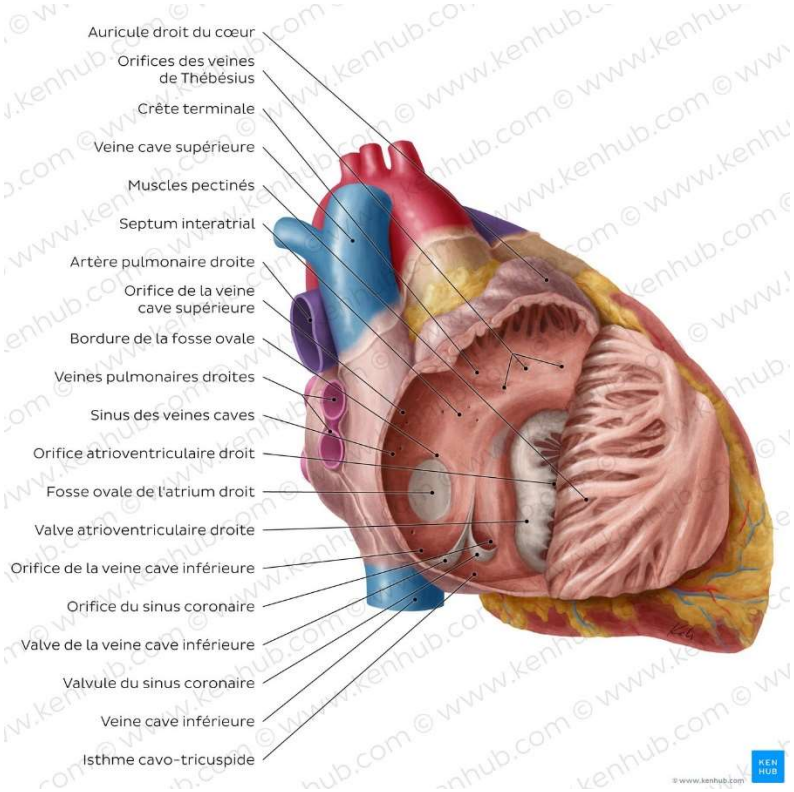



**Figure 1.2:** Schéma du cœur : Atrium et ventricule droits du cœur (légendé)

Les côtés gauche et droit du cœur sont séparés par les septums inter auriculaire et interventriculaire qui existent en continuité l'un avec l'autre. De plus, les oreillettes sont séparées des ventricules par un septum atrioventriculaire. Le sang circule des oreillettes aux ventricules à travers les orifices atrioventriculaires (droit et gauche) -des ouvertures dans le septum atrioventriculaire. Ces ouvertures sont fermées et ouvertes périodiquement par les valves du cœur selon la phase du cycle cardiaque.[1]

## 1.1.3-Les valves cardiaques :

Séparent les oreillettes des ventricules, et les ventricules des gros vaisseaux. Ces valves comprennent deux ou trois feuillets (ou "cuspides") autour des orifices atrioventriculaires et racines des gros vaisseaux.

Les valves sont poussées ouvertes pour permettre la circulation du sang dans une direction, puis refermées pour sceller les orifices et empêcher le reflux sanguin. Le prolapsus des valves vers l'arrière est bloqué par les cordages tendineux -ou cordages du cœur

Il existe deux groupes de valves : atrioventriculaire et semilunaire. Les valves atrioventriculaires empêchent le reflux sanguin des ventricules aux oreillettes :

**La valve atrioventriculaire droite/valve tricuspide** se trouve entre l'oreillette droite et le ventricule droit. Elle est composée de trois feuillets/cuspides.

**La valve atrioventriculaire gauche**/valve bicuspide est aussi appelée **la valve mitrale**, puisqu'elle ne possède que deux cuspides et ressemble ainsi à une mitre. Elle est située entre l'oreillette gauche et le ventricule gauche et est formée de deux feuillets/cuspides.

**La valve semilunaire pulmonaire** se trouve entre le ventricule droit et l'orifice du tronc pulmonaire. Elle possède trois feuillets/cuspides.

**La valve semilunaire aortique** se situe entre le ventricule gauche et l'orifice de l'aorte. Elle est composée de trois feuillets/cuspides semilunaires.

En pratique clinique, les valves du cœur peuvent être auscultées, d'habitude en utilisant un stéthoscope. [1]

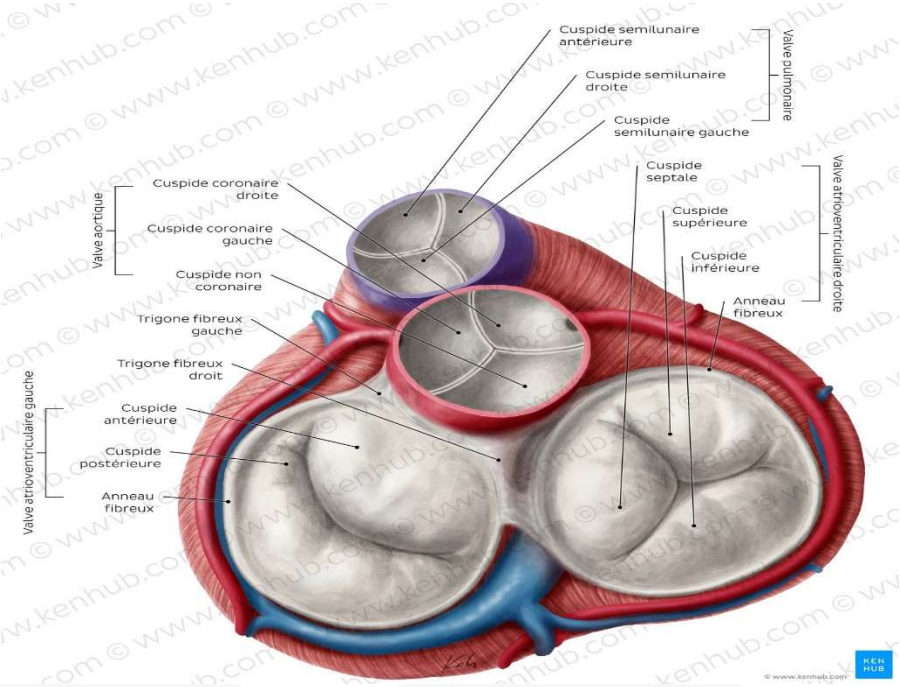



**Figure 1.3:** Schéma Valves cardiaques

# 1.2.Concept de la circulation systémique (grand circulation) et pulmonaire (petite circulation), et rôle des valves dans l'assurance d'un flux sanguin à sens unique

## 1.2.1-La Circulation Pulmonaire (Petite Circulation) :

Ce circuit assure l'oxygénation du sang. Le sang désoxygéné, arrivant de la partie supérieure et inférieure du corps **(1)**, pénètre dans l'oreillette droite **(2)**. Celle-ci se contracte pour l'éjecter dans le ventricule droit **(3)**. Après la fermeture de la valve pour éviter tout reflux, le ventricule droit se contracte à son tour et propulse le sang dans le tronc pulmonaire **(4)**. Le sang est alors acheminé vers les poumons **(5)** où il se débarrasse du dioxyde de carbone et s'enrichit en oxygène au niveau des capillaires pulmonaires. [2]

## 1.2.2-La Circulation Systémique (Grande Circulation) :

Ce circuit distribue le sang oxygéné à l'ensemble de l'organisme. Le sang riche en oxygène arrivant des poumons **(6)** est recueilli par l'oreillette gauche **(7)**, qui l'expulse ensuite dans le ventricule gauche **(8)**. Lors de la systole ventriculaire, le ventricule gauche, plus musclé, propulse le sang avec force dans l'aorte **(9)**. De là, le flux sanguin est distribué dans tout l'organisme **(10)** via le réseau artériel. Les contractions se produisent en parfaite synchronie : les deux oreillettes **(2, 7)** se vident simultanément, suivies par l'éjection conjointe des deux ventricules **(3, 8)**.[2]

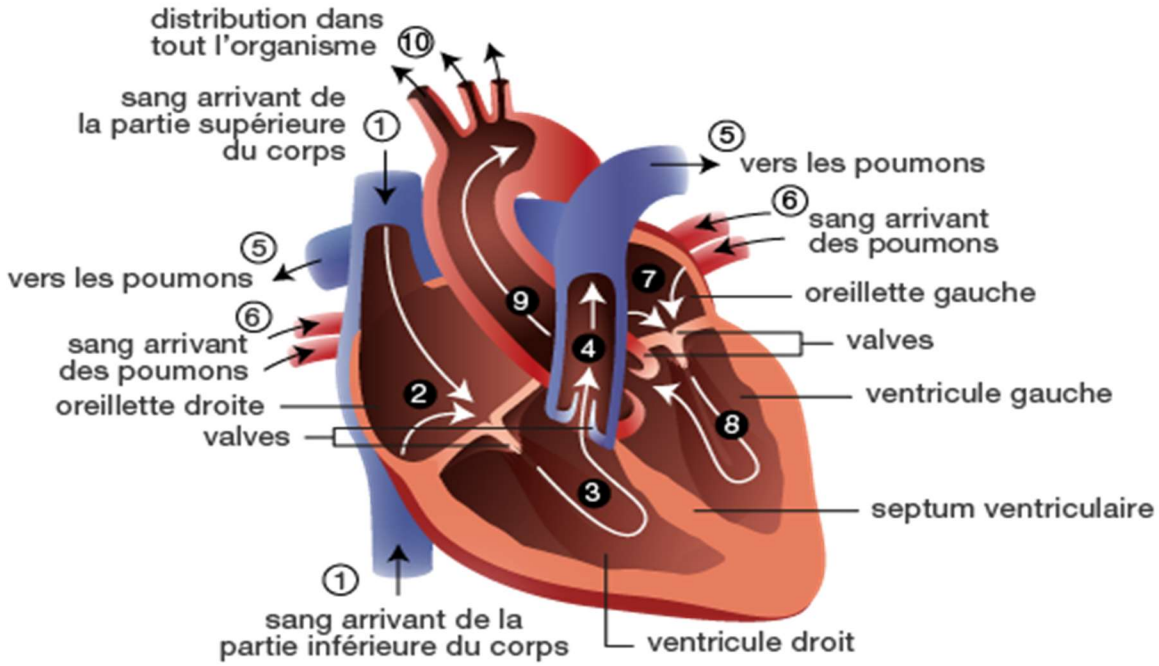



**Figure 1.4 :** Schéma de la circulation du cœur

### 1.2.3-Rôle des valves dans le flux unidirectionnel :

Les quatre valves cardiaques agissent comme des clapets à sens unique : elles s'ouvrent quand la pression pousse le sang dans le bon sens, puis se ferment pour empêcher tout reflux.
• **Les valves atrioventriculaires** (tricuspide et mitrale) empêchent le retour du sang des ventricules vers les oreillettes.
• **Les valves semilunaires** (pulmonaire et aortique) empêchent le sang de revenir des artères pulmonaire et aorte vers les ventricules.

## 1.3.-Cycle cardiaque (The Cardiac Cycle):

Le **cycle cardiaque** regroupe l'ensemble des événements électriques et mécaniques survenant entre deux battements, durant environ **0,8 seconde**. Ce processus repose sur une succession rythmée de variations de pression et de volume, déclenchées par les changements électrochimiques du myocarde (dépolarisation et repolarisation). Ces fluctuations de pression dictent l'ouverture et la fermeture passive des valves (mitrale, tricuspide, aortique et pulmonaire), imposant un flux unidirectionnel à travers les cavités cardiaques et vers les circuits pulmonaire et systémique.

Le cycle s'articule autour de deux grandes phases interdépendantes :

1. **La Diastole Ventriculaire (Relaxation et Remplissage) :** Après l'éjection, les ventricules **[3, 9-5]** entrent en **relaxation isovolumétrique** (pression en baisse, volume constant) jusqu'à l'ouverture des valves atrioventriculaires. Le remplissage s'opère alors en trois temps : un **remplissage rapide** précoce, suivi d'une phase de remplissage lent appelée **diastasis**, et s'achève par la **systole auriculaire**. Cette contraction des oreillettes **[4-5] [8-7]** donne le "coup de pouce" final pour optimiser le volume télé diastolique avant la contraction ventriculaire.
2. **La Systole Ventriculaire (Contraction et Éjection) :** Le muscle ventriculaire se contracte fortement, provoquant d'abord une phase de **contraction iso volumétrique** où la pression grimpe brusquement tandis que toutes les valves sont fermées (marquant le premier bruit cardiaque). Dès que la pression intraventriculaire excède la pression artérielle, les valves sigmoïdes s'ouvrent, permettant l'**éjection ventriculaire** du sang vers le tronc pulmonaire **[5-8]** et l'aorte **[8-7]**. En fin de phase, la fermeture de ces valves (deuxième bruit) marque le début d'un nouveau cycle.

   Ce mécanisme assure un débit sanguin régulier, soutenu par le **mécanisme de Frank-Sterling**, qui permet au cœur d'ajuster sa force de contraction au volume de sang reçu, garantissant ainsi l'équilibre constant de la distribution dans tout l'organisme [**9-8]**.

## 1.4. Les bruits cardiaques :

Les bruits cardiaques sont des vibrations mécaniques et acoustiques provoquées par la pression systolique et diastolique du myocarde, l'ouverture et la fermeture des valvules cardiaques ainsi que l'impact de l'écoulement sanguin dans la paroi ventriculaire et les différents vaisseaux sanguins qui entoure le cœur, et qui produit un signal appelé phono cardiogramme(PCG) qui peut être visualiser à l'aide d'un stéthoscope électronique. **[11]**

**1.4.1-Le phono cardiogramme (PCG)** : constitue une représentation graphique des phénomènes acoustiques cardiaques. Par la conversion des vibrations sonores en signaux électriques, cet examen permet de matérialiser sur un tracé les bruits du cœur. Il offre ainsi une analyse objective et détaillée venant compléter ou affiner les observations recueillies par le cardiologue lors de l'auscultation au stéthoscope.**[12]**

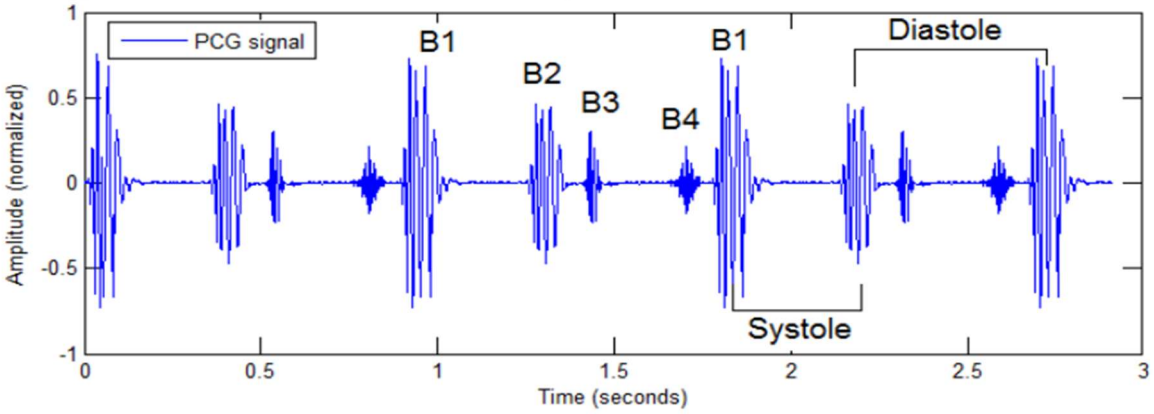



**Figure 1.5:** Le signal phono cardiogramme pendant un cycle cardiaque.

## 1.4.2-Le premier bruit cardiaque (B1) :

- B1 correspond au début de la systole ventriculaire (contraction du myocarde), marquant la fin de la diastole et le début de l'éjection ventriculaire, juste avant le pouls carotidien.

- Son sourd et grave (« poum »), plus long et plus intense que B2.

- Contenu fréquentiel dominé par les basses fréquences par rapport à B2.

- Généré en grande partie par la fermeture des valvules auriculo-ventriculaires (mitrale et tricuspide).

- Se décompose en 4 groupes de vibrations :

- **Groupe 1** : oscillations de faible amplitude et basse fréquence, non audibles → début de la contraction ventriculaire.

- **Groupe 2 :** oscillations de grande amplitude, audibles → fermeture mitrale (M1) puis tricuspide (T1). Chez le sujet sain, l'intervalle M1-T1 est < 30 ms, perçu comme un bruit unique.

- **Groupe 3** : oscillations de grande amplitude, audibles → ouverture des valves sigmoïdes (aortique et pulmonaire).

- **Groupe 4** : oscillations de faible amplitude et basse fréquence, non audibles → flux sanguin turbulent dans l'aorte et l'artère pulmonaire brusquement dilatées au début de l'éjection ventriculaire.

- Entendu le plus fortement à la pointe du cœur.

### 1.4.3-Le deuxième bruit cardiaque (B2) :

- B2 correspond à la fermeture des valves sigmoïdes (aortique et pulmonaire) à la fin de la systole ventriculaire, marquant le début de la diastole ventriculaire.

- Sa durée ne dépasse généralement pas 100 ms.

- Caractère acoustique : son sec, claquant, plus aigu que B1, de tonalité élevée (contenu fréquentiel dominé par les fréquences aiguës), pouvant atteindre jusqu'à 200 Hz. Il est souvent décrit comme « Ta ».

- Audibilité maximale : à la base du cœur.

- Se compose essentiellement de deux groupes de vibrations :

- **Groupe 1 :** vibrations de faible amplitude, non audibles, dues aux tourbillons sanguins précédant la fermeture des sigmoïdes.

- **Groupe 2** : vibrations audibles de hautes fréquences, correspondant aux fermetures des sigmoïdes aortique (composante A2) et pulmonaire (composante P2). Chacune des composantes A2 et P2 dure moins de 50 ms.

- Séparation des composantes A2 et P2 : généralement 3 à 4 ms chez le sujet sain ; cet intervalle augmente lors de l'inspiration, ce qui peut entraîner un léger dédoublement physiologique de B2.

- **Bruits supplémentaires (S3 et S4) ** : ce sont des bruits diastoliques qui s'ajoutent au schéma classique S1–S2. Ils peuvent être physiologiques ou pathologiques selon l'âge, l'état cardiaque et le contexte clinique.

### 1.4.4-Le troisième bruit cardiaque (S3 ou B3) :

- Survient en début de diastole, juste après le deuxième bruit (B2/S2), à la fin de la phase de remplissage ventriculaire rapide.

- Correspond à la décélération brusque du sang entrant dans un ventricule (souvent dilaté ou peu complaint).

- Caractère acoustique : bruit de basse fréquence.

- Fréquemment enregistré chez le sujet jeune (enfant, adolescent) et parfois chez la femme enceinte, où il peut être un galop physiologique normal.

- Chez l'adulte de plus de 40 ans, sa présence témoigne le plus souvent d'un état pathologique, généralement un signe d'insuffisance cardiaque, de cardiomyopathie dilatée ou de surcharge volumique.

### 1.4.5-Le quatrième bruit cardiaque (S4 ou B4) :

- Apparaît en fin de diastole, juste avant le premier bruit (B1/S1).

- Débute juste avant B1.

- Lié à la contraction auriculaire (oreillette) qui force le remplissage d'un ventricule rigide ou hypertrophique.

- Caractère acoustique : bruit habituellement de basse fréquence.

En complément des descriptions fonctionnelles et acoustiques des bruits cardiaques B1, B2, B3 et B4, les caractéristiques physiques suivantes (fréquence, amplitude, durée) précisent leurs propriétés spectrales et temporelles essentielles.

