

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A.MIRA-BEJAIA



Faculté de Technologie

Département d'Automatique, de Télécommunication et d'Electronique

Projet de Fin d'étude

Pour l'obtention du diplôme de MASTER en Automatique et Informatique Industrielle

Thème

Réalisation d'un dispositif portable pour l'analyse de signaux ECG.

Préparé par :

ABDELFETTAH Mustapha

Dirigé par :

M^{me} S.MEZZAH

2017/2018

Remerciement

Je tiens à exprimer mes vifs remerciements à ma promotrice « MEZZAH Samia », pour sa disponibilité, et ses judicieux conseils, sa patience et pour avoir proposé ce thème.

Et un remerciement chaleureux à monsieur ALOU HICHEM ; Mme brahmi thiziri ; à Mr ZIANI ; aux Dr kachbi et Dr AIT MOKHTAR ; Mr TOUATI pour m' avoir aidé dans ce travail.

Mes remerciements s'adressent également à tous les membres du jury pour l'honneur qu'ils me font en acceptant de juger mon travail.

Mes remerciements vont aussi à tous ceux et celles qui ont participé de près ou de loin à l'élaboration de ce mémoire.

Enfin, On tient à remercier tous nos amis et collègues pour leur soutien moral tout au long de cette préparation.

Dédicace

Je dédie ce modeste travail à :

Mes chers parents qui n'ont pas cessé de croire en moi et m'encourager et me soutenir dans les moments difficiles et qui ont prié Dieu pour moi, et qui ont su m'entourer de toute leur affection et amour.

Mes frères et sœurs et à toute la famille.

Mes binômes de fortune jugo et sami .

A tous mes amis, moon, lyas, yanis, redoune, rabeH, jony et bien sûr anis .

Mes camarades de promotion : Lydia ,thiziri ,souad, aicha, youyou, Imad,

Tables des matières

Liste des figures	1
Liste des tableaux	2
Introduction Générale	3
I.1 Introduction	4
I.2 Le Cœur	4
I.2.1 Fonctionnement électrique du cœur	5
I.3 Électrocardiographie (ECG)	6
I.3.1 Génération du complexe QRS	6
I.3.2 Différents segment du complexe QRS	7
I.3.3 Le rythme cardiaque	8
I.3.4 Rythme sinusal	8
I.4 Les pathologies cardiaques	8
I.4.1 Arythmies courantes	8
I.4.2 Extrasystole	9
I.4.3 Bradycardie	9
I.4.4 Tachycardie	9
I.5 Principe d'enregistrement	10
I.5.1 Dérivations précordiales unipolaires	10
I.5.2 Dérivations périphériques bipolaires	11
I.5.3 Dérivations périphériques unipolaires	11
I.6 Les sources de bruits dans les signaux électrocardiographiques	12
I.7 Techniques de réduction du bruit [9]	13
I.8 Conclusion	14
II.1 Introduction	15
II.2 Algorithme de Pan J. et Tompkins W.J [11]	16

II.3 Algorithme de la machine a état [12]	20
II.4 Conclusion	25
III.1 Introduction.....	26
III.2 Matériel utilisé	26
III.2.1 Carte Arduino UNO.....	27
III.2.2 Moniteur de fréquence cardiaque AD8232	28
III.3 Adaptation de l'algorithme de la machine a état pour l'implémentation matérielle	33
III.3.1 Algorithme en temps réel.....	33
III.3.2 Validation des données ECG synthétisées	33
III.4 Comparaison des résultats	36
III.4.1 Comparaison entre les résultats entre Matlab et Arduino	36
III.4.1.1 ECG réel mémorisé :.....	36
III.4.1.2 ECG réel mesuré.....	38
III.5 Conclusion	40
Conclusion Générale.....	42
Annexe	44