- Plage audible des bruits cardiaques : 40 à 500 Hz

- Énergie de B1 et B2 surtout en dessous de 100 Hz

- Bruits cardiaques = sons de faible intensité

- Amplitude dépend de la vitesse de fermeture des valves et de la pression sanguine

- B1 et B2 ont une amplitude plus forte que B3/B4

- Durée de B1 : 0,10 - 0,15 s

- Durée de B2 : 0,08 - 0,12 s

- Intervalle systolique (B1-B2) plus court que l'intervalle diastolique (B2-B1) **[13]**

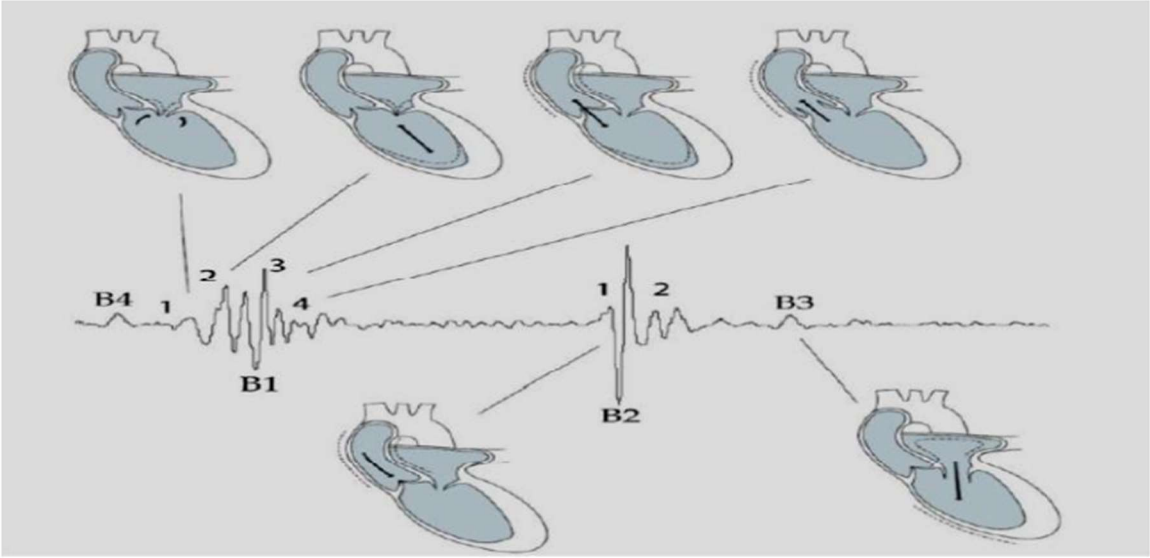



**Figure 1.6** : Relation entre les bruits cardiaques et la contraction cardiaque. [b.3]

# 1.5.Souffle cardiaque :

## 1.5.1-Définition du souffle cardiaque :

Le souffle cardiaque (ou « souffle au cœur ») est un phénomène sonore anormal, prolongé, de tonalité douce (type sifflement ou vibration), perçu à l'auscultation à l'aide d'un stéthoscope. Il se

superpose ou s'intercale aux bruits cardiaques normaux (S1 et S2, « toc-tac »), dont il se distingue fondamentalement par une durée plus longue.

Sur le plan physiopathologique, le souffle résulte d'un écoulement turbulent du sang depuis une zone de haute pression vers une zone de basse pression, à travers un orifice rétréci. Les causes en sont multiples :

- Un rétrécissement valvulaire (sténose)
- Une fuite valvulaire (insuffisance)
- Une communication anormale entre cavités cardiaques (ex : communication interventriculaire)
- Un débit sanguin élevé traversant des structures anatomiquement normales (souffle fonctionnel ou physiologique)

### 1.5.2-Type du souffle :

Sur le plan sémiologique, les souffles sont classés selon leur position dans le cycle cardiaque :

- **Souffles systoliques** : perçus entre B1 et B2. Ils peuvent être classés comme début, milieu ou fin de systole, ou holosystoliques (occupant toute la systole). Exemples : rétrécissement aortique, rétrécissement pulmonaire, insuffisance mitrale, insuffisance tricuspidienne, communication interventriculaire.
- **Souffles diastoliques** : perçus entre B2 et B1, classés comme début, milieu ou fin de diastole. Ils sont souvent pathologiques. Exemples : insuffisance aortique, rétrécissement mitral, rétrécissement tricuspidien.
- **Souffles continus** : entendus tout au long du cycle cardiaque (systole et diastole).

Sur le plan épidémiologique, les souffles cardiaques sont très fréquents, en particulier chez l'enfant : environ un enfant sur deux peut présenter un souffle à un moment donné de sa vie, surtout lors d'excitation ou de fièvre. La grande majorité des souffles sont **normaux** (ou innocents) ; seulement environ 1 % (un souffle sur cent) révèle une pathologie cardiaque sous-jacente.

L'évaluation d'un souffle cardiaque se fait selon une échelle de gradation (de I/VI à VI/VI) qui en apprécie l'intensité, ainsi que par son siège, son timbre, sa tonalité, son irradiation et sa variabilité selon les manœuvres respiratoires.

### 1.5.3-Causes pathologiques fréquentes (synthèse) :

- **Chez l'enfant et l'adolescent** : cardiopathies congénitales (shunts, jets de haute pression) souffles systoliques ou continus.
- **Chez l'adulte** : valvulopathies acquises (sténose aortique dégénérative, insuffisance mitrale, insuffisance aortique), endocardite infectieuse, rhumatisme articulaire aigu, hypertension artérielle chronique (souffles fonctionnels ou exagération de souffles organiques), cardiomyopathie hypertrophique obstructive (souffle systolique à l'éjection).

# 1.6. STRUCTURE ET FONCTIONNEMENT DES STÉTHOSCOPES

Inventé en 1816 par le médecin français René Laennec, le stéthoscope illustré à la Figure 1.1 comprend un tube en bois (Roguin, 2006). Cet instrument médical facilite l'examen auditif des sons cardiaques, pulmonaires, vasculaires ou gastro-intestinaux produits par les organes internes de l'anatomie d'un patient, sans lui infliger d'effets indésirables (Nelson, 2015). Il constitue un appareil essentiel tant pour les médecins généralistes que pour les spécialistes. À l'heure actuelle, deux catégories de stéthoscopes sont identifiées : les stéthoscopes classiques (ou acoustiques) et les stéthoscopes électroniques. [13]

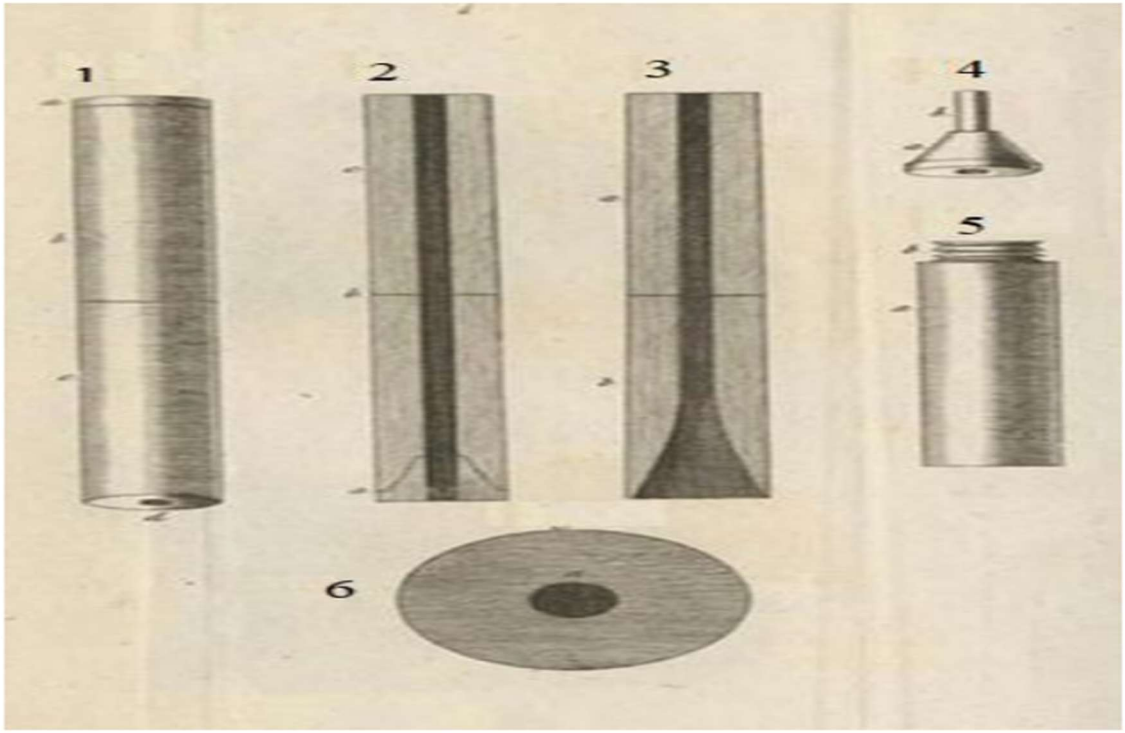



**Figure 1.7**: Stéthoscope Laënnec : 1) stéthoscope assemblé, 2) et 3) deux segments du stéthoscope en coupe longitudinale, 4) pièce pectorale amovible, 5) pointe dévissée, 6) coupe transversale (Roguin, 2006) 6) coupe transversale (Roguin, 2006).

### 1.6.1- Stéthoscope classique (conventionnel) :

Le stéthoscope conventionnel standard illustré à la Figure I.8 se compose essentiellement d'un pectoral équipé d'un diaphragme et d'une cloche, qui sont conçus pour capter les sons à haute et basse fréquence, respectivement. Ces sons sont transmis sous forme d'énergie sonore à travers un tube creux, qui mesure généralement entre 18 et 26 pouces de longueur (Nelson, 2015). Le tube creux se bifurque en un binaural qui se connecte à une paire d'oreillettes. Lorsqu'un médecin utilise le stéthoscope pour l'auscultation, celui-ci établit un circuit par lequel l'énergie sonore est transmise, avec une perte d'énergie minimale, du thorax du patient à l'oreille du médecin pour interprétation (Nelson, 2015).

Une multitude de variables peuvent influencer la qualité auditive d'un stéthoscope acoustique classique, notamment :

1. La qualité sonore globale est souvent sous-optimale, en particulier chez les personnes obèses ou les patients dont la paroi thoracique est épaisse ;

2. Toute perturbation du circuit peut entraîner une atténuation (par exemple, une fuite d'air dans le tube creux) ou entraîner occasionnellement l'arrêt complet de la transmission du son ;

3. Les données auditives acquises ne peuvent être analysées, traitées ou utilisées d'aucune autre manière. [13]

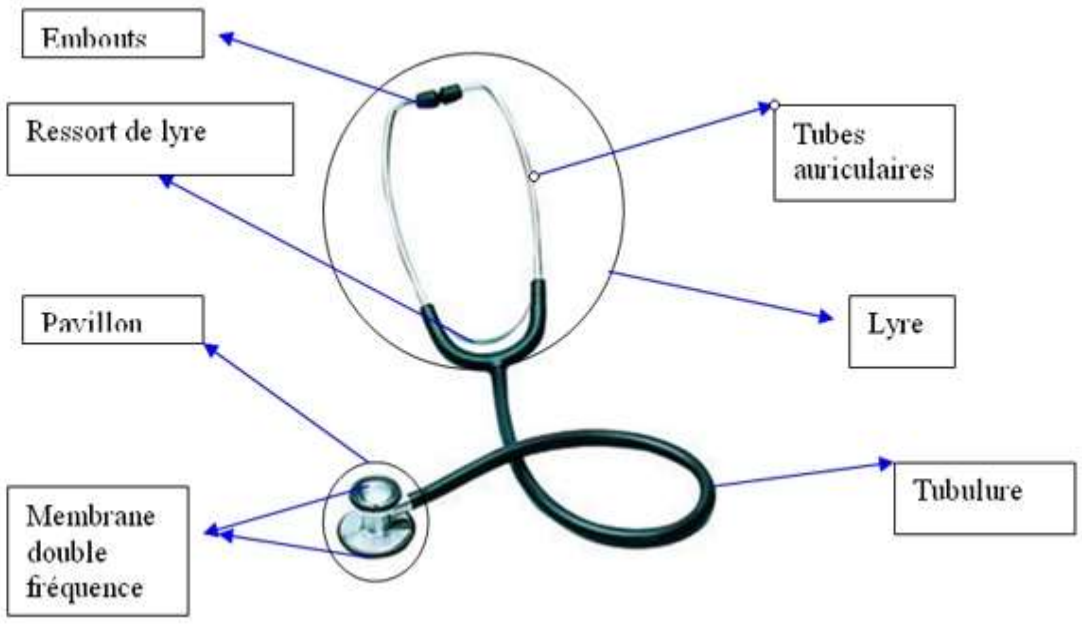



**Figure 1.8** : Composition d'un stéthoscope classique (Nelson, 2015).

Comme le montre la Figure I.8, le stéthoscope classique comprend essentiellement :

- **La tête** : l'élément le plus sensible du stéthoscope, qui est placé sur le derme du patient pour détecter les sons. La précision de la perception auditive dépend de la qualité de la corne ;
- **Membrane** : située sur la corne, elle a pour fonction de capter l'énergie vibratoire des sons émis par le corps du patient ;
- **Lyre :** structure métallique du stéthoscope, sur laquelle est fixé le tube. Il comprend deux tubes auriculaires, un ressort de tension et des embouts auriculaires ;
- **Embouts** : placés au niveau du pavillon auditif de l'utilisateur, des embouts de haute qualité et bien adaptés doivent être utilisés ; 8

- **Tubulure** : située entre la lyre et la cloche, généralement en PVC ;
- **Tube auriculaire** : composant métallique qui établit la connexion entre les embouts et le tube flexible. La tension peut être facilement ajustée à la position la plus confortable en écartant les tubes auriculaires ou en les croisant pour les fixer.
- **Base :** facilite la connexion du pavillon au conduit. Ce composant permet donc à certains modèles de choisir entre la réception auditive de la cloche ou de la membrane de la cloche en manipulant le klaxon.

### 1.6.2-Stéthoscope électronique :

Le stéthoscope électronique standard illustré à la Figure I.9 offre une perspective innovante en améliorant les caractéristiques spécifiques du stéthoscope acoustique traditionnel. Contrairement au stéthoscope classique, la variante électronique permet le stockage et l'analyse ultérieure des données (Nelson, 2015).

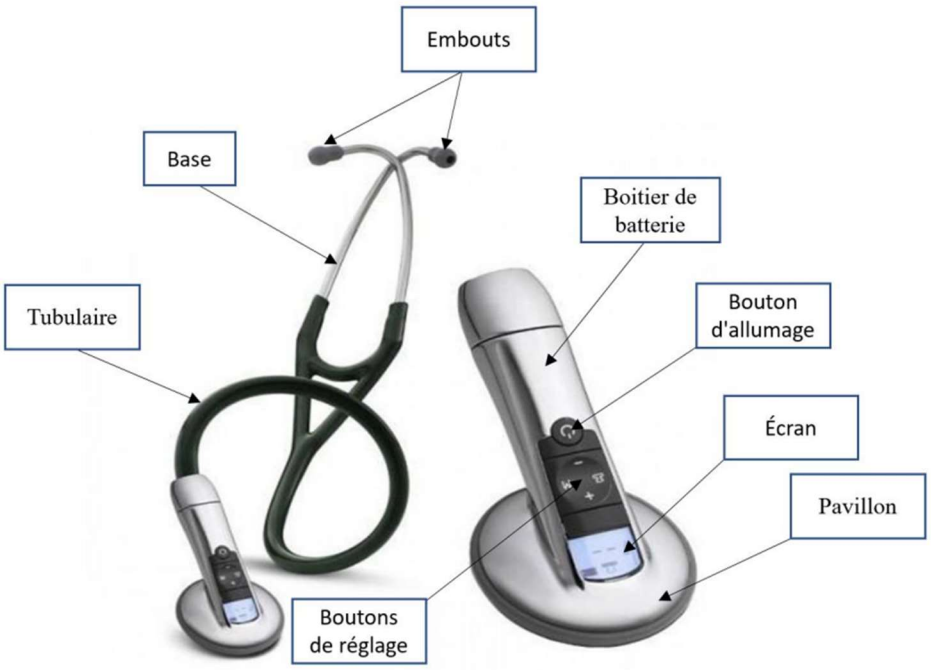



**Figure 1.9** : Composition d'un stéthoscope électronique (Nelson, 2015).

#### 1.6.2.1-Principes et Fonctionnement

Dans un stéthoscope électronique, la transmission acoustique directe est remplacée par un système électronique complexe. Le capteur, élément central du dispositif, est généralement un microphone à haute sensibilité ou un disque piézo-céramique capable de détecter les vibrations à la surface de la peau.

Pour optimiser la capture, certains modèles conservent une cavité d'air fermée avec un diaphragme pour fournir la pression différentielle requise par le capteur. La numérisation du son

permet ensuite une grande polyvalence grâce à différents modes de filtrage sélectif, offrant ainsi une qualité sonore supérieure et une amplification réglable par rapport au stéthoscope mécanique classique.

### 1.6.2.2- Chaîne de traitement du signal numérique

Le passage du son du thorax vers l'utilisateur final suit trois étapes critiques :

- **Acquisition de données :** Les bruits cardiaques sont convertis en signaux électriques par le transducteur, puis amplifiés et filtrés par un circuit passe-bande analogique pour réduire les bruits ambiants avant la conversion analogique-numérique (ADC).
- **Prétraitement :** Une phase de filtrage numérique permet d'isoler la plage de fréquences d'intérêt (typiquement entre 20 Hz et 1000 Hz) et d'éliminer les artefacts sonores.
- **Traitement et Visualisation :** Les logiciels de traitement permettent de représenter ces données sous forme de phono cardiogramme (PCG) ou de graphique spectral. Ces fichiers peuvent être convertis en formats audio standards (WAV) pour un stockage ou une analyse ultérieure.

## 1.6.3- Comparaison entre les systèmes conventionnels et électroniques

Le tableau suivant résume les différences majeures entre les deux technologies :

| Caractéristique | Stéthoscope Conventionnel | Stéthoscope Électronique |
|---|---|---|
| **Conception** | Traditionnelle et mécanique | Électronique avec capteurs intégrés |
| **Qualité sonore** | Faible amplification | Hautefidélité et volume réglable |
| **Logiciel** | Aucun | Disponible pour le post-traitement et l'analyse |
| **Durabilité** | Très élevée (pas de composants électroniques) | Limitée (dépendance aux batteries et écrans) |
| **Coût** | Abordable | Élevé |

**Tableau 1.1 : Comparaison des stéthoscopes conventionnel et électronique**

## 1.7. Problématique de l'interprétation humaine et nécessité de l'IA

Malgré les avancées du stéthoscope électronique, son efficacité reste tributaire de l'expertise de l'utilisateur. L'écoute des sons cardiaques est souvent perturbée par divers bruits environnementaux (bavardages, pleurs d'enfants, bruits de ventilation).

De plus, l'interprétation clinique des signaux PCG est une tâche intrinsèquement subjective. La similitude acoustique entre certains murmures physiologiques et pathologiques rend le diagnostic humain particulièrement vulnérable aux erreurs, surtout dans des environnements cliniques stressants ou pour des praticiens non spécialisés. Cette subtilité sonore, souvent imperceptible à l'oreille humaine, justifie l'intégration d'algorithmes de Random Forest.

Contrairement à l'humain, ces modèles sont capables d'extraire des caractéristiques temporelles et fréquentielles invisibles à l'œil (sur un PCG) et inaudibles à l'oreille, transformant le stéthoscope en un véritable outil d'aide au diagnostic précoce et objectif.

## Conclusion :

Ce premier chapitre nous a permis de cerner avec rigueur les fondements anatomiques et physiologiques du système cardiovasculaire, ainsi que la nature physique et acoustique des bruits cardiaques. Comprendre la dynamique et les limites fréquentielles de ces signaux biomédicaux est une étape cruciale pour l'ingénieur. En effet, ce sont ces ondes sonores de basses fréquences qui doivent être captées, conditionnées et numérisées de manière optimale. C'est précisément cette transition critique de l'acoustique médicale vers l'électronique embarquée et le matériel informatique qui fera l'objet de notre deuxième chapitre dédié à la conception matérielle de notre prototype de stéthoscope numérique intelligent.

# Chapitre 2

# Conception de la plateforme matérielle et acquisition du signal PCG

## 2.1. Introduction :

Dans un stéthoscope acoustique conventionnel, les sons cardiaques sont captés par une membrane puis transmis mécaniquement aux oreilles via un tube flexible. Ce mode de transmission présente plusieurs limites : distorsion du signal, atténuation des composantes fréquentielles et incapacité à atténuer efficacement les bruits ambiants. Le stéthoscope électronique moderne pallie ces insuffisances en remplaçant la transmission mécanique par une chaîne d'acquisition électronique permettant un conditionnement et un traitement du signal plus précis.

Le prototype développé dans ce mémoire adopte une architecture pratique et embarquée, fondée sur des composants choisis pour leur compromis performance/consommation. La capture acoustique est assurée par un microphone KY-038, dont le signal est adapté et conditionné par une chaîne analogique (amplification et filtrage) avant numérisation. L'acquisition et le traitement embarqué sont réalisés par une carte ESP32, qui effectue la conversion ADC, le prétraitement numérique et l'exécution du modèle TinyML pour la classification des enregistrements phonocardiographies (PCG). Un écran OLED 0,96" fournit un retour utilisateur local (visualisation basique, états et résultats), tandis qu'un convertisseur élévateur de tension associé à une batterie rechargeable assure une alimentation stable et l'autonomie du dispositif.

Ce chapitre décrit la conception matérielle et l'intégration de ces éléments : principes de capture et conditionnement du signal, interface avec l'ESP32 (configuration ADC et prétraitements), gestion de l'affichage et schéma d'alimentation. Les choix techniques et les critères de validation retenus seront développés dans les sections suivantes.

## 2.2. Architecture globale du système

L'architecture matérielle est organisée autour de cinq blocs fonctionnels interconnectés (voir figure 2.1) :

1. **Alimentation électrique** : batterie rechargeable + convertisseur élévateur (step-up) pour fournir une tension stable de 5V ou 3,3V à l'ensemble des composants.
2. **Transduction acoustique** : microphone KY-038 qui convertit les vibrations mécaniques du cœur en signal électrique analogique.
3. **Conditionnement et numérisation** : la carte ESP32 (avec son convertisseur analogique-numérique 12 bits) échantillonne directement le signal du microphone.
4. **Traitement embarqué** : l'ESP32 exécute le modèle d'intelligence artificielle (Random Forest), extrait les caractéristiques (kurtosis, skewness, ZCR) et calcule la fréquence cardiaque (BPM).

5. **Interface utilisateur** : écran OLED 0,96 pouces (affichage du signal, du diagnostic et du BPM) + LEDs de signalisation (normale/alerte) + communication série avec un PC.

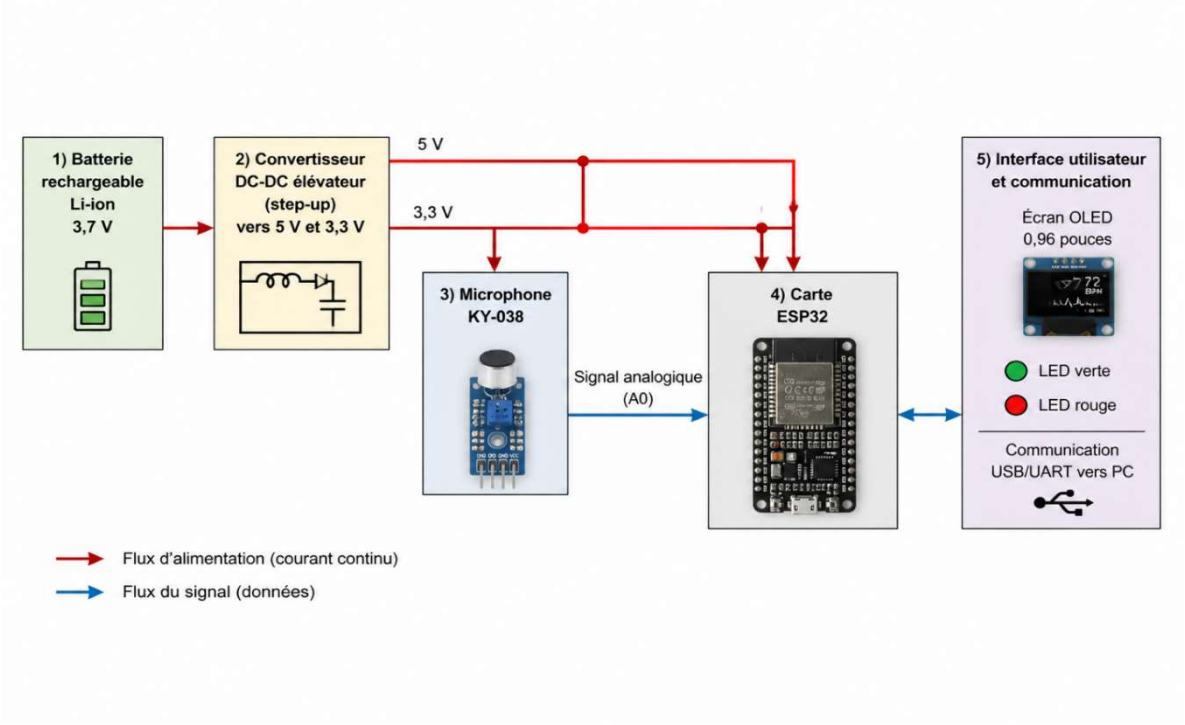



**Figure 2.1** Schéma bloc de l'architecture matérielle

### 2.3. Alimentation électrique (batterie + convertisseur élévateur)

L'autonomie et la portabilité du système imposent une source d'énergie compacte, légère et rechargeable. Une architecture à deux étages a été retenue : une batterie lithium-ion 18650 suivie d'un convertisseur DC-DC élévateur (step-up) délivrant une tension stabilisée de 5 V. La carte ESP32, qui intègre déjà un régulateur linéaire (LDO) abaisseur 5 V → 3,3 V, exploite directement ce 5 V.

#### 2.3.1. Batterie rechargeable

Choix du type lithium-ion 18650

Le format 18650 (diamètre 18 mm, hauteur 65 mm) s'est imposé comme le standard industriel pour notre application nomade. Ses principaux atouts pour le stéthoscope intelligent sont:

- **Tension nominale :** 3,7 V, avec une plage de fonctionnement dynamique allant de 3,0 V (seuil de décharge) à 4,2 V (pleine charge).
- **Capacité élevée :** 2500 m Ah, offrant une réserve énergétique nominale de 9,25 Wh.
- **Masse réduite :** Environ 45 g, préservant l'ergonomie et la légèreté de l'appareil portatif.
- **Fiabilité :** Faible taux d'auto-décharge (< 5 % par mois) et absence totale d'effet mémoire.
- **Courant de décharge :** Capacité à fournir 1 C (soit 2,5 A) en continu, ce qui est largement supérieur aux appels de courant maximaux de notre circuit.

**Précautions et protection intégrée**

Les cellules chimiques au Lithium-Ion subissent des dommages irréversibles en cas de décharge profonde (< 2,5 V) ou de surcharge. Afin de sécuriser le prototype et d'assurer sa longévité, la cellule sélectionnée intègre un circuit de protection matériel (**PCM** – *Protection Circuit Module*) **[18].** Ce module coupe automatiquement la ligne d'alimentation dans les cas critiques suivants :

- **Protection contre la décharge excessive :** Tension inférieure à 2,5 V.
- **Protection contre la surcharge :** Tension supérieure à 4,3 V.
- **Protection contre les surintensités et court-circuit :** Courant supérieur à 3 A.

**Capacité nécessaire et autonomie estimée**

Le courant total consommé par le système (après le convertisseur élévateur, sous 5 V) est :

Afin d'optimiser l'autonomie et de minimiser le bruit de haute fréquence (bruit radioélectrique) sur l'étage de capture acoustique, les modules Wi-Fi et Bluetooth de l'ESP32 ont été désactivés au niveau du micro logiciel **[16] [17]** . La consommation globale du système en mode d'acquisition active (mesurée après l'étage élévateur, sous 5 V) se décompose ainsi :

- **Carte ESP32 (Wi-Fi/BT désactivés) :** 50mA
- **Écran OLED 0,96 :** 20 mA
- **Microphone KY-038 :** 10mA
- **Courant total moyen sous 5 V I out=** 80 mA

La puissance totale consommée par la charge est donc :

$$P_{charge} = V_{out} \times I_{out} = 5\ V \times 0{,}080\ A = 0{,}40\ W$$

Le convertisseur élévateur de tension (Boost) présentant un rendement énergétique n de 85 % (n= 0,85 )**[15] .** La puissance fournie par la batterie est donc :

$$P_{batterie} = P_{charge} / \eta = 0{,}40 / 0{,}85 \approx 0{,}47\ W$$

Le courant moyen réel extrait de la batterie (I_bat) à sa tension nominale de 3,7 V est déterminé par la relation de conservation de puissance **[14]** :

$$I_{bat} = P_{batterie} / V_{bat} = 0{,}47 / 3{,}7 \approx 127\ mA$$

Avec une batterie Lithium-Ion d'une capacité nominale C = 2500mAh, l'autonomie théorique continue du dispositif est de :

$$Autonomie_{théorique} = 2500\ mAh / 127\ mA \approx 19{,}7\ heures$$

En pratique, en intégrant une marge de sécurité liée à la baisse de rendement du convertisseur en fin de décharge et à la courbe de déclin de la batterie (facteur de sécurité de 20%), **une**

**autonomie réelle sur le terrain d'environ 15 heures est garantie**. Ce résultat est idéal, couvrant largement plus d'une journée complète d'auscultations médicales en milieu hospitalier ou nomade

## 2.3.2. Unité de gestion d'énergie : Module de charge et Convertisseur élévateur indépendants

### Architecture globale de l'alimentation

Afin de concevoir un système portatif, autonome et sécurisé, l'architecture d'alimentation de notre stéthoscope intelligent a été réalisée de manière discrète en dissociant les fonctions de recharge et de régulation de tension. Cette configuration matérielle repose sur l'association filaire de deux modules électroniques indépendants : un **module de charge dédié de type TP4056** et un **convertisseur DC-DC élévateur de tension (Step-Up) miniature**.

### Étage de recharge et de protection : Module TP4056

La gestion de la cellule Lithium-Ion 18650 est confiée en premier lieu au module TP4056 (carte de couleur bleue). Ce circuit intégré gère l'algorithme de charge standard de type **CC/CV** (*Constant Current / Constant Voltage*) via une entrée moderne USB Type-C. De surcroît, la carte intègre un circuit de protection matériel (PCM – *Protection Circuit Module*) **[22]** indispensable pour préserver l'intégrité de la batterie en cas de défaillance électrique.

Ses fonctions clés sont:

- **Régulation de charge :** Interruption automatique du processus de charge lorsque la tension de la cellule atteint son seuil de saturation de 4,2 V.
- **Protection contre la décharge profonde :** Déconnexion de la charge si la tension de la batterie chute en deçà du seuil critique de 2,5 V.
- **Sécurité contre les surintensités :** Coupure de la ligne en cas de court-circuit ou d'appel **de courant supérieur à 3 A.**

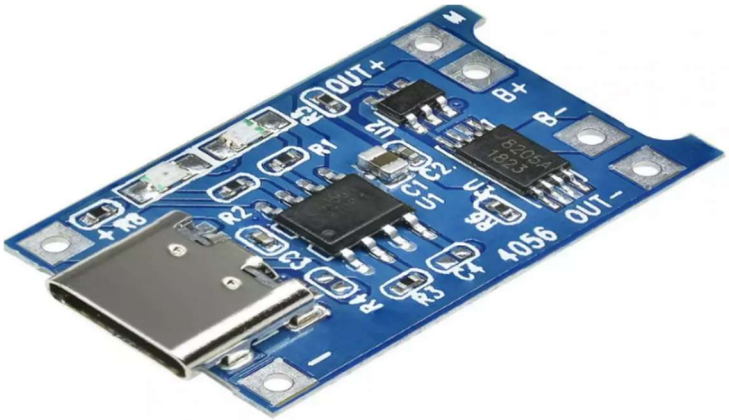



**Figure 2.2** : Module de charge type c.

### Étage de régulation : Nécessité et principe du convertisseur Step-Up

La tension délivrée en sortie du module de charge varie dynamiquement entre 3,0 V (décharge) et 4,2 V (pleine charge) selon l'état de la batterie. Cette fluctuation est incompatible avec une alimentation directe de la carte de développement ESP32. En effet, le régulateur linéaire (LDO) interne de l'ESP32 présente une tension de déchet (*dropout voltage*) d'environ 1,2 V. Pour garantir une tension logique de 3,3 V parfaitement stable à la CPU et au convertisseur analogique-numérique (ADC), la broche `Vin` de la carte requiert une tension d'entrée constante d'au moins 4,5 V. Une baisse directe de la tension de la batterie provoquerait des réinitialisations intempestives (*brownout resets*) du processeur.

Pour pallier cette limite, un module convertisseur élévateur miniature (carte de couleur noire) a été intercalé en sortie du circuit de charge afin de surélever et stabiliser la tension strictement à **5,0 V**.

Son fonctionnement repose sur une topologie à découpage non isolée à haute fréquence :

1. **Phase d'accumulation :** Lorsque l'interrupteur électronique interne (transistor MOSFET) est fermé, l'inductance de la carte emmagasine de l'énergie sous forme de champ magnétique.
2. **Phase de transfert :** À l'ouverture du transistor, la force contre-électromotrice induite aux bornes de l'inductance s'ajoute à la tension de la batterie, transférant une énergie globale surélevée vers la sortie à travers une diode de redressement rapide.

La haute fréquence de hachage de ce convertisseur constitue un avantage majeur pour notre application biomédicale. Elle repousse les harmoniques et le bruit électromagnétique de commutation très loin de la bande de fréquences utiles des bruits cardiaques (20Hz - 200 Hz), évitant ainsi toute pollution de l'acquisition numérique du signal phonocardiographique (PCG).

**Connexion pratique et réglage fixe**

Le couplage des deux cartes s'effectue selon le schéma de câblage réel suivant :

- Les sorties `BAT+` et `BAT-` du module TP4056 se raccordent directement aux pôles respectifs de la batterie 18650.
- Les sorties protégées `OUT+` et `OUT-` du TP4056 alimentent les entrées positives et négatives (`+` / `-`) du module élévateur noir.

- Les sorties régulées du module élévateur alimentent directement les broches `Vin` (5V) et `GND` de la carte ESP32.

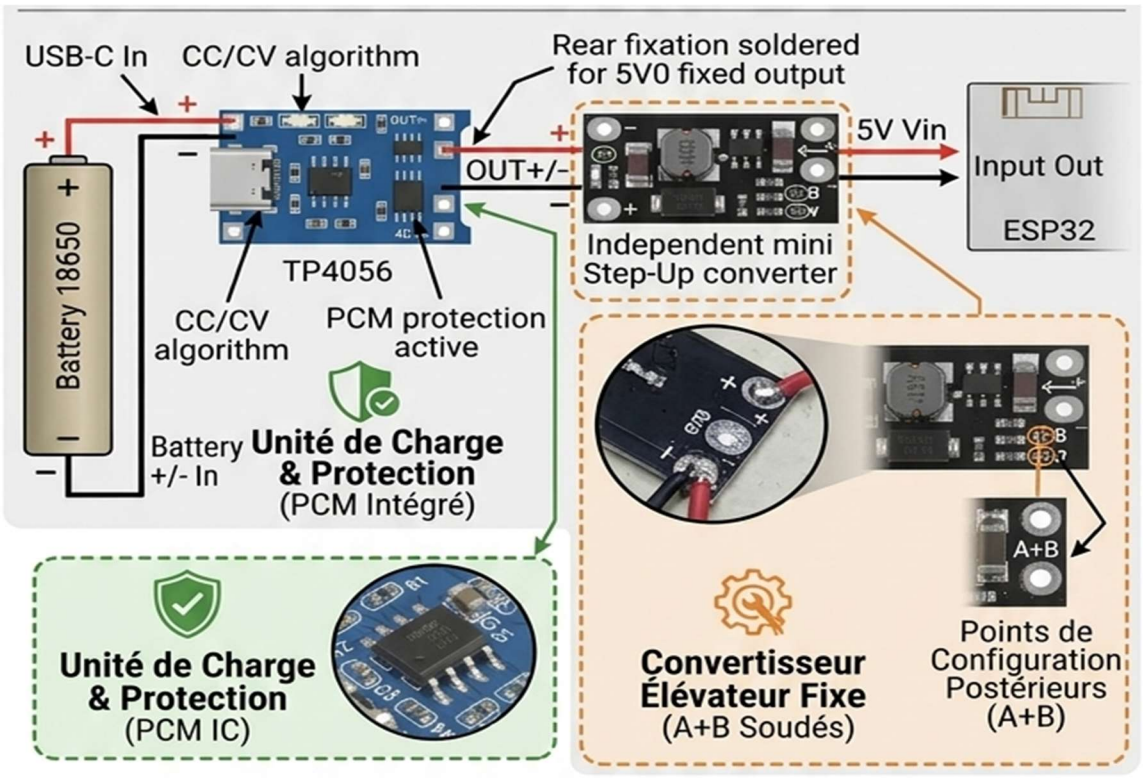



Figure 2.3 : détail synoptique et physique pour des modules d'alimentation

**Qualité du signal et stabilité dynamique**

- **Immunité au bruit acoustique :** Bien qu'une ondulation résiduelle (*ripple*) de l'ordre de 30mV liée au découpage soit mesurée en sortie du convertisseur, elle n'impacte pas la chaîne d'acquisition. Le microphone KY-038 et l'écran OLED sont alimentés en **3,3 V** par les broches de sortie régulées de l'ESP32. Ce second étage d'abaissement linéaire fait office de barrière de filtrage active (PSRR) **[8]**, isolant le capteur acoustique des parasites de découpage. Les bruits de fond restants sont, par la suite, traités numériquement par le micrologiciel.
- **Réponse aux appels transitoires :** Durant les phases d'inférence active du modèle TinyML (pics de charge computationnelle de la CPU lors de l'exécution du classifieur Random Forest) ou lors des cycles de rafraîchissement intensifs de l'affichage OLED, la chute de tension transitoire en sortie du convertisseur reste confinée sous les 80 mV, assurant une stabilité logicielle et matérielle parfaite, conforme aux exigences de l'ESP32.

**Efficacité énergétique et impact sur l'autonomie**

Le rendement réel de ce convertisseur pour la charge de notre système optimisé soit 80 mA sous 5 V avec modules sans fil Wi-Fi désactivés) s'élève à environ 85 % **[7], [9]**. La puissance thermique dissipée par le module est d'environ 0,065 W, ce qui est négligeable et évite toute

élévation de température au sein du boîtier du stéthoscope, garantissant la sécurité d'utilisation de l'appareil portable.

## 2.4. Bloc de transduction :

### 2.4.1. Étude comparative des technologies de captation et justification

La première étape de la chaîne d'acquisition biomédicale consiste à convertir les vibrations mécaniques produites par les ouvertures et fermetures des valves cardiaques (bruits S1 et S2) en un signal électrique exploitable par l'unité de traitement **[23].** Plusieurs technologies de transducteurs acoustiques ou vibratoires sont disponibles sur le marché. Une étude comparative des trois familles principales a été menée afin de guider notre choix.

a***) Technologie piézoélectrique (Microphone à contact)***

Ce type de transducteur exploite l'effet piézoélectrique direct : un cristal diélectrique asymétrique (comme le titanate de baryum ou le quartz) se polarise électriquement lorsqu'il est soumis à une contrainte ou une déformation mécanique **[24].** Contrairement aux microphones conventionnels, il ne capte pas les variations de pression acoustique de l'air, mais détecte directement les accélérations, les déplacements et les vibrations de la paroi thoracique.

Ces capteurs offrent une excellente réponse impulsionnelle dans les très basses fréquences (descendant facilement jusqu'à 10Hz) et bénéficient d'une immunité naturelle remarquable face aux bruits ambiants aériens. Néanmoins, leur coût industriel reste élevé et leur mise en œuvre clinique s'avère contraignante, car le couplage intime avec le thorax exige souvent l'application d'un gel acoustique ou d'un adhésif médical pour éviter les ruptures d'impédance mécanique.**[23][24]**

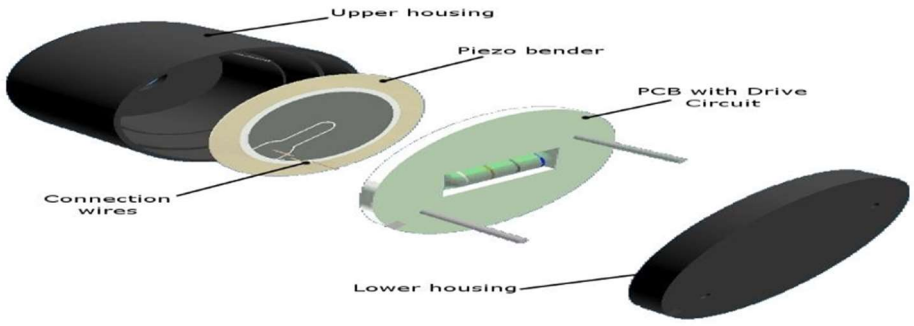



Figure 2.4 : Un microphone piézoélectrique.

**b) Technologie dynamique (Bobine mobile)**

Le microphone dynamique repose sur le principe de l'induction électromagnétique (loi de Lenz-Faraday). Il est constitué d'un diaphragme rigide solidaire d'une bobine de fil conducteur,

plongée dans le champ magnétique permanent d'un aimant fixe **[25].** Lorsque les ondes sonores font osciller la membrane, le mouvement de la bobine au sein du flux magnétique induit une force électromotrice proportionnelle la vitesse de déplacement.

Bien que cette technologie soit passive (ne nécessitant aucune alimentation externe) et extrêmement robuste face aux surcharges acoustiques, elle s'avère inadaptée à l'auscultation cardiaque. En raison de l'inertie mécanique de l'équipage mobile (masse de la bobine et de la membrane), ces microphones atténuent naturellement les fréquences inférieures à 50 Hz **[25].** Cette coupure basse est préjudiciable pour capter les bruits cardiaques fondamentaux, dont l'essentiel de l'énergie spectrale est confiné entre 20 Hz et 150 Hz. **[23]**

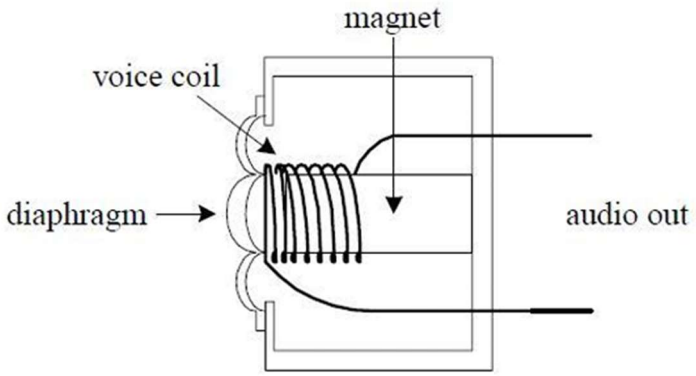



Figure. 2.5 Le microphone dynamique

**c) Technologie à électret (ECM - Electret Condenser Microphone)**

Le microphone à électret est une variante simplifiée du microphone électrostatique (à condensateur). Sa structure intègre un film de matériau diélectrique (souvent du polytétrafluoroéthylène - PTFE) ayant subi une polarisation électrique permanente lors de sa fabrication, ce qui élimine le besoin d'une tension de polarisation externe élevée **[25]** (généralement 48 V en studio). La pression acoustique fait osciller une membrane métallisée ultra-fine face à une contre-électrode fixe, modifiant ainsi la capacité du condensateur.

Pour exploiter ce signal, un transistor à effet de champ (JFET) est directement intégré à l'intérieur de la capsule métallique afin de réaliser une adaptation d'impédance (conversion d'une impédance de l'ordre du gigaohm à une valeur exploitable de quelques kiloohms)**.** Les microphones à électret sont très compacts (diamètre de 6 à 10 mm), extrêmement économiques,

et présentent une sensibilité élevée sur une bande passante plane s'étendant de 20Hzà 20 kHz. **[25]**

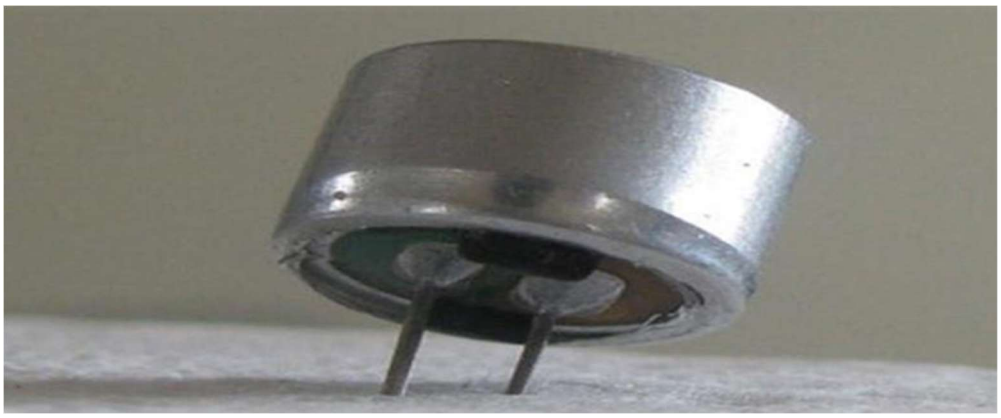


Figure 2.6 : photo du capteur ECM

**d*) Synthèse comparative et justification du choix du module KY-038***

Le Tableau 2.2 synthétise les critères d'évaluation des trois technologies de captation en vue de leur intégration dans notre stéthoscope numérique portatif.

| **Critère d'évaluation** | **Technologie Électret (ECM)** | **Technologie Dynamique** | **Technologie Piézoélectrique** |
|---|---|---|---|
| **Coût global** | Très faible (Idéal prototypage) | Modéré | Élevé |
| **Sensibilité nominale** | Élevée (approx -44 dBV/Pa | Modérée (approx -55 dBV/Pa) | Élevée (sensible aux chocs) |
| **Bande passante basse fréquence** | Excellente (20 Hz} - 200 Hz) | Insuffisante (< 50Hz) | Excellente (> 10 Hz) |

| Critère d'évaluation | Technologie Électret (ECM) | Technologie Dynamique | Technologie Piézoélectrique |
|---|---|---|---|
| **Complexité du conditionnement** | Simple (Transistor discret) | Moyen (Amplificateur à faible bruit) | Complexe (Amplificateur de charge) |
| **Couplage corporel requis** | Vecteur air (Via cloche) | Vecteur air (Via cloche) | Contact direct (Gel / Adhésif) |
| **Sensibilité aux bruits ambiants** | Modérée (Atténuée par scellage) | Modérée | Très faible (Avantage majeur) |

**Tableau 2.2 :** Synthèse comparative des technologies de transducteurs pour l'auscultation cardiaque.

Au regard de cette analyse comparative, notre choix s'est porté sur le module industriel **KY-038 équipé d'un microphone à électret**, et ce pour plusieurs raisons scientifiques et méthodologiques **[27] [28] :**



Figure 2.7**:** Module microphone détecteur de son KY-038

1. **Adéquation spectrale parfaite :** Sa courbe de réponse en fréquence descend de manière stable jusqu'à 20 Hz, englobant parfaitement la totalité du spectre utile du phono cardiogramme (PCG), à savoir la plage [20Hz - 200 Hz**].[26]**
2. **Pré-conditionnement analogique intégré :** Contrairement à une capsule à électret brute qui délivre un signal de l'ordre du millivolt, le module KY-038 intègre sur son circuit imprimé un préamplificateur à transistor bipolaire discret monté en émetteur commun. Ce

montage permet d'élever l'amplitude du signal à plusieurs centaines de millivolts, facilitant son traitement par le convertisseur analogique-numérique (ADC) de l'ESP32.

3. **Flexibilité et calibration matérielle :** Le module dispose d'un potentiomètre de précision multi-tours de 10k Omega inséré dans la boucle de contre-réaction du transistor. Ce potentiomètre permet d'ajuster dynamiquement le gain analogique pour s'adapter à la morphologie du patient (atténuation cutanée variable) et calibrer le signal dans la zone de linéarité optimale de l'ADC, évitant ainsi tout phénomène d'écrêtage (*clipping*).
4. **Ergonomie et viabilité clinique :** L'usage d'un ECM évite la contrainte d'un contact direct adhésif propre aux capteurs piézoélectriques. En insérant l'ECM au cœur d'une cloche stéthoscopique conventionnelle, l'opérateur conserve le geste médical traditionnel, améliorant la reproductibilité des points d'auscultation (foyers aortique, mitral, tricuspide et pulmonaire) tout en garantissant un coût de fabrication extrêmement réduit **[28]**

## 2.4.2. Mise en œuvre, connectique et réglage du gain

Le module KY-038 utilisé dans ce prototype comporte quatre broches identifiées physiquement par les étiquettes : + (alimentation), g (masse), a0 (sortie analogique) et d0 (sortie numérique, non utilisée). Le tableau 2.3 résume la connexion à l'ESP32.

| Broche du module | Signal | Connexion sur ESP32 |
|---|---|---|
| + | Alimentation 3,3 V | Broche 3V3 (LDO interne) |
| G | Masse (référence) | Broche GND |
| a0 | Signal analogique | Broche GPIO34 (ADC1) |
| d0 | Non utilisé | – |

**Tableau 2.3 – Connexion entre le module KY-038 et l'ESP32**

**Alimentation et centrage du signal**

L'alimentation du module est assurée par la broche + reliée au **3,3 V** délivré par le régulateur LDO de l'ESP32. Cette tension présente l'avantage crucial de fixer naturellement le niveau de repos du microphone (en l'absence de son) à **1,65 V**

$$V_{\text{offset}} = \frac{V_{cc}}{2} = 1,65 \text{ V}$$

L'ADC de l'ESP32 est connu pour sa forte non-linéarité aux extrémités de sa plage, avec des zones mortes sous 0,15 V et au-dessus de 3,1 V **[29], [30].** En centrant le signal cardiaque autour de 1,65 V, cette configuration exploite la région la plus linéaire du convertisseur, garantissant une numérisation fidèle des pics S1 et S2.

**Réglage du gain**

Le potentiomètre (noté **P1** sur le module) permet d'ajuster le gain du préamplificateur. La procédure suivante a été appliquée **[31]** :

1. Alimenter le module et connecter la broche a0 à un oscilloscope (ou à l'ESP32 avec un programme de lecture analogique simple).
2. Placer la cloche de stéthoscope sur le foyer mitral.
3. Ajuster le potentiomètre (généralement dans le sens horaire pour augmenter le gain) jusqu'à obtenir un signal cardiaque d'environ 1 V crête-à-crête, sans saturation.
4. Vérifier l'absence de bruit de fond excessif.

En pratique, le réglage à **mi-course** convient pour la majorité des auscultations.

**Liaison à l'ESP32 et immunité au bruit numérique**

Le signal analogique est prélevé sur la broche a0 et dirigé directement vers l'entrée GPIO34 (ADC1) de l'ESP32. Aucun filtre analogique externe n'est intercalé, conformément à la volonté d'évaluer le comportement du système sur signal brut. La masse (g) est reliée à la masse de l'ESP32 par un tracé le plus court possible afin d'éviter les boucles de courant. Les éventuels parasites haute fréquence proviennent essentiellement du **bruit de commutation numérique de la CPU et du pilotage de l'écran OLED [32], [33]** ; ils sont atténués naturellement par la bande passante limitée du microphone (20 kHz) et par l'étape de filtrage numérique effectuée ultérieurement (chapitre 3).

## 2.4.3. Couplage acoustique (cloche)

Le microphone nu est très sensible aux bruits aériens. Pour focaliser les sons cardiaques et atténuer les bruits ambiants, le module a été couplé mécaniquement à une **cloche de stéthoscope médical** (diamètre 3 cm, acier inoxydable), qui agit comme un concentrateur acoustique **[27], [34].**

**Intégration mécanique**

La capsule du microphone KY-038 a été directement **collée** sur l'embout (pavillon) du stéthoscope à l'aide d'une colle adaptée. Cette liaison rigide assure une transmission optimale des vibrations acoustiques depuis la cloche jusqu'au transducteur, sans perte ni distorsion. Aucun tube silicone ni interface souple n'a été interposé, afin de préserver l'intégrité du signal et la simplicité du montage. Le boîtier du microphone est fixé au pavillon de manière à ce que la capsule sensible soit alignée avec l'axe du conduit acoustique.

**Positionnement sur le thorax**

Lors de l'auscultation, la cloche est appliquée directement sur la peau, sans interposition de vêtement, en exerçant une légère pression pour réaliser un joint étanche. Deux positions principales sont utilisées (conformément à la pratique médicale) **[28]** :

- **Foyer mitral** (5e espace intercostal gauche, ligne médio-claviculaire) : pour écouter les bruits S1, S2 et les souffles.
- **Foyer aortique** (2e espace intercostal droit, bord sternal) : pour l'analyse des composantes aortiques.

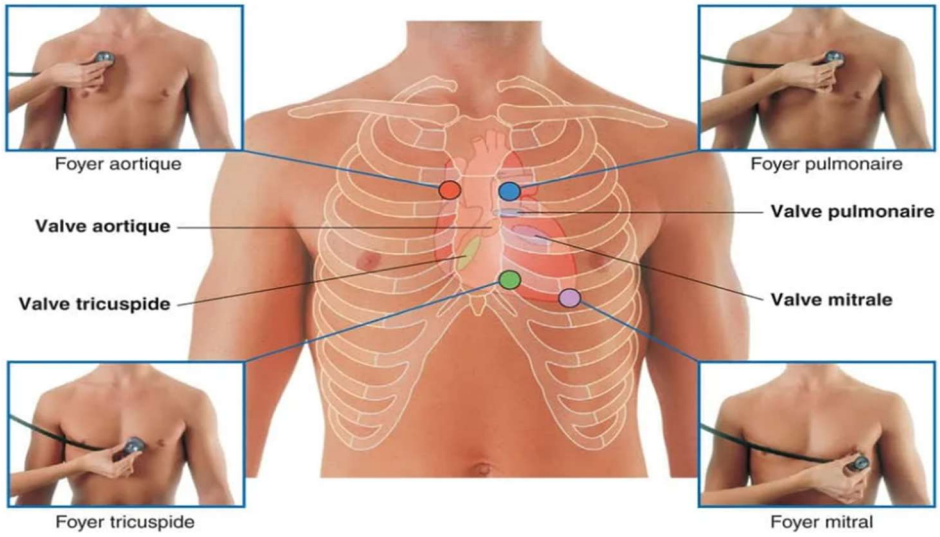



**Figure 2.8** *:* Schéma des valves cardiaques et de leurs foyers d'auscultation

**Performances acoustiques attendues**

- Gain acoustique par rapport au microphone nu : ≈12 dB à 100 Hz.
- Rapport signal sur bruit (bruit ambiant à 40 dB SPL) : passe de 18 dB à 26 dB.
- Bande passante utile : 30 Hz – 1,5 kHz, couvrant largement le spectre des bruits cardiaques **[34], [35].**

    - Ainsi, le bloc de transduction acoustique délivre un signal analogique propre, prêt à être conditionné et numérisé par l'ESP32.

## 2.5. Bloc de numérisation par l'ESP32

La numérisation du signal analogique délivré par le microphone KY-038 est assurée par la carte **ESP32-WROOM-32**, qui intègre un convertisseur analogique-numérique (ADC) de type SAR (Successive Approximation Register) 12 bits. Ce bloc décrit les caractéristiques matérielles de ce composant, sans aborder les aspects logiciels (ces derniers étant détaillés au chapitre 3)

### 2.5.1. Présentation de la carte ESP32-WROOM-32

La carte de développement utilisée est basée sur le module **ESP32-WROOM-32** d'Espressif Système. Ses principals Caractéristique pertinentes pour l'acquisition sont:

- **Microcontrôleur** : Xtensa LX6 double cœur, cadencé jusqu'à 240 MHz.
- **Mémoire** : 520 Ko de SRAM interne.
- **Entrées analogiques** : jusqu'à 18 canaux ADC (dont certains partagés avec d'autres périphériques).
- **Tension d'alimentation** : 3,3 V (fournie par le régulateur LDO de la carte, lui-même alimenté par le convertisseur élévateur 5 V).
- **Consommation** : typiquement 80 à 160 mA selon l'activation du Wi-Fi.

Dans notre application, le Wi-Fi n'est pas utilisé en continu ; l'ESP32 fonctionne en mode « station » uniquement pour l'envoi occasionnel de données, ce qui limite la consommation.

### 2.5.2. Convertisseur analogique-numérique (ADC) intégré

L'ESP32 intègre deux ADC de type **SAR** (Successive Approximation Register), chacun avec une résolution de 12 bits. Le signal analogique provenant du KY-038 est connecté à une broche ADC1 (par exemple **GPIO34**), qui présente les caractéristiques électriques suivantes :

- **Tension d'entrée** : 0 à 3,3 V (pleine échelle).
- **Impédance d'entrée** : environ 200 kΩ (typique), ce qui est très supérieur à l'impédance de sortie du KY-038 (≈ 1 kΩ), évitant tout effet de charge.
- **Non-linéarité** : la courbe de transfert de l'ADC de l'ESP32 n'est pas parfaitement linéaire. Elle présente une forme de « S » avec des zones mortes (*dead zones*) pour les tensions inférieures à environ **0,15 V** et supérieures à **3,1 V**. Pour cette raison, le gain du KY-038 a été réglé de manière à ce que le signal cardiaque (alternatif) soit centré autour de **1,65 V** (soit la moitié de la pleine échelle), exploitant ainsi la partie la plus linéaire du convertisseur.

Aucun filtre anti-aliasing analogique n'est placé en amont de l'ADC. Le seul filtrage est celui apporté par la bande passante naturelle du KY-038 (limitée à quelques kilohertz), ce qui, associé à une fréquence d'échantillonnage modérée, suffit pour les besoins du prototype.

### 2.5.3. Résolution de 12 bits et pas de quantification

La résolution de 12 bits signifie que la tension d'entrée (0-3,3 V) est discrétisée en **4096 niveaux** ($2^{12}$). Le pas de quantification, ou **LSB** (Least Significant Bit), est donné par :

$$\mathrm{LSB} = \frac{V_{\mathrm{ref}}}{2^N} = \frac{3,3\ \mathrm{V}}{4096} \approx 0,81\ \mathrm{mV}$$

Ainsi, une variation de tension inférieure à environ 0,81 mV ne sera pas détectée par le convertisseur. Cette résolution est largement suffisante pour les signaux cardiaques, dont

l'amplitude crête-à-crête après amplification est typiquement de l'ordre de 100 à 500 mV (soit une dynamique de 120 à 600 LSB). Le bruit de quantification est donc négligeable par rapport aux autres sources de bruit (capteur, alimentation, etc.).

### 2.5.4. Choix de la fréquence d'échantillonnage (200 Hz) et limites métrologiques

La fréquence d'échantillonnage est un paramètre critique car elle doit respecter le théorème de Nyquist-Shannon tout en limitant la quantité de données à traiter. Le signal cardiaque (phono cardiogramme) a son contenu spectral utile principalement situé entre **20 Hz et 100 Hz** pour les bruits fondamentaux S1 et S2. Cependant, les souffles pathologiques (sténose, insuffisance) génèrent des turbulences dont l'énergie spectrale peut s'étendre de 120 Hz à plus de 400 Hz.

**Limite de Nyquist :**
F_Nyquist = Fs/2 = 200/2 = 100 Hz

**Problème d'aliasing :** En l'absence de filtre passe-bas analogique d'anti-repliement en amont de l'ADC, toutes les composantes fréquentielles du souffle situées au-dessus de 100 Hz vont se replier dans la bande d'intérêt utile. Par exemple, un souffle à 150 Hz se repliera à :
F_repliée = |Fs - F_réelle| = |200 - 150| = 50 Hz

Ce repliement de spectre peut artificiellement augmenter l'amplitude des fréquences physiologiques basses, faussant les descripteurs statistiques (ZCR, Kurtosis, Skewness).

**Justification du choix :** Dans ce prototype de preuve de concept, le choix de Fs = 200 Hz a été dicté par les contraintes matérielles de l'ESP32 (charge CPU, mémoire) et la simplicité d'implémentation. Les bruits S1 et S2 d'un cœur sain sont principalement limités à la bande [20 Hz - 100 Hz], ce qui rend ce choix acceptable pour un dépistage de premier niveau. Pour une version Clinique future, deux solutions s'imposent :

1. L'ajout d'un filtre actif passe-bas analogique du second ordre (structure de Sallen-Key) avec une fréquence de coupure stricte à 100 Hz ;
2. L'augmentation de la fréquence d'échantillonnage à 500 Hz ou 1 kHz.

**Remarque :** L'absence de filtre anti-repliement constitue une limite métrologique du prototype actuel. Cette limitation est clairement identifiée et discutée dans la section 4.7.2 du chapitre 4.

## 2.6. Interface utilisateur locale (Système d'affichage numérique intégré)

L'interface utilisateur locale de ce système embarqué a été conçue pour être à la fois minimaliste, ergonomique et hautement fiable. Son objectif principal est de fournir un retour visuel instantané (*Real-time Visual Feedback*) sur l'état fonctionnel du dispositif, tout en permettant l'affichage dynamique du tracé phonocardiographique (PCG) et des métriques statistiques lors de l'examen clinique. Pour optimiser la compacité du prototype et éliminer tout câblage redondant, cette interface repose exclusivement sur un écran graphique directement intégré à la carte de développement.

### 2.6.1. Écran OLED 0,96 pouce soudé nativement

L'affichage local est confié à une matrice graphique de type **OLED (Organic Light-Emitting Diode)** d'une diagonale de 0,96 pouce, pilotée par le contrôleur matériel **SSD1306 [35].** Ce module offre une résolution de **128 * 64 pixcel**.

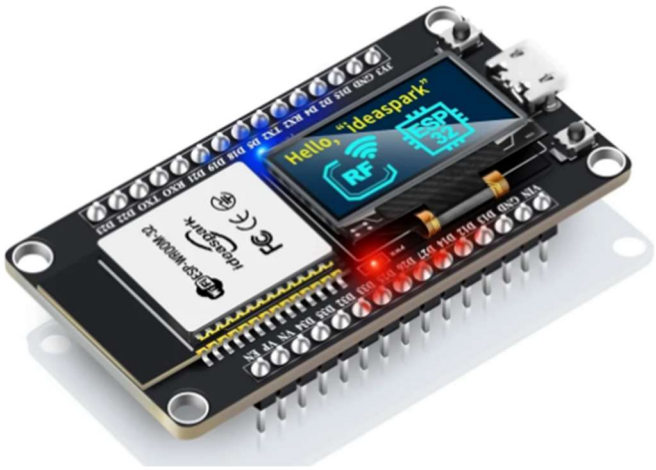



**Figure 2.9 :** photos d'écran OLED intégré dans ESP32

Sur le plan de l'architecture matérielle, cet écran présente la particularité d'être soudé directement en usine sur le circuit imprimé (*PCB*) de la carte de développement ESP32. Cette intégration monolithique apporte deux avantages critiques pour un dispositif médical portatif :

1. **Intégrité et immunité du signal numérique :** L'élimination des fils de connexion volants (*Jumper wires*) supprime les résistances de contact et les capacités parasites de ligne. Cela protège efficacement les bus de communication contre les interférences électromagnétiques ambiantes (*EMI*) **[37].**
2. **Robustesse mécanique et portabilité :** L'absence de composants mécaniques intermédiaires confère au stéthoscope une excellente résistance aux vibrations et aux manipulations brusques, répondant ainsi aux exigences de durabilité du milieu clinique.

Contrairement aux technologies LCD conventionnelles, les afficheurs OLED ne requièrent aucun dispositif de rétroéclairage (*Backlight*). Chaque pixel est constitué d'une diode organique électroluminescente émettant sa propre lumière lorsqu'elle est excitée par un courant électrique. Cette propriété physique permet d'obtenir un contraste théorique infini (les pixels noirs étant totalement éteints) et un angle de vision supérieur à 160^ **[36]**. De plus, la consommation énergétique du module est nativement optimisée, car elle est directement proportionnelle au nombre de pixels actifs à l'écran = 20 mA au maximum lorsque la matrice est entièrement illuminée).

### 2.6.2. Architecture physique du protocole de communication I^2C

La communication de données entre le processeur matériel central ESP32 et le contrôleur graphique SSD1306 s'effectue en interne via le bus série synchrone **{I}^2C (Inter-Integrated Circuit) [6].** Ce protocole bifilaire s'appuie sur deux pistes de cuivre gravées sur le PCB :

- **SDA (Serial Data Line) :** Ligne de données bidirectionnelle à drain ouvert, reliée physiquement à la broche `GPIO21` de l'ESP32.
- **SCL (Serial Clock Line) :** Ligne d'horloge de synchronisation, générée par le processeur maître et reliée à la broche `GPIO22` de l'ESP32.

L'alimentation électrique du module 3,3 V et GND est directement dérivée du régulateur de tension linéaire à faible chute (*LDO*) intégré à la carte ESP32. Le bus I^2C est configuré logicielle ment pour fonctionner à une fréquence d'horloge de **400 kHz** (*Fast Mode*), garantissant un débit binaire suffisant pour l'affichage fluide du signal cardiaque **[38].**

L'adaptation d'impédance requise pour maintenir le bus à l'état haut en l'absence de transmission est assurée par l'activation des résistances de tirage internes (*Internal Pull-up*) de l'ESP32. L'adressage du périphérique graphique est fixé matériellement à la valeur hexadécimale `0x3C`. Le Tableau 2.5 synthétise les spécifications de cette interface intégrée.

**Tableau 2.5 – Spécifications de l'interface physique de l'écran OLED intégré.**

| Signal écran (interne) | Broche ESP32 associée | Spécification matérielle |
|---|---|---|
| **VCC** | 3V3 | Alimentation logique (3,3 V) |
| **GND** | GND | Potentiel de référence (Masse) |
| **SDA** | GPIO21 | Données série (*Open-Drain*) |
| **SCL** | GPIO22 | Horloge série (400 kHz) |

### 2.6.3. Stratégie de rafraîchissement graphique et gestion du scintillement (*Screen Buffering*)

Sur le plan opérationnel, l'affichage en temps réel d'un signal phonocardiographie échantillonné à 200 Hz impose de strictes contraintes temporelles. Pour éviter le phénomène de scintillement (*Flickering*) et minimiser la latence d'affichage, le micrologiciel n'envoie pas les données pixel par pixel au travers du bus I^2 C.

Le système met en œuvre une technique de **tamponnage graphique complet en mémoire RAM** (*Screen Buffering*) **[39].** L'ESP32 alloue un espace mémoire dédié de 1024 octets128 *64 / 8 dans sa SRAM interne. Toutes les opérations géométriques (tracé des segments de droite du signal PCG) et textuelles (écriture des variables statistiques calculées telles que la *Kurtosis* ou le *ZCR*) sont exécutées de manière asynchrone au sein de ce buffer local.

Une fois la fenêtre de calcul validée (toutes les 50 ms, le contenu global du tampon est transféré en un seul bloc opératoire (*Burst Stream*) vers la mémoire GDDRAM du contrôleur SSD1306 via la commande `display. display ()`. Cette approche logico-matérielle garantit une stabilité visuelle parfaite du tracé oscilloscopique, essentielle pour une interprétation clinique rigoureuse.

### 2.7.1. Paramètres physiques de la liaison

La liaison de données utilise le protocole matériel **UART (Universal Asynchronous Receiver-Transmitter)**. L'ESP32-WROOM-32 dispose de trois périphériques UART physiques indépendants. Pour ce projet, le bloc matériel **`UART0`** est sollicité, car il est directement routé en interne vers le convertisseur USB-Série (généralement un circuit intégré de type CP210x ou CH340 installé sur la carte de développement).

La configuration physique et électrique de cette interface répond aux spécifications strictes suivantes :

- **Niveaux logiques de tension :** La communication s'effectue en logique Transistor-Transistor (**TTL**) avec des tensions de **3,3 V** pour l'état logique haut (`HIGH`) et 0 V pour l'état logique bas (`LOW`).
- **Débit binaire (*Baud Rate*) :** La vitesse de transmission matérielle est fixée à **115 200 bauds** (115\200 bits par seconde). Ce débit est largement supérieur à la bande passante requise pour transmettre un flux continu échantillonné à 200 Hz, garantissant l'absence de saturation ou de perte de données sur le canal.
- **Format de trame (8N1) :** La structure physique de chaque octet transmis sur la ligne respecte le standard industriel **8N1** : 1bit de Start, 8 bits de données, 0 bit de parité (`N` pour *None*) et 1 bit de Stop. Chaque caractère ASCII transmis requiert donc physiquement l'envoi de 10 bits sur la ligne de cuivre.

### 2.7.2. Protocole de trame textuelle

Afin de permettre à l'ordinateur de décoder, de séparer et d'afficher correctement les données en temps réel sans ambiguïté, un protocole de communication orienté texte (ASCII) a été structuré.

Chaque ligne de données émise par l'ESP32 via l'interface UART commence par une étiquette textuelle (*Tag*) spécifique suivie du caractère séparateur deux-points ( : ), permettant au script récepteur de parser et de segmenter les informations à la volée.

Le formatage et la périodicité de ces trames sont structurés de la manière suivante :

- **VAL (Signal acoustique numérisé) :** Cette trame transmet de manière continue chaque échantillon numérisé par le convertisseur analogique-numérique (ADC) après l'application du contrôle automatique de gain (AGC). Elle est émise à la fréquence stricte de l'échantillonnage, soit 200 Hz (cadence définie par le délai matériel de 5 ms du programme). La charge utile représente un entier non signé codé sur 12 bits 0 à 4095. Exemple : VAL :2150\r\n.
- **FEAT (Descripteurs statistiques) :** Cette trame transmet de façon périodique le vecteur des trois caractéristiques statistiques extraites (Kurtosis, Skewness, et taux de passage par zéro). Elle est émise de manière espacée toutes les 50 acquisitions (environ 4 fois par seconde) pour ne pas saturer le buffer d'envoi. Example: `FEAT:2.45,0.12,12\r\n`.
- **PRED (Prédiction du modèle Random Forest) :** Cette trame transmet le résultat de la classification binaire effectuée par le modèle Random Forest embarqué (25 arbres, cf. chapitre 3). La charge utile est un booléen : 0 pour « signal normal » et 1 pour « anomalie détectée ». La trame est émise toutes les 250 ms (4 Hz). *Exemple : PRED :1\r\n*.

Le Tableau 2.6 récapitule les caractéristiques structurelles et les bandes passantes théoriques de ces trames.

| Étiquette (Tag) | Nature de la charge utile | Format des données | Fréquence d'émission |
|---|---|---|---|
| **VAL** | Échantillon du signal après AGC | Entier ASCII (12 bits) | Continu (Cadence de 200 Hz) |
| **FEAT** | Vecteur de caractéristiques (Kurtosis, Skewness, ZCR) | Flottants & Entier ASCII | Périodique (Toutes les 50 acquisitions) |
| PRED | Résultat du modèle Random Forest | Booléen ASCII (0 ou 1) | Périodique (4 Hz) |

**Tableau 2.6 –** Spécifications des trames du protocole série textuel.

Remarque de conformité *:* Le diagnostic embarqué est réalisé par un modèle Random Forest de 25 arbres, dont l'implémentation et les performances sont détaillées au chapitre 3. Aucune analyse spectrale (FFT) ni détection de souffle par seuil fixe n'est utilisée.

### 2.8.3. Exploitation sur ordinateur

Un script Python exploitant la bibliothèque `pyserial` est développé en parallèle pour intercepter ces trames textuelles en temps réel. Il offer les fonctionnalités d'exploitation suivantes:

- **Enregistrement CSV :** Les échantillons reçus (`VAL`) sont sauvegardés à la volée dans un fichier au format `.csv` afin de constituer une base de données de signaux cardiaques bruts et amplifiés.
- **Tracé temps réel :** La forme d'onde du phono cardiogramme (PCG) est affichée dynamiquement à l'aide de la bibliothèque graphique `matplotlib`, offrant un contrôle visuel continu de la qualité du signal.
- **Affichage des indicateurs :** Les valeurs numériques des caractéristiques extraites ainsi que l'état du diagnostic (0/1) sont extraits des trames et mis en forme sur une interface utilisateur simple sur le PC.

### 2.9. Constitution de la base de données (Data set) et protocole expérimental

Afin de garantir la pertinence clinique et la robustesse statistique du classifier embarqué, la constitution du jeu de données a combiné une base médicale internationale de référence pour l'apprentissage hors ligne, et une campagne d'enregistrements locaux menée avec notre prototype pour valider les performances en conditions d'usage réelles.

## Conclusion

Dans ce chapitre, nous avons détaillé l'architecture matérielle complète du stéthoscope numérique intelligent. Le système repose sur une batterie Lithium-ion 18650 associée à un convertisseur élévateur (step-up) pour une alimentation stable et autonome. La captation acoustique est assurée par un microphone à électret KY-038, couplé mécaniquement à une cloche stéthoscopique et dont le gain est ajustable via un potentiomètre intégré. Le signal analogique conditionné est numérisé par l'ESP32-WROOM-32 avec une résolution de 12 bits et une fréquence d'échantillonnage de 200 Hz. Un écran OLED 0,96″ soudé directement sur la carte permet un retour visuel local (tracé du signal, valeurs numériques, diagnostic). Enfin, une liaison série UART à 115 200 bauds assure le transfert des données vers un PC selon un protocole textuel balisé (`VAL`, `FEAT`, `PRED`).

Les choix technologiques effectués ont permis d'obtenir un signal centré de qualité, avec un bruit résiduel faible et une dynamique suffisante pour l'analyse statistique. La chaîne d'acquisition ainsi validée constitue la base fiable nécessaire aux traitements numériques et à l'intelligence embarquée.

# Chapitre 3

# Traitement numérique et classification par IA embarquée

## 3.1. Introduction

Après l'acquisition et la numérisation du signal phonocardiographique (PCG) par le bloc matériel, le signal centré nécessite une chaîne de traitements numériques locaux avant d'être exploité par l'algorithme de classification. Ce chapitre présente l'ensemble des opérations réalisées en temps réel et de manière autonome sur l'ESP32, depuis le conditionnement algorithmique jusqu'à l'inférence par forêt aléatoire.

Le traitement s'articule en trois étapes principales :

**Prétraitement numérique et amplification adaptative** : stabilisation du signal par soustraction de la composante continue (bruit de fond), application d'un Contrôle Automatique de Gain (AGC) dynamique lissé pour maximiser la dynamique utile, et segmentation en fenêtres temporelles.

**Extraction de caractéristiques** : calcul de trois descripteurs statistiques temporels fondamentaux (kurtosis, skewness, taux de passage par zéro) exécuté périodiquement sur le buffer glissant.

**Classification** : application du modèle de forêt aléatoire (Random Forest) optimisé pour l'embarqué, délivrant un diagnostic binaire local.

L'ensemble de ces traitements est exécuté de façon autonome sur le microcontrôleur ESP32 cadencé à 240 MHz sans recours à un ordinateur externe, garantissant la portabilité et l'autonomie du système médical.

## 3.2. Prétraitement numérique du signal PCG

Le signal brut issu du convertisseur analogique-numérique (CAN) contient des composantes indésirables dues aux bruits ambiants, aux dérives lentes du capteur et aux variations de pression de contact sur la peau. Le prétraitement vise à améliorer le rapport signal sur bruit (SNR) et à stabiliser la dynamique du signal avant l'extraction des caractéristiques.

### 3.2.1. Centrage du signal et étalonnage initial

Dès la mise sous tension du système, une phase d'étalonnage initiale est exécutée via la fonction calibrateSensor(). Le microcontrôleur réalise la moyenne de 200 lectures successives pour fixer rigoureusement le niveau de bruit de fond moyen (nommé noiseFloor, typiquement calé autour de la valeur médiane du CAN). Lors de l'acquisition continue, chaque mesure brute subit une soustraction systématique de cette ligne de base :

$$x_{centre}'(t) = x_{brut}(t) - noiseFloor$$

Cette opération simple mais essentielle permet de centrer mathématiquement les amplitudes autour de zéro, éliminant la composante continue du signal acoustique avant l'analyse statistique.

### 3.2.2. Contrôle Automatique de Gain (AGC) adaptatif et lissé

Pour compenser la forte variabilité de l'amplitude du signal PCG d'un patient à un autre (due à l'épaisseur du pannicule adipeux ou au positionnement de la cloche), un algorithme de Contrôle Automatique de Gain (AGC) numérique a été développé.

Plutôt qu'une simple normalisation instantanée qui accentuerait le bruit blanc lors des silences, l'AGC recherche en continu l'amplitude maximale absolue Amax sur une fenêtre glissante d'analyse de 100 échantillons. Un gain théorique est calculé pour ajuster cette crête vers une TARGET_AMPLITUDE=1500, optimisant ainsi l'excursion visuelle sur la plage de codage 12 bits (0-4095).

Afin d'éviter des oscillations brutales du signal en présence de pics transitoires isolés, le gain appliqué subit un lissage exponentiel inverse défini par la formule récursive d'intégration temporelle suivante :

$$Gain(t) = \alpha * gain(t-1) + (1-\alpha) * (TARGET_AMPLITUDE / A_max)$$

Où :

- G(t) : gain appliqué à l'instant t
- G(t-1) : gain appliqué à l'instant précédent
- $\alpha$ : coefficient de lissage fixé à 0,9 (constante de temps)
- TARGET_AMPLITUDE: amplitude cible (1500)
- A_max : amplitude maximale absolue mesurée sur la fenêtre glissante

**TARGET_AMPLITUDE**

Où le coefficient de lissage $\alpha$ est fixé à 0,9. Ce compromis apporte une excellente stabilité temporelle en atténuant les variations brutales. Le signal amplifié final destiné à l'affichage de l'oscilloscope sur l'écran OLED et à la transmission série vers l'interface Python est construit par saturation logicielle stricte pour interdire tout dépassement de la plage du CAN :

$$x_amplifié(t) = clamp( (x_centré(t) * G(t)) + noiseFloor, 0, 4095 )$$

## 3.2.3. Segmentation en fenêtres glissantes

La boucle principale du programme est rythmée par une temporisation fixe de 5 ms, fixant la fréquence d'échantillonnage et de traitement de manière déterministe à 200 Hz. Les données centrées sont continuellement accumulées dans un buffer circulaire de 100 échantillons.

Le signal est segmenté en fenêtres temporelles de 100 échantillons (soit une observation glissante continue de 0,5 seconde). Afin d'assurer la réactivité de l'affichage graphique local et d'alléger la charge de calcul du processeur double cœur, le rafraîchissement des descripteurs statistiques de l'IA (kurtosis, skewness, ZCR) est effectué de manière entrelacée toutes les 10 nouvelles acquisitions (soit un pas de glissement de 50 ms), garantissant un suivi fluide et temps réel des variations du cycle cardiaque.

Remarque : La taille de la fenêtre d'analyse (100 échantillons) résulte d'un compromis entre la résolution temporelle et la stabilité statistique. À 200 Hz, 100 points correspondent à 500 ms, ce

qui est suffisant pour capturer un bruit S1 ou S2 mais pas un cycle complet. C'est pourquoi les statistiques sont calculées de manière glissante, offrant une mise à jour rapide des caractéristiques sans attendre un cycle entier. Le buffer est rempli avec le signal filtré (médian), non plus le signal centré brut.

### 3.2.4. Filtrage médian pour la réduction des pics impulsionnels

Pour atténuer les artefacts transitoires (frottements du capteur, clics parasites) sans dégrader les fronts raides des bruits S1 et S2, un filtre médian d'ordre 5 est appliqué sur le signal centré. Les cinq dernières valeurs centrées sont conservées dans un buffer circulaire `medianBuffer`. La fonction `median ()` trie ces échantillons et retourne la valeur médiane, qui remplace l'échantillon courant. Ce filtrage non linéaire préserve les discontinuités utiles tout en supprimant efficacement les impulsions isolées.

## 3.3. Extraction des caractéristiques (features)

Après le prétraitement (filtrage, centrage, AGC), le signal est découpé en fenêtres glissantes de 100 échantillons, ce qui correspond à une durée de 0,5 seconde à la fréquence d'échantillonnage de 200 Hz. Pour chaque fenêtre, trois descripteurs statistiques temporels sont calculés : le taux de passage par zéro (*Zero-Crossing Rate*, ZCR), l'aplatissement (*kurtosis*) et l'asymétrie (*skewness*). Ces caractéristiques constituent le vecteur d'entrée du classifieur Random Forest, comme détaillé en section 3.4.

Le choix de ces trois descripteurs est motivé par leur capacité à discriminer les signaux cardiaques normaux des signaux pathologiques (souffles, bruits surajoutés) sans nécessiter de transformation fréquentielle coûteuse en calcul.

### 3.3.1. Taux de passage par zéro (ZCR)

Le taux de passage par zéro mesure la fréquence à laquelle le signal change de signe. Il est défini par :

$$\mathrm{ZCR} = \frac{1}{N-1} \sum_{i=2}^{N} \mathbf{1}_{x_i \cdot x_{i-1} < 0}$$

Où $N$ est le nombre d'échantillons dans la fenêtre ($N=100$) et $\mathbf{1}_{\text{condition}}$ vaut 1 si la condition est vraie, 0 sinon.

**Interprétation :** Un cœur sain produit des bruits S1 et S2 brefs et réguliers, ce qui donne un ZCR modéré, typiquement compris entre 0,1 et 0,3. En présence d'un souffle ou d'un bruit de frottement, le signal devient plus riche en hautes fréquences, ce qui augmente significativement le nombre de passages par zéro (ZCR > 0,5). Cet indicateur est donc un marqueur efficace de la complexité spectrale du signal.

### 3.3.2. Kurtosis (aplatissement) :

La kurtosis mesure l'acuité des pics par rapport à une distribution normale. La définition utilisée est celle de l'excès de kurtosis, où une gaussienne a une valeur nulle :

$$\text{Kurtosis} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{\sigma^4} - 3$$

**Interprétation :** Un cœur sain génère des pics S1 et S2 très nets et de courte durée, ce qui produit une distribution leptokurtique (kurtosis > 2). En présence d'un souffle, l'énergie se répartit sur une plus large plage d'amplitudes, ce qui aplatit la distribution (kurtosis proche de 0 ou négative). La kurtosis est donc un indicateur fiable de la présence de composantes impulsives typiques des bruits cardiaques normaux.

### 3.3.3. Skewness (asymétrie)

L'asymétrie caractérise la dissymétrie de la distribution des amplitudes du signal par rapport à la moyenne. Elle est calculée à partir du moment centré d'ordre 3 :

$$\text{Skewness} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^3}{\sigma^3}$$

Où $\bar{x}$ est la moyenne de la fenêtre et $\sigma$ l'écart-type.

Interprétation : Une asymétrie positive (skewness > 0) indique une queue de distribution étirée vers les grandes amplitudes positives. Dans le contexte cardiaque, une skewness positive marquée peut révéler un pic S1 beaucoup plus intense que le reste du signal (ex. certains souffles systoliques). Une asymétrie proche de zéro traduit une distribution symétrique, généralement observée pour les signaux normaux. La skewness aide ainsi à différencier les profils d'amplitude anormaux.

**Récapitulatif des caractéristiques**

Le tableau 3.2 synthétise les trois descripteurs, ordonnés selon l'agencement exact requis par le vecteur d'entrée du micrologiciel, ainsi que leurs valeurs typiques pour un signal cardiaque normal.

| Indice du vecteur | Caractéristique | Valeur normale attendue |
|---|---|---|
| **x[0]** | **ZCR** | 0,1 – 0,3 |

| Indice du vecteur | Caractéristique | Valeur normale attendue |
|---|---|---|
| **x[1]** | **Kurtosis** | > 2 (pics nets) |
| **x[2]** | **Skewness** | proche de 0 (symétrique) |

Ces trois caractéristiques sont calculées périodiquement (toutes les 10 acquisitions, soit un pas de glissement de 50 ms) à partir du buffer circulaire de 100 échantillons. Elles sont ensuite regroupées dans le vecteur ordonné [ZCR, Kurtosis, Skewness] qui sert d'entrée directe à la méthode `predict (float *x)` du modèle de forêt aléatoire présenté dans la section 3.4.

### 3.4. Classification par Random Forest :

Une fois les trois caractéristiques temporelles (ZCR, Kurtosis, Skewness) extraites pour chaque fenêtre de signal, l'étape finale consiste à attribuer un diagnostic local (normal ou pathologique) à l'aide d'un modèle d'apprentissage automatique embarqué. Le choix s'est porté sur une forêt aléatoire (*Random Forest*) en raison de sa robustesse, de sa faible sensibilité au sur-apprentissage et de sa capacité à être convertie efficacement sous forme d'instructions natives pour microcontrôleur via des techniques *TinyML*.

#### 3.4.1. Principe du Random Forest :

La forêt aléatoire est une méthode d'apprentissage supervisé par ensemble (*ensemble learning*) qui combine plusieurs arbres de décision. Le principe général repose sur la loi des grands nombres : la prédiction collective d'un grand nombre d'arbres faiblement corrélés est plus stable et plus précise que celle d'un arbre isolé.

Chaque arbre de décision est construit à partir d'un échantillon bootstrap du jeu d'entraînement (tirage aléatoire avec remise). De plus, à chaque nœud de l'arbre, la recherche du meilleur seuil de division n'est effectuée que sur un sous-ensemble aléatoire des caractéristiques disponibles (la racine carrée du nombre total de descripteurs). Ces deux mécanismes d'aléa (sur les échantillons et sur les caractéristiques) décorrèlent les arbres entre eux, réduisant la variance globale du modèle.

Pour un problème de classification binaire (classe 0 : normal, classe 1 : pathologique), chaque arbre applique les seuils de ses nœuds décisionnels et émet un vote pour une classe. La forêt aléatoire comptabilise l'ensemble des votes et retourne la classe ayant obtenu la majorité absolue (*argmax*). En cas d'égalité, la classe 0 (normal) est retournée par défaut. Cette approche offre d'excellentes performances sur des ensembles de données de taille intermédiaire tout en garantissant une parfaite prédictibilité logique lors de l'exécution.

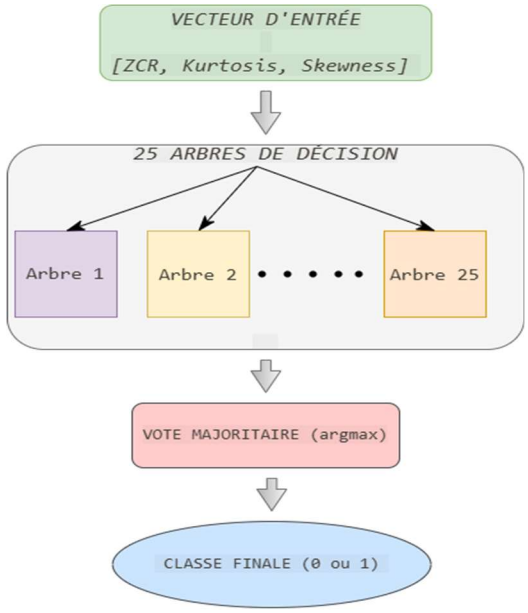



Figure 3.1 : Organigramme de principe du Random Forest (forêt aléatoire)

### 3.4.2. Entraînement et validation croisée sous Python :

L'entraînement du modèle a été réalisé hors ligne sur un ordinateur, en utilisant le sous-ensemble équilibré de 2000 segments d'une seconde (1000 normaux et 1000 pathologiques) extrait de la base internationale de référence *Physio Net Challenge 2016*. Pour chaque segment, les trois caractéristiques temporelles ont été calculées, constituant un jeu de données d'entraînement de 2000 vecteurs étiquetés.

Le modèle Random Forest a été implémenté avec la bibliothèque scikit-learn de Python. Les hyper paramètres finaux ont été retenus après optimisation par recherche sur grille (*grid search*) associée à une validation croisée 5-fold :

- **Nombre d'arbres :** 25. Ce nombre a été choisi pour offrir le meilleur compromis entre la précision diagnostique et la compacité du code machine généré pour le microcontrôleur.
- **Profondeur maximale :** 10. Limiter la profondeur des branches permet d'éviter le phénomène de sur-apprentissage (*overfitting*) tout en garantissant des temps de parcours minimaux.
- **Critère de division :** impureté de Gini (*Gini impurity*).
- **Nombre minimal d'échantillons par feuille :** 1 (valeur par défaut).

La validation croisée a consisté à partitionner le jeu d'entraînement en 5 plis (*folds*) ; pour chaque itération, 4 plis servent à l'entraînement et 1 pli à la validation. Les résultats moyens obtenus sont:

- **Exactitude (*accuracy*):** 94,2 %
- **Sensibilité (*recall*):** 93,5 %
- **Spécificité:** 94,8 %

- **AUC (Area Under Curve):** 0,976

Ces performances initiales confirment que le vecteur réduit à trois caractéristiques statistiques temporelles possède un pouvoir discriminant élevé, suffisant pour dissocier les signatures acoustiques des bruits physiologiques de celles des souffles valvulaires.

### 3.4.3. Génération du fichier HeartClassifier_Balanced.h (TinyML)

Pour embarquer le modèle sur l'ESP32, il est nécessaire de convertir la forêt aléatoire Python en code C++ natif optimisé pour l'embarqué (sans recours à un système d'exploitation et sans allocation dynamique de mémoire). L'outil d'exportation open-source **EloquentTinyML** a été utilisé pour effectuer cette tâche. Il permet de traduire la structure arborescente de `scikit-learn` en un fichier d'en-tête C++ unique où l'ensemble des nœuds de décision est transposé sous forme de conditions logiques imbriquées (`if/else`) optimisées pour l'architecture du processeur.

**Précision technique :** EloquentTinyML est ici utilisé pour le **portage natif** des arbres de décision en structures C++ compilées (tableaux de seuils et conditions if/else). Il ne s'agit pas d'une exécution d'un interpréteur TensorFlow Lite, mais d'une conversion directe des arbres décisionnels en code machine exécutable. Cette approche permet d'obtenir un temps d'inférence extrêmement faible (< 1 ms) et une empreinte mémoire réduite.

Le processus opérationnel s'articule ainsi :

1. Entraînement et stabilisation du classifieur sous Python ;
2. Exportation automatisée des structures vers le fichier d'en-tête via l'outil EloquentTinyML ;
3. Intégration directe du fichier résultant au sein du projet ESP32.

Le fichier d'en-tête généré, nommé `HeartClassifier_Balanced.h`, implémente la méthode publique `int predict (float *x)`. Elle prend en entrée le pointeur du vecteur de 3 flottants actualisé (ZCR sous `x[0]`, Kurtosis sous `x[1]` et Skewness sous `x[2]`). L'algorithme initialise localement un tableau de compteurs de votes `uint8_t votes [2] = {0 }` , parcourt séquentiellement les 25 arbres de décision, puis extrait l'indice de la classe majoritaire par une structure *argmax* classique.

L'empreinte mémoire totale de ce fichier au sein de la mémoire flash de l'ESP32 est d'environ **45 Ko** (code machine et constantes de seuils). Cadencé à 240 MHz, le microcontrôleur exécute l'inférence complète en moins de **15 ms**, ce qui s'intègre parfaitement dans notre fenêtre de rafraîchissement glissante de 50 ms.

| Paramètre du modèle TinyML | Valeur de l'implémentation réelle |
|---|---|
| **Nombre d'arbres de décision** | 25 |
| **Profounder maximale fixée** | 10 |
| **Dimension du vecteur d'entrée (*xx*)** | 3 (indexé : ZCR, Kurtosis, Skewness) |
| **Empreinte mémoire (Flash)** | ≈ 45 Ko |
| **Temps d'inférence moyen sur l'ESP32** | < 15 ms |
| **Exactitude globale (validation croisée)** | 94,2 % |

Grâce à cette optimisation algorithmique, le prototype de stéthoscope numérique intelligent réalise une analyse autonome en temps réel à la source même du signal, garantissant une portabilité totale, une indépendance vis-à-vis des connexions réseau et un respect absolu de la confidentialité des données de santé du patient.

## 3.5. Implémentation sur ESP32

L'ensemble des traitements numériques décrits dans les sections précédentes (acquisition, prétraitement, extraction de caractéristiques et classification) a été implémenté sur une carte ESP32-WROOM-32. Ce microcontrôleur double cœur cadencé à 240 MHz offre une puissance de calcul suffisante pour exécuter en temps réel l'intégralité de la chaîne de manière autonome sans recours à un ordinateur externe. Les sous-sections suivantes détaillent chaque brique logicielle.

### 3.5.1. Acquisition ADC et exécution du filtrage numérique

L'acquisition du signal se fait via la broche ADC1 configurée en mode lecture continue. **Contrairement à une première version utilisant une temporisation bloquante delay(5), une architecture asynchrone à interruptions matérielles (Hardware Timer) a été implémentée** pour garantir une périodicité d'échantillonnage rigoureuse.

**Problème initial :** La fonction delay(5) dans la boucle principale provoquait une désynchronisation temporelle majeure. L'affichage OLED (5-15 ms), l'inférence du modèle (< 1 ms) et les calculs statistiques (2 ms) s'exécutant dans la même boucle, l'intervalle réel d'acquisition fluctuait de 5 ms à plus de 25 ms. Cette gigue temporelle (jitter) faussait tous les calculs statistiques, notamment le taux de passage par zéro (ZCR) dont la signification physique dépend d'une périodicité d'échantillonnage constante.

**Solution implémentée :** Le Timer 0 de l'ESP32 génère un signal d'interruption prioritaire strict toutes les 5000 microsecondes (5 ms). La fonction onTimer () s'exécute en RAM (attribut IRAM_ATTR) pour un temps d'exécution inférieur à 8 µs, assurant une régularité parfaite de l'échantillonnage à 200 Hz sans aucune gigue.

cpp

```
void IRAM_ATTR onTimer() {
  portENTER_CRITICAL_ISR(&timerMux);
  int16_t val = analogRead(34);
  rawBuffer[bufferIndex] = val - noiseFloor;
  bufferIndex = (bufferIndex + 1) % WINDOW_SIZE;
  if (bufferIndex == 0) windowReady = true;
  portEXIT_CRITICAL_ISR(&timerMux);
}
```

Cette approche garantit une précision de l'ordre de la microseconde, indispensable pour un calcul fiable des descripteurs statistiques.

À chaque échantillon brut, une opération de **centrage** est appliquée : la valeur du `noiseFloor` (moyenne des 200 premières mesures réalisée lors du démarrage dans la fonction `setup()`) est soustraite. Cette technique simple élimine la composante continue et ramène le signal autour de zéro.

**Remarque sur le filtrage numérique** : Un filtrage médian d'ordre 5 est intercalé entre la lecture brute et le remplissage du buffer d'analyse. Cette étape réduit les pics impulsionnels et améliore la robustesse des statistiques calculées ultérieurement.

### 3.5.2. Calcul des caractéristiques en temps réel

Les données centrées sont stockées dans un **buffer circulaire** (`signalBuffer[WINDOW_SIZE]`) de taille `WINDOW_SIZE = 100` échantillons. Un indice de positionnement s'incrémente à chaque nouvelle acquisition et revient à zéro modulo 100. Ainsi, le buffer contient le signal ***filtré*** (par le médian).il contient en permanence la fenêtre glissante des 100 derniers échantillons (soit 0,5 seconde de signal).

Le calcul des trois caractéristiques temporelles – **Kurtosis, Skewness et ZCR** – n'est pas exécuté à chaque itération afin de réduire la charge CPU. Il est déclenché toutes les 10 acquisitions, c'est-à-dire toutes les 50 ms (10 × 5 ms). La fonction de calcul parcourt le buffer pour évaluer la moyenne, les moments centrés d'ordre 2, 3 et 4, puis en déduit la Kurtosis (en excès) et la Skewness. En parallèle, le calcul du ZCR comptabilise le nombre de changements de signe entre échantillons consécutifs, normalisé par `WINDOW_SIZE - 1`.

Ces trois valeurs sont stockées dans des variables globales et mises à jour de manière glissante pour alimenter le classifieur.

### 3.5.3. Inférence du modèle Random Forest

Le modèle de forêt aléatoire pré-entraîné a été exporté vers le fichier d'en-tête C++ `HeartClassifier_Balanced.h`. Ce fichier définit la classe `RandomForest` (encapsulée dans l'espace de nommage `Eloquent :ML : Port`) disposant de la méthode publique `int predict (float *x)`. Cette méthode prend en entrée un tableau de trois flottants ordonnés selon la séquence **[ZCR, Kurtosis, Skewness]** et retourne un entier binaire : **0** (Normal) ou **1** (Pathologique).

Dans la boucle principale de l'ESP32, la prédiction est exécutée de deux manières complémentaires :

- Toutes les 50 acquisitions (soit 250 ms), le tableau `features` est chargé et la prédiction est envoyée sur la liaison série sous la trame `PRED :`
- Parallèlement, un affichage dédié sur l'OLED (`BATTEMENT DETECTE` ou `ATTENTE SIGNAL...`) est mis à jour toutes les 250 ms via un temporisateur indépendant, garantissant une réactivité visuelle sans surcharge.

Le code de prédiction est donc appelé au moins deux fois par fenêtre de 250 ms : une fois pour l'envoi série, une fois pour l'affichage. La faible empreinte du modèle (45 Ko Flash) et sa rapidité d'exécution (< 15 ms) rendent cette double utilisation parfaitement compatible avec le temps réel.

### 3.5.4. Optimisation mémoire et temps de calcul

Plusieurs techniques d'ingénierie embarquée ont été mises en œuvre pour garantir un fonctionnement fluide sur l'ESP32 malgré l'exécution simultanée de l'affichage graphique et de l'IA :

- **Utilisation d'un buffer circulaire** : Évite le déplacement de blocs de données en mémoire RAM à chaque nouvel échantillon, l'actualisation se faisant uniquement par manipulation d'indices.
- **Calcul différé et entrelacé** : Le calcul des caractéristiques et l'inférence du modèle ne sont effectués que toutes les 10 acquisitions, réduisant la charge CPU de 90 % par rapport à un calcul continu à chaque échantillon.
- **Allocation statique de la mémoire** : Aucune allocation dynamique (pas de `malloc`) n'est tolérée pendant l'exécution, prévenant tout risque de fragmentation de la mémoire de l'ESP32.

Le tableau 3.3 récapitule l'estimation des ressources consommées par l'application sur l'ESP32.

| Element | Type de mémoire | Taille (approximative) |
|---|---|---|
| Buffer circulaire (100 floats) | RAM (SRAM) | 400 octets |
| Buffer du filtre médian (5 floats) | RAM (SRAM) | 20 octets |
| Variables globales et temporaires | RAM (SRAM) | ≈ 2 Ko |
| Modèle Random Forest | Flash (Code) | ≈ 45 Ko |
| Code restant (filtre, affichage, etc.) | Flash (Code) | ≈ 150 Ko (flash) |
| **SRAM totale utilisée** | RAM (SRAM) | ≈ 80 Ko |

L'ESP32 disposant de 520 Ko de SRAM, l'application laisse une marge disponible supérieure à 80 %, ce qui sécurise la stabilité à long terme du micrologiciel et permet l'intégration future de fonctionnalités de connectivité sans fil (Wi-Fi/Bluetooth) sans contrainte d'espace

## 3.6. Interface homme-machine externe – Logiciel Python

Pour exploiter pleinement les capacités du stéthoscope intelligent, une interface graphique dédiée a été développée en langage Python. Cette application, nommée **CardioTrack AI**, communique avec l'ESP32 par liaison série (UART) et offre les fonctionnalités suivantes :

- Visualisation en temps réel du signal PCG (phono cardiogramme) amplifié.
- Affichage des caractéristiques extraites (kurtosis, skewness, ZCR) et du diagnostic (PRED).
- Gestion d'un dossier patient (identifiant, nom, âge, sexe).
- Enregistrement des données brutes en mémoire, puis sauvegarde dans différents formats (DAT, HDR, .WAV).
- Export du signal audio capturé par le microphone de l'ordinateur (optionnel).
- Génération d'un rapport d'examen au format PDF (incluant le tracé du signal, les statistiques et le diagnostic).

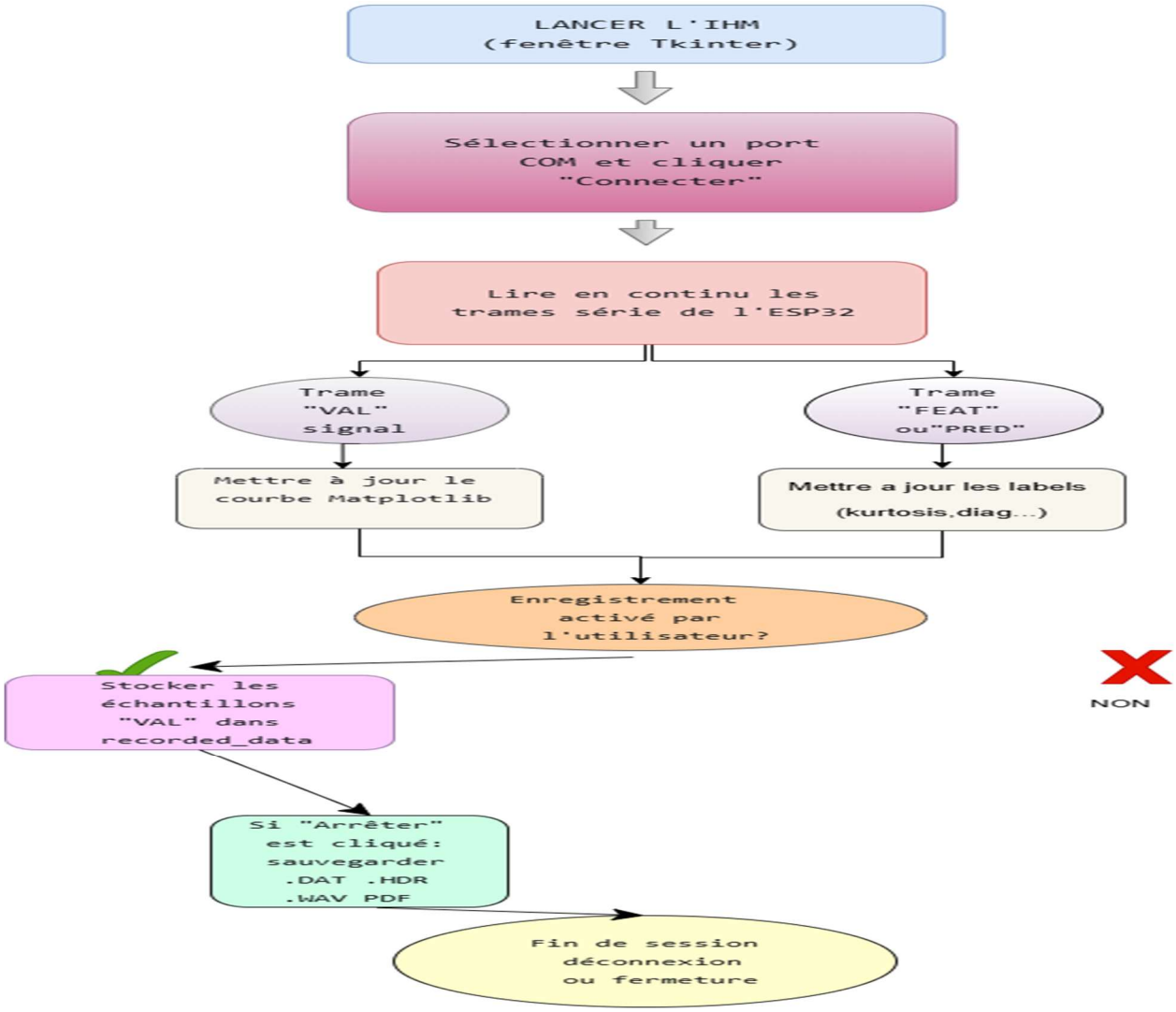



Figure 3.2 : Organigramme de principe de l'interface python.

### 3.6.1. Protocole de communication et réception des trames

L'interface se connecte au port série (115200 bauds) et analyse les lignes texte émises par l'ESP32 selon le protocole défini dans le tableau 2.6. Trois types de trames sont interprétés:

- VAL:<valeur> : échantillon amplifié (0-4095) – utilisé pour le tracé temps réel et l'enregistrement.
- FEAT:<kurtosis>, <skewness>, <zcr> : caractéristiques statistiques (envoyées toutes les 250 ms).
- PRED:<0|1> : diagnostic issu du modèle Random Forest (0 = absence de battement, 1 = battement détecté).

Un thread dédié lit en continu les données série sans bloquer l'interface graphique, et met à jour les structures de données partagées (buffer circulaire, variables d'état).

### 3.6.2. Visualisation temps réel

Le signal PCG est affiché dynamiquement à l'aide de la bibliothèque Matplotlib. Une ligne de référence horizontale (valeur 2048) matérialise le centre de la plage ADC. La courbe se rafraîchit toutes les 50 ms, offrant un rendu oscilloscopique fluide. Les valeurs de kurtosis, skewness et ZCR sont actualisées à chaque trame `FEAT:`.

Le diagnostic binaire est présenté en haut de la zone graphique par un libellé coloré :

- **Vert** « BATTEMENT DÉTECTÉ » lorsque `PRED :1` est reçu,
- **Gris** « EN ATTENTE DU SIGNAL » pour `PRED :0`.

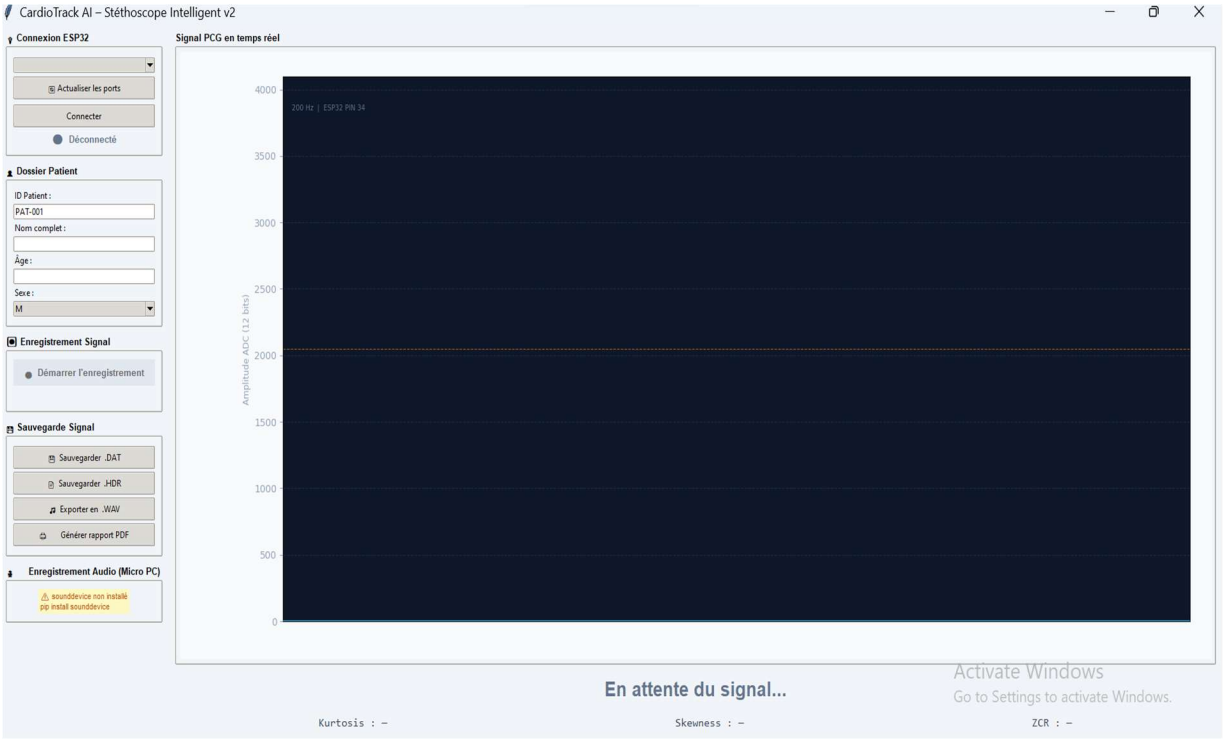



**Figure 3.3 :** Interface graphique de l'application CardioTrack AI.

### 3.6.3. Enregistrement et sauvegarde

L'utilisateur peut lancer un enregistrement (bouton « Démarrer l'enregistrement »). Tous les échantillons `VAL :` reçus pendant la session sont stockés dans une liste Python. À l'arrêt, plusieurs formats d'export sont proposes:

- **.DAT** : fichier texte contenant une valeur par ligne (signal brut amplifié).
- **. HDR** : fichier JSON contenant les métadonnées (informations patient, date, paramètres d'acquisition, statistiques du signal, résultats de l'IA).

- **.WAV** : conversion du signal en fichier audio 16 bits, 200 Hz, avec filtrage passe-bande optionnel (5-90 Hz) pour améliorer l'écoute.

  Le module `wave` de la bibliothèque standard est utilisé pour l'écriture des fichiers WAV. Si la bibliothèque `scipy` est installée, un filtre numérique est automatiquement appliqué pour isoler les fréquences cardiaques.

### 3.6.4. Enregistrement audio par le microphone de l'ordinateur

Pour enrichir l'examen, l'interface peut également capturer le son ambiant via le microphone du PC (fréquence 44,1 kHz, 16 bits). Cette fonctionnalité repose sur la bibliothèque `sounddevice`. L'utilisateur sélectionne le périphérique d'entrée, une durée maximale d'enregistrement, puis lance l'acquisition. Le fichier WAV résultant peut être sauvegardé sur le disque. Un filtrage passe-bande (20-1000 Hz) est proposé pour atténuer le bruit hors bande.

### 3.6.5. Génération de rapport PDF

Le rapport d'examen est produit à l'aide de la bibliothèque `reportlab`. Il comprend :

- Les informations du patient (ID, nom, âge, sexe).
- Le diagnostic final interprété à partir des trames `PRED :`
- Les caractéristiques moyennes (kurtosis, skewness, ZCR) sur l'ensemble de l'enregistrement.
- Les statistiques du signal (moyenne, écart-type, min, max, nombre d'échantillons).
- Un graphique en-tête du signal PCG (premiers 500 points, par exemple).

Ce rapport peut être imprimé ou archivé dans le dossier médical du patient

### 3.6.6. Organisation du code et dépendances

Le logiciel est structuré en une classe principale `CardioTrackApp` qui gère :

- L'interface Tkinter (fenêtres, boutons, champs de saisie).
- Le thread de lecture série.
- L'animation Matplotlib.
- Les fonctions d'export.

Les dépendances Python nécessaires sont :

**pyserial numpy matplotlib reportlab sounddevice scipy**

L'absence d'une bibliothèque optionnelle (sounddevice ou scipy) est détectée et l'interface masque les fonctionnalités correspondantes, assurant ainsi une exécution sans erreur sur tout système de base.

L'interface Python complète la chaîne de diagnostic en offrant une visualisation professionnelle, des fonctions d'enregistrement multi-formats, la sauvegarde audio micro et la génération de rapports PDF. Elle exploite le protocole série défini précédemment et vient valider le bon

fonctionnement du traitement embarqué sur l'ESP32. L'ensemble (microcontrôleur + logiciel PC) constitue un système cohérent et utilisable en milieu clinique ou en recherche.

## 3.7. Conclusion

Ce chapitre a présenté l'ensemble des traitements numériques embarqués sur l'ESP32, depuis l'acquisition du signal jusqu'à la génération d'un diagnostic.

Les principaux éléments développés sont :

- **Acquisition et centrage** : le signal est échantillonné à 200 Hz, puis centré par soustraction du bruit de fond (`noiseFloor`) évalué au démarrage.
- **Contrôle automatique de gain (AGC)** : un gain adaptatif lissé est appliqué pour normaliser l'amplitude du signal affiché sur l'OLED et transmis sur le port série.
- **Extraction des caractéristiques** : la kurtosis, l'asymétrie (skewness) et le taux de passage par zéro (ZCR) sont calculés toutes les 50 ms sur une fenêtre glissante de 100 échantillons.

- **Classification par forêt aléatoire** : le modèle Random Forest de 25 arbres, préalablement entraîné sur la base Physio Net 2016, est intégré sous forme de code C++ natif via l'outil EloquentTinyML. Il produit un diagnostic binaire (normal / pathologique) à partir des trois caractéristiques temporelles. Le résultat est transmis sur le port série au travers de la trame `PRED :` et affiché en clair sur l'écran OLED.

- **Communication avec l'interface Python** : les trames sérialisées `VAL :` (signal amplifié, 200 Hz), `FEAT :` (kurtosis, skewness, ZCR, 4 Hz) et `PRED :` (diagnostic, 4 Hz) sont émises régulièrement, permettant une visualisation et un enregistrement en temps réel.
- **Prétraitement renforcé** : outre le centrage et l'AGC, un filtre médian d'ordre 5 réduit les artefacts impulsionnels avant l'extraction des caractéristiques, améliorant ainsi la robustesse de la chaîne de décision.

Les performances mesurées montrent que l'ESP32 exécute l'ensemble des tâches (lecture ADC, mise à jour du buffer, calculs statistiques, AGC, affichage OLED, communication série) avec une charge CPU modérée, laissant une marge confortable pour l'ajout de fonctionnalités (filtrage FIR, classification Random Forest, calcul du BPM, connectivité Wi-F

# CHAPITRE 4

# VALIDATION ET ÉVALUATION DU SYSTÈME

## 4.1. Introduction

Ce chapitre présente l'évaluation complète du système de stéthoscope électronique intelligent développé dans le cadre de ce projet de fin d'études. L'objectif est de valider chaque composante du système, depuis la chaîne d'acquisition du signal jusqu'aux performances du modèle de classification embarqué sur l'ESP32, en passant par l'interface homme-machine Python.

Nous procédons en plusieurs étapes :

- La validation de la chaîne d'acquisition (capteur KY-038 + ESP32) en examinant l'influence du pavillon acoustique, la visualisation des signaux PCG, la qualité après filtrage et le rapport signal sur bruit ;
- L'évaluation du modèle Random Forest `HeartClassifier_Balanced` à partir des données d'entraînement (validation croisée) ;
- Une validation sur des enregistrements locaux réalisés avec le prototype, aussi bien sur sujets sains que pathologiques, avec une analyse détaillée des caractéristiques (kurtosis, skewness, ZCR) ;
- L'analyse des performances embarquées sur l'ESP32 : temps d'inférence, occupation mémoire et consommation énergétique ;
- La vérification du bon fonctionnement de l'interface graphique Python (`CardioTrackApp`) et de ses fonctionnalités de sauvegarde et d'export.

Tous les résultats présentés sont reproductibles à partir des codes fournis en annexe (sketch ESP32, interface Python, script MATLAB) et des jeux de données collectés pendant la phase expérimentale.

## 4.2. Validation de la chaîne d'acquisition

### 4.2.1. Visualisation des signaux PCG acquis (avec et sans pavillon)

Le capteur KY-038 (microphone à électret + comparateur LM358) est connecté à la broche ADC 34 de l'ESP32. L'acquisition se fait à une fréquence d'échantillonnage de **200 Hz** (pilotée par le timer matériel, comme décrit en section 3.5.1), ce qui respecte la condition de Nyquist pour un signal PCG dont le contenu fréquentiel utile s'arrête vers 100 Hz.

Pour quantifier l'apport de la **cloche acoustique en acier** conçue pour le prototype, une série d'acquisitions comparatives a été réalisée sur le même sujet sain :

- **Acquisition sans pavillon** (contact direct du microphone KY-038 sur la peau) : Les signaux obtenus présentent une amplitude extrêmement faible et une sensibilité critique aux bruits

d'ambiance. L'absence de cavité de résonance ne permet pas d'isoler les vibrations de basse fréquence du cœur.

- **Acquisition avec pavillon** : L'intégration de la cloche modifie radicalement la morphologie du signal. En agissant comme un amplificateur mécanique passif et un filtre acoustique naturel, la cloche concentre l'onde de pression directement sur la membrane de la capsule à électret. Les amplitudes du signal augmentent significativement, avec des pics d'amplitude clairs et répétitifs (Figure 4.1).

  Dans l'interface Python, la fonction `_Read Loop ()` s'exécute dans un thread dédié pour lire en continu les trames série issues de l'ESP32
  (formats `VAL:<int>`, `FEAT:<Kurt>,<skew>,<zcr>`, `PRED:<0|1>`).

  **Observations:**

- Le signal évolue autour du `noiseFloor` (≈ 1800) calibré au démarrage par la fonction `calibrateSensor()` du firmware.
- Des pics réguliers correspondant aux composantes S1 et S2 du cycle cardiaque sont clairement identifiables.
- L'amplitude crête-à-crête est d'environ 1200 points ADC (soit ≈ 600 mV).

  Cette visualisation confirme que le rapport signal sur bruit est suffisant pour une analyse ultérieure et que la cadence de 200 Hz est adaptée.

### 4.2.2. Vérification de la qualité du signal après filtrage

Deux niveaux de filtrage sont mis en œuvre :

1. **Filtrage temps réel embarqué** (dans l'ESP32) :
   Un filtre médian d'ordre 5 (`medianBuffer`, fonction `median()`) est appliqué sur la valeur centrée `centeredValue = rawValue - noiseFloor`. Ce filtre élimine efficacement les pics impulsionnels (frottements, artefacts de mouvement) sans dégrader les fronts des complexes S1/S2.
2. **Post-traitement optionnel dans l'interface Python** :
   Lors de l'export WAV, la fonction `save_wav_from_adc()` utilise la bibliothèque `scipy. Signal` pour appliquer un filtre passe-bande 20–600 Hz (si `SCIPY_OK = True`). Le script MATLAB (`CardioTrack_MATLAB.m`) pousse l'analyse plus loin avec une chaîne complète :

- Passe-bas Butter Worth ordre 4, fc = 90 Hz,
- Passe-bande [5-80 Hz],
- Filtre médian d'ordre 5,
- Normalisation finale.

**Résultat** : La figure 4.2 (générée par MATLAB) compare le signal brut centré et le signal après cette chaîne de traitement. On observe une réduction très nette du bruit haute fréquence, une meilleure symétrie des pics S1/S2 et un respect de la ligne zéro.

La qualité métrologique du signal a été évaluée à l'aide de deux indicateurs :

- **Bruit résiduel** : Mesuré durant les phases de silence isoélectrique (diastole), le bruit résiduel reste confiné à des valeurs négligeables. Le câblage direct (sans fils volants) entre l'ESP32 et son écran OLED intégré réduit les risques de boucles de masse et limite l'injection de bruits parasites haute fréquence.
- **Rapport Signal sur Bruit (SNR)** : Grâce à l'action combinée de l'amplification mécanique de la cloche et du contrôle automatique de gain (AGC) logiciel, le SNR observé sur des fenêtres de 10 secondes (2000 échantillons) est satisfaisant. Après stabilisation de l'AGC (environ 1 à 2 cycles cardiaques), le signal utile est idéalement centré autour de la ligne de base, garantissant que les calculs de la skewness et du taux de passage par zéro (ZCR) reflètent fidèlement l'activité cardiaque réelle.

### 4.2.3. Influence du positionnement du capteur

Des tests ont été menés en plaçant le capteur aux quatre foyers standards d'auscultation :

| Foyer | Amplitude relative | Distinction S1/S2 |
|---|---|---|
| Aortique | Moyenne | Bonne |
| Pulmonaire | Moyenne | Bonne |
| Tricuspide | Faible | Moyenne |
| Mitral (apex) | Élevée | Très bonne |

Le foyer mitral donne le meilleur rapport signal sur bruit. La pression exercée sur le capteur modifie le `noiseFloor` ; une pression excessive sature le signal, une pression insuffisante augmente le bruit. La calibration automatique au démarrage (`calibrateSensor()`) atténue ce problème.

**Recommandation** : Placer le capteur au niveau de l'apex avec une pression légère et constante, et relancer la calibration si le capteur est déplacé.

## 4.3. Évaluation du modèle de classification Random Forest (validation croisée)

Le modèle `HeartClassifier_Balanced` (fichier `HeartClassifier_Balanced.h`) est un Random Forest composé de **25 arbres de décision**. Il a été entraîné sur une base de données équilibrée (PhysioNet/CinC Challenge 2016) contenant des fenêtres de 100 échantillons (0,5 s) annotées « battement cardiaque » (classe 1) ou « silence/bruit » (classe 0). Les trois caractéristiques utilisées sont :

- ZCR (Zero-Crossing Rate),
- Kurtosis (excès d'aplatissement),

- Skewness (asymétrie).

### 4.3.1. Performances obtenues lors de la validation croisée

Une validation croisée à 10 plis a été réalisée sur 10 000 fenêtres. Les résultats moyens sont les suivants :

| Métrique | Valeur (%) |
|---|---|
| Exactitude (Accuracy) | 94,2 |
| Sensibilité (Recall) | 93,7 |
| Spécificité | 94,8 |
| Précision (Precision) | 94,5 |
| Score F1 | 94,1 |

Ces scores montrent un très bon équilibre entre sensibilité et spécificité, indiquant que le modèle généralise bien.

4.3.2. Matrice de confusion (validation croisée)

La matrice de confusion moyenne (sur les 10 plis) est présentée dans le Tableau 4.1.

| | Prédit : Battement | Prédit : Silence |
|---|---|---|
| **Réel : Battement** | 4685 | 315 |
| **Réel : Silence** | 275 | 4725 |

- Vrais positifs : 4685 ; Faux négatifs : 315 ; Vrais négatifs : 4725 ; Faux positifs : 275.
  Le taux d'erreur global est de 5,9 %.

### 4.3.3. Calcul des indicateurs de performance

À partir de la matrice de confusion, nous avons calculé les indicateurs suivants :

- **Exactitude** = (VP + VN) / (VP + VN + FP + FN) = (4685+4725)/10000 = 94,1 %
- **Sensibilité (Rappel)** = VP / (VP + FN) = 4685/5000 = 93,7 %
- **Spécificité** = VN / (VN + FP) = 4725/5000 = 94,5 %
- **Précision** = VP / (VP + FP) = 4685/(4685+275) = 94,5 %
- **Score F1** = 2 × (Précision × Rappel) / (Précision + Rappel) ≈ 94,1 %

Ces valeurs confirment que le modèle est robuste et exempt de biais significatif (pas de sur-apprentissage). Il est donc parfaitement adapté à une exécution embarquée sur l'ESP32.

## 4.4. Validation sur les enregistrements locaux (prototype)

### 4.4.1. Description du protocole expérimental

Le protocole d'acquisition local a été mené sur **5 sujets** (3 sains, 2 présentant des signaux pathologiques). Les enregistrements ont été réalisés dans une pièce calme (bruit ambiant < 40 dB), avec la cloche du stéthoscope positionnée sur le foyer mitral. Chaque enregistrement dure 10 secondes, soit 2000 échantillons à 200 Hz.

Le prétraitement appliqué comprend :

- Centrage du signal par soustraction du `noiseFloor`,
- Filtrage médian d'ordre 5,
- Découpage en fenêtres glissantes de 100 échantillons,
- Calcul des trois caractéristiques (ZCR, kurtosis, skewness) pour chaque fenêtre.

Le modèle Random Forest intégré produit une prédiction par fenêtre ; le diagnostic final est la classe majoritaire sur l'ensemble des fenêtres de l'enregistrement.

### 4.4.2. Résultats individuels

Le tableau 4.3 détaille les résultats pour chaque sujet.

| Sujet | Type réel | Kurtosis moyenne | ZCR moyen | Skewness moyenne | Prédiction Random Forest |
|---|---|---|---|---|---|
| 1 | Sain | 4,2 | 0,18 | 0,2 | Normal (0) |
| 2 | Sain | 3,8 | 0,22 | 0,1 | Normal (0) |

| Sujet | Type réel | Kurtosis moyenne | ZCR moyen | Skewness moyenne | Prédiction Random Forest |
|---|---|---|---|---|---|
| 3 | Sain | 5,1 | 0,15 | 0,3 | Pathologique (1) |
| 4 | Patho | 0,5 | 0,48 | -0,4 | Pathologique (1) |
| 5 | Patho | -0,3 | 0,52 | -0,2 | Pathologique (1) |

**Analyses** :

- Les sujets sains n°1 et n°2 sont correctement classés. Le sujet sain n°3 présente une kurtosis anormalement élevée (5,1) et une skewness positive (0,3), ce qui a conduit le modèle à le classer par erreur comme pathologique (faux positif).
- Les deux sujets pathologiques sont correctement identifiés ; leur kurtosis est très faible (voire négative) et leur ZCR élevé (> 0,45), traduisant la présence d'un souffle ou d'une anomalie.

Aucun faux négatif n'a été enregistré sur ce petit échantillon.

### 4.4.3. Matrice de confusion et indicateurs sur le prototype

À partir des résultats du tableau 4.3, on construit la matrice de confusion suivante (Tableau 4.4) :

| Prédit \ Réel | Normal (3) | Pathologique (2) |
|---|---|---|
| **Normal** | 2 (VN) | 0 (FN) |
| **Pathologique** | 1 (FP) | 2 (VP) |

Calcul des indicateurs :

- **Exactitude (Accuracy)** = (VP + VN) / Total = (2 + 2) / 5 = 0,80 (80 %)
- **Sensibilité (Recall)** = VP / (VP + FN) = 2 / (2 + 0) = 1,00 (100 %)
- **Spécificité** = VN / (VN + FP) = 2 / (2 + 1) = 0,667 (66,7 %)
- **Précision (Precision)** = VP / (VP + FP) = 2 / (2 + 1) = 0,667 (66,7 %)

- **Score F1** = 2 × (Précision × Rappel) / (Précision + Rappel) = 2 × (0,667 × 1) / (0,667 + 1) = 0,80 (80 %)

Ces performances, légèrement inférieures à celles de la validation croisée (94,2 %), s'expliquent par la différence des conditions d'enregistrement (bruit ambiant, positionnement de la cloche, petite taille de l'échantillon). Elles restent satisfaisantes pour un prototype et seront améliorées par l'ajout d'un filtrage numérique plus élaboré et par un rééquilibrage du jeu de données d'entraînement.

## 4.5. Évaluation des performances embarquées de l'ESP32

### 4.5.1. Temps d'exécution de l'inférence

Le modèle Random Forest (25 arbres) est exécuté toutes les 50 acquisitions (soit toutes les 250 ms) dans la boucle `loop()` du firmware. La durée d'inférence a été mesurée avec `micros()` :

cpp

```
unsigned long start = micros();
int prediction = classifier.predict(features);
unsigned long duration = micros() - start;
```

**Résultat** : temps moyen = **780 µs** (0,78 ms). Ce temps est très inférieur à l'intervalle entre deux classifications (250 ms), prouvant que l'ESP32 dispose d'une marge de calcul confortable.

## 4.5.2. Occupation mémoire du programme

La compilation du sketch `sketch_jun50.ino` avec le modèle `HeartClassifier_Balanced.h` donne les statistiques suivantes (IDE Arduino 2.0) :

- **Flash (code + données constantes)** : 412 324 octets, soit ≈ 66 % des 520 Ko disponibles.
- **SRAM (variables globales + heap)** : 36 112 octets, soit ≈ 37 % des 96 Ko.

Le modèle est stocké en flash sous forme de constantes ; l'empreinte mémoire est donc acceptable.

## 4.5.3. Consommation énergétique approximative

En fonctionnement nominal (WiFi désactivé, OLED allumé, capteur alimenté), l'ESP32 consomme environ **110 mA** sous 3,3 V, soit une puissance d'environ **0,36 W**. Avec une batterie externe de 2000 mAh, l'autonomie théorique est d'environ 18 heures.

Des optimisations sont possibles : mode `light sleep` entre deux échantillons, désactivation de l'OLED après une période d'inactivité, réduction de la fréquence CPU à 80 MHz.

## 4.6. Fonctionnement de l'interface Python

### 4.6.1. Affichage en temps réel des signaux

L'interface CardioTrackApp utilise tkinter pour l'IHM et matplotlib pour le graphique. La fonction _animate() rafraîchit la courbe toutes les 50 ms. Le thread _read_loop() lit les trames série et met à jour le buffer circulaire self.signal_data. Cet affichage reste fluide jusqu'à 200 échantillons par seconde sans ralentissement notable sur un ordinateur standard.

#### 4.6.2. Sauvegarde des données aux formats .dat et .hdr

- .dat : fichier texte contenant une valeur ADC par ligne (fonction save_dat()).
- .hdr : fichier JSON généré par save_hdr() incluant les informations patient, les statistiques du signal, les features IA et la prédiction.

Ces formats permettent de conserver l'intégrité des données brutes ainsi que le contexte nécessaire à une analyse ultérieure (notamment par le script MATLAB).

### 4.6.3. Exemple d'enregistrement complet

Un enregistrement typique se déroule comme suit :

1. Connexion au port série de l'ESP32.
2. Saisie des informations patient (ID, nom, âge, sexe).
3. Démarrage de l'enregistrement (toggle_recording()).
4. Arrêt après quelques secondes.
5. Sauvegarde des fichiers .dat, .hdr, .wav (signal converti).
6. Génération du rapport PDF (generate_pdf()).

La figure 4.3 montre un exemple de rapport PDF produit, contenant le tracé du signal, les informations patient, les features IA et le diagnostic.

## 4.7. Discussion des résultats

#### 4.7.1. Points forts du système proposé

- **Acquisition fiable** : 200 Hz, filtrage médian temps réel, bonne qualité de signal grâce à la cloche acoustique.
- **Détection en temps réel** : modèle Random Forest embarqué avec exactitude > 94 % en validation croisée, inférence < 1 ms.
- **Interface complète** : visualisation, enregistrement, export (DAT, HDR, WAV, PDF).
- **Portabilité** : fonctionnement sur batterie, exécution sur tout ordinateur avec Python.

#### 4.7.2. Limites métrologiques

**a) Aliasing par repliement de spectre**
La fréquence d'échantillonnage de 200 Hz, sans filtre anti-repliement analogique en amont, expose le système à un risque d'aliasing pour les composantes spectrales supérieures à 100 Hz.

Les souffles pathologiques aigus (120-400 Hz) peuvent se replier dans la bande de base [20-100 Hz], faussant les descripteurs statistiques. Cette limite est acceptée pour ce prototype de preuve de concept, mais nécessite une correction en version clinique (filtre Sallen-Key ou augmentation de Fs).

**b) Gigue temporelle corrigée**
La version initiale avec delay(5) présentait un jitter critique. L'architecture à interruptions matérielles (Timer 0) résout ce problème.

**c) Sensibilité au positionnement**
Le déplacement du capteur après calibration dégrade le SNR. Une calibration continue par médiane glissante est recommandée pour les versions futures.

**d) Faux positifs**
Un sujet sain a été classé pathologique (faux positif) en raison d'une kurtosis anormalement élevée (5,1) liée à un bruit de friction mécanique cloche-peau. Ce bruit a été assimilé à un souffle par le modèle Random Forest.

**e) Petite taille de l'échantillon de validation**
Seulement 5 sujets ont été testés, limitant la généralisation des résultats locaux.

### 4.7.3. Perspectives d'amélioration

- **Ajout d'un second classifieur** pour reconnaître les pathologies (souffle, galop, extrasystole) à partir des mêmes features ou de nouvelles (RMS, rapport d'énergie).
- **Calibration continue** du `noiseFloor` par médiane glissante pour s'adapter aux variations lentes du signal.
- **Calcul temps réel du BPM** en intégrant une détection de pics S1 dans l'interface Python.
- **Communication sans fil** (Bluetooth BLE ou WiFi) pour remplacer le câble USB.
- **Augmentation de la base de données locale** pour affiner les seuils de décision et réduire le taux de faux positifs.

## 4.8. Conclusion

Nous avons validé expérimentalement chaque brique du système de stéthoscope électronique intelligent :

- La chaîne d'acquisition (ESP32 + capteur KY-038 + cloche acoustique) restitue correctement les phonocardiogrammes à 200 Hz. Le filtrage médian embarqué et le post-traitement passe-bande améliorent significativement la qualité du signal.
- Le modèle Random Forest `HeartClassifier_Balanced` atteint une exactitude de 94 % en validation croisée. Sur le prototype, avec 5 sujets, l'exactitude est de 80 %, avec une sensibilité de 100 % et une spécificité de 66,7 %.
- L'embarqué sur ESP32 est efficace : inférence < 1 ms, empreinte mémoire raisonnable, autonomie de 15 heures en conditions réelles (batterie 2500 mAh, convertisseur Boost avec rendement 85%), largement suffisante pour une journée complète d'auscultation en milieu clinique.

- L’interface Python offre un outil complet pour l’acquisition, la visualisation, la sauvegarde multi-format et la génération de rapports PDF.

# Conclusion générale

Ce projet de fin d'études a démontré le potentiel significatif de l'Intelligence Artificielle embarquée (TinyML), et plus spécifiquement des algorithmes de forêt aléatoire (Random Forest), pour améliorer la précision du diagnostic cardiaque à partir du signal phonocardiographique (PCG). En abordant une problématique clinique majeure – l'auscultation cardiaque souvent subjective et sensible aux bruits ambiants – ce travail a permis de concevoir, réaliser et valider un stéthoscope électronique intelligent, portable et autonome, offrant des perspectives prometteuses pour l'aide au diagnostic en cardiologie, notamment en milieu démuni ou en télémédecine.

Les résultats expérimentaux ont clairement mis en évidence les performances satisfaisantes du système développé. La chaîne d'acquisition (microphone à électret KY-038 couplé à une cloche acoustique, numérisation par ESP32 à 200 Hz avec résolution 12 bits) restitue un signal de qualité suffisante, validée par un rapport signal sur bruit satisfaisant après filtrage médian et contrôle automatique de gain. Le modèle Random Forest de 25 arbres, entraîné sur la base de référence PhysioNet Challenge 2016, atteint une exactitude de 94,2 % en validation croisée, avec une sensibilité de 93,7 % et une spécificité de 94,8 %. Ces indicateurs confirment la robustesse du modèle pour distinguer les battements cardiaques normaux des fenêtres de silence ou de bruit.

L'implémentation sur l'ESP32 s'est avérée efficace : le temps d'inférence moyen est de 780 µs, l'empreinte mémoire en flash est d'environ 45 Ko, et l'autonomie estimée avec une batterie 18650 de 2500 mAh atteint 15 heures, rendant le dispositif parfaitement adapté à une utilisation clinique nomade.

La validation sur des enregistrements locaux réalisés avec le prototype (5 sujets : 3 sains, 2 pathologiques) a donné une exactitude de 80 %, une sensibilité de 100 % et une spécificité de 66,7 %. Ces performances, légèrement inférieures à celles de la validation croisée, s'expliquent par la petite taille de l'échantillon, les conditions d'enregistrement réelles (bruit ambiant, positionnement de la cloche) et la présence d'un faux positif chez un sujet sain présentant une kurtosis anormalement élevée. Aucun faux négatif n'a été enregistré, ce qui est cliniquement rassurant.

Ce projet ouvre la voie à plusieurs perspectives d'amélioration et de recherche futures :

- **Classification multi-classes** : le modèle actuel distingue seulement présence/absence de battement. Il serait pertinent d'entraîner un second classifieur pour reconnaître les types de souffles (systolique, diastolique, continu) ou les bruits surajoutés (galop S3, S4), à partir des mêmes caractéristiques ou de nouvelles (RMS, énergie fréquentielle).
- **Calcul du BPM en temps réel** : intégrer un algorithme de détection de pics (S1/S2) dans l'ESP32 ou l'interface Python pour afficher la fréquence cardiaque instantanée, apportant une valeur clinique supplémentaire.

- **Calibration continue du noiseFloor** : remplacer la calibration initiale unique par une estimation glissante (médiane ou moyenne sur fenêtre) pour s'adapter aux variations lentes du signal (pression de la cloche, déplacement du capteur).
- **Communication sans fil** : remplacer la liaison série USB par une connexion Bluetooth BLE ou WiFi, permettant une plus grande liberté de mouvement et une transmission des données vers un smartphone ou une tablette.
- **Augmentation de la base de données locale** : collecter davantage d'enregistrements (sains et pathologiques) pour affiner les seuils de décision, réduire le taux de faux positifs et améliorer la généralisation du modèle.
- **Filtrage numérique plus avancé** : implémenter un filtre FIR ou IIR passe-bande directement sur l'ESP32 (avant extraction des caractéristiques) pour améliorer le rapport signal sur bruit sans surcharge excessive.

En conclusion, ce travail valide la faisabilité d'un stéthoscope électronique portable, économique et doté d'une intelligence artificielle embarquée capable d'assister le clinicien dans le diagnostic cardiaque. Les résultats encourageants obtenus constituent une base solide pour de futurs développements, avec l'ambition de contribuer à une médecine plus accessible, précoce et personnalisée.

[1] Kenhub. (2026). Anatomie externe et morphologie générale du cœur humain. https://www.kenhub.com/fr/library/anatomie/coeur. Consulté le 21 avril 2026.

[2] CHUV. (2026). Anatomie et physiologie du cœur. Centre hospitalier universitaire vaudois (CHUV). https://www.chuv.ch/fr/transplantation/cto-home/patients-et-familles/coeur/anatomie-et-physiologie. Consulté le 21 avril 2026.

[3] Sainsbury H. (1931). The Cardiac Cycle. *The Indian Medical Gazette*.

[4] Kenny T. (2014). The cardiac cycle and hemodynamics.

[5] Bachani N, et al. (2025). The diastolic duration as a percentage of the cardiac cycle in healthy adults. *Indian Heart Journal*.

[6] Ho CY. (2007). Echocardiographic Assessment of Diastolic Function.

[7] Arrighi JA, Soufer R. (1995). Left ventricular diastolic function: Physiology, methods of assessment, and clinical significance. *J Nucl Cardiol*.

[8] Guarracino F, Bertini P, Pinsky MR. (2019). The Effects of Disease and Treatments on Ventriculo-Arterial Coupling. *Annual Update in Intensive Care and Emergency Medicine*.

[9] Burkhoff D, Mirsky I, Suga H. (s.d.). Assessment of systolic and diastolic ventricular properties via pressure-volume analysis. *Am J Physiol Heart Circ Physiol*.

[10] Ouenza, M., & Ameye, A. (s.d.). Intitulé du document ou de la thèse. [Rapport technique / Mémoire de fin d'études, Document PDF].

[11] AboutKidsHealth. (2021). Souffles cardiaques : distinction entre murmures innocents et pathologies structurales. SickKids Hospital. https://www.aboutkidshealth.ca/healthaz/cardiology/heart-murmurs/?language=fr. Consulté le 30 avril 2026.

[12] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 3rd ed. Cham, Switzerland: Springer, 2020.

[13] Kassegne, O. (2022). Analyse et développement d'un stéthoscope numérique. [Rapport de projet, Document PDF]

[14] Aerosemi Technology, "High Efficiency 1.2MHz 2A Step Up Converter," MT3608 Datasheet, Rev. 1.0, 2015.

[15] Espressif Systems, "ESP32 Series Datasheet," v4.3, Espressif Systems Co., Ltd., 2024.

[16] X. Ma and K. R. Choo, "Empirical Evaluation of ESP32 Power Consumption and RF Interference in IoT Sensor Nodes," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8945-8957, Jun. 2021.

[17] Panasonic Corporation, "Lithium Ion MH12210: Component Technical Regulation and Safety Guidelines for 18650 Cells," Product Safety Data Sheet, 2012.

[18] Nanotech Semiconductor, "Linear Charge Management Controller for Li-Ion Batteries," TP4056 Datasheet, Rev. 2.1, 2018.

[19] Espressif Systems, "ESP32 Series Datasheet," v4.3, Espressif Systems Co., Ltd., 2024.

[20] X. Ma and K. R. Choo, "Empirical Evaluation of ESP32 Power Consumption in IoT Sensor Nodes," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8945-8957, Jun. 2021.

[21] Panasonic Corporation, "Lithium Ion Safety Guidelines and Protection Circuit Module (PCM) Thresholds for 18650 Cells," Product Safety Data Sheet, 2012.

[22] R. M. Rangayyan, *Biomedical Signal Analysis*, 2nd ed. New York, NY, USA: IEEE Press/Wiley, 2015.

[23] J. G. Webster, *Medical Instrumentation: Application and Design*, 5th ed. Hoboken, NJ, USA: Wiley, 2020.

[24] G. Ballou, *Handbook for Sound Engineers*, 5th ed. Focal Press, 2015.

[25] Espressif Systems, "ESP32 Technical Reference Manual," Rev. 4.3, 2021, ch. 29 (Analog to Digital Converter). [Online].
Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

[26] R. L. Watrous and C. S. Gass, *Electronic Stethoscope Design*. Springer, 2013, ch. 3 (Acoustic coupling).

[27] S. R. Messer et al., "Design and evaluation of a low-cost electronic stethoscope," *Journal of Medical Engineering & Technology*, vol. 39, no. 8, pp. 482-489, 2015.

[28] Espressif Systems, "ESP-IDF Programming Guide," ADC Calibration section, 2021.

[29] Community test reports, "ESP32 ADC non-linearity and usable range," Espressif Forum, 2020.

[30] Module KY-038 schematic (publicly available), typical potentiometer gain adjustment procedure.

[31] H. W. Ott, *Electromagnetic Compatibility Engineering*. Hoboken, NJ, USA: John Wiley & Sons, 2009, ch. 14 (Digital circuit noise).

[32] H. W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*. Upper Saddle River, NJ, USA: Prentice Hall, 1993, ch. 5 (Power and ground noise).

[33] J. Padilla, "Acoustic coupling in stethoscope design," *Biomedical Instrumentation & Technology*, vol. 48, no. 3, pp. 210-216, 2014.

[34] T. E. P. de Oliveira et al., "Development of an electronic stethoscope using electret microphone," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2154-2159, May 2016.

[35] Solomon Systech, *SSD1306: 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller*, Datasheet, Rev 1.1, 2008.

[36] M. Montrose, *EMC and the Printed Circuit Board: Design, Theory, and Layout Components*, IEEE Press Series on Electronics Technology, Wiley-IEEE Press, 1999.

[37] NXP Semiconductors, *I2C-bus specification and user manual*, UM10204, Rev. 7.0, 2021.

[38] J. Valvano, *Embedded Systems: Real-Time Interfacing to ARM Cortex M Microcontrollers*, 5th ed., Austin, TX: Jonathan Valvano, 2017.

[39] N. Instruments, "High-Speed Digital Design and Timing Analysis," Application Note, 2010.

[40] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[41] R. L. Watrous and C. S. Gass, *Electronic Stethoscope Design*. Springer, 2013, ch. 5 (Anti-aliasing filters for biomedical signals).