Liste des figures

Figure I.2-1 Anatomie du cœur humain	5
Figure I.3-1 Activité électrique du cœur et phases de l'ECG	7
Figure I.5-1 Dérivations précordiales unipolaires	11
Figure I.5-2 Dérivations bipolaires périphériques standard	11
Figure I.5-3 Dérivations unipolaires	12
Figure II.2-1 Décomposition des six étapes de l'algorithme	17
Figure II.2-2 Validation de détection du complexe QRS par seuillage	18
Figure II.2-3 Si aucun QRS n'est détecté 166% de l'intervalle RR moyen des 5 derniers complexes validés -, l'algorithme recommence la recherche de maximum avec un seuil abaissé à 10% de la moyenne des amplitudes des 5 derniers maxima validés	19
Figure II.3-1 Diagramme de la machine à état	21
Figure II.3-2 résultat de l'exécution du programme de la machine à état	25
Figure III.2-1 schéma simplifié du système réalisé.....	26
Figure III.2-2 Schéma du montage final.....	27
Figure III.2-3 Carte Arduino UNO.....	28
Figure III.2-4 La carte de monitoring de fréquence cardiaque AD8232.....	29
Figure III.2-5 configuration du moniteur de fréquence cardiaque AD8232	30
Figure III.2-6 électrode et câble pour moniteur de fréquence cardiaque AD8232	31
Figure III.2-7 Module bluetooth HC-05	31
Figure III.3-1 signal ECG généré à partir de la mémoire de l'Arduino	33
Figure III.3-2 organigramme du programme	35
Figure III.4-1 Variation des états par rapport au signal ECG	36
Figure III.4-2 Comparaison entre les pics R détectés avec le programme Arduino	36
Figure III.4-3 Comparaison entre les pics Q détectés avec le programme Arduino	37
Figure III.4-4 Comparaison entre les pics S détectés avec le programme Arduino	37
Figure III.4-5 Comparaison entre les pics T détectés avec le programme Arduino	38
Figure III.4-6 signal acquis par la carte Arduino	39
Figure III.4-8 pics R du signal mesuré détecté par le programme Arduino	39
Figure III.4-9 Comparaison entre les amplitudes des pics R détectés avec le programme Arduino et celui de Matlab pour le signal mesuré	40

Liste des tableaux

Tableau III.2-1 caractéristiques techniques Arduino UNO	28
Tableau III.2-2 raccordement du moniteur cardiaque à la carte Nano	30
Tableau III.2-3 emplacement des électrodes sur le corps [14].Module bluetooth HC-05	31
Tableau III.2-4 caractéristiques techniques du HC-05.....	32
Tableau III.2-5 raccordement module HC-05 à la carte Uno.	32
Tableau III.3-1 temps d'exécution moyen des étapes du programme	34

Introduction Générale

En Algérie les maladies cardiaques sont la première cause de mortalité, d'après les chiffres de 2014 fournis par la Société algérienne de cardiologie, elle représente à elle seule 41% des décès soit 25000 personnes chaque année. Un diagnostic précoce est donc nécessaire pour la gestion efficace des patients pour cela le plus souvent on analyse les signaux physiologiques.

Il représente un grand intérêt dans le domaine médical pour le diagnostic des maladies ils sont la représentation de l'activité électrique interne du corps humain et peuvent donc fournir de nombreuses informations sur l'état du patient.

Le développement d'appareils portatifs capables de récupérer ces données est une solution très intéressante puisqu' ils sont généralement non intrusifs ; légers ; bon marché et relativement faciles à utiliser, ils n'ont donc pas besoin de personnels médicaux qualifiés.

Ces systèmes peuvent communiquer à distance ce qui évite au patient et au médecin le déplacement il permettent donc le suivi du patient à distance et en temps réel.

Ce projet vise à réaliser un appareil portable dont la fonction est de détecter le signal ECG qui est le signal physiologique lié à l'activité cardiaque afin de le traiter et d'extraire des informations pertinentes concernant les paramètres du rythme cardiaque qui permettront de détecter certaines pathologies.

Généralités sur l'électrocardiogramme

I.1 Introduction

La première partie de ce chapitre est consacrée de manière générale à tout ce qui se rapporte à l'anatomie cardiaque et à la génération du complexe QRS, nous allons aussi aborder quelques pathologies cardiaques les plus fréquentes.

Dans la deuxième partie nous ferons le rappel de quelques notions de base sur l'électrocardiographie ; quelques techniques d'enregistrement ainsi que les sources de bruit et les techniques de réductions de ces dernières,

I.2 Le Cœur

Le cœur est composé de ventricules, deux cavités séparées entre elles par une cloison épaisse (Le septum intraventriculaire) et deux oreillettes qui sont deux petits réservoirs qui reçoivent le sang de la circulation et le retransmettent à leur ventricule respectif. L'oreillette droite reçoit le sang désoxygéné des veines caves supérieure et inférieure ainsi que du sinus coronaire qui draine le sang du myocarde. L'oreillette gauche récupère le sang oxygéné des poumons à travers quatre veines pulmonaires [1]. Le cœur éjecte simultanément le sang riche en oxygène vers la périphérie par le biais du ventricule gauche et le sang pauvre en oxygène vers les poumons par le biais du ventricule droit.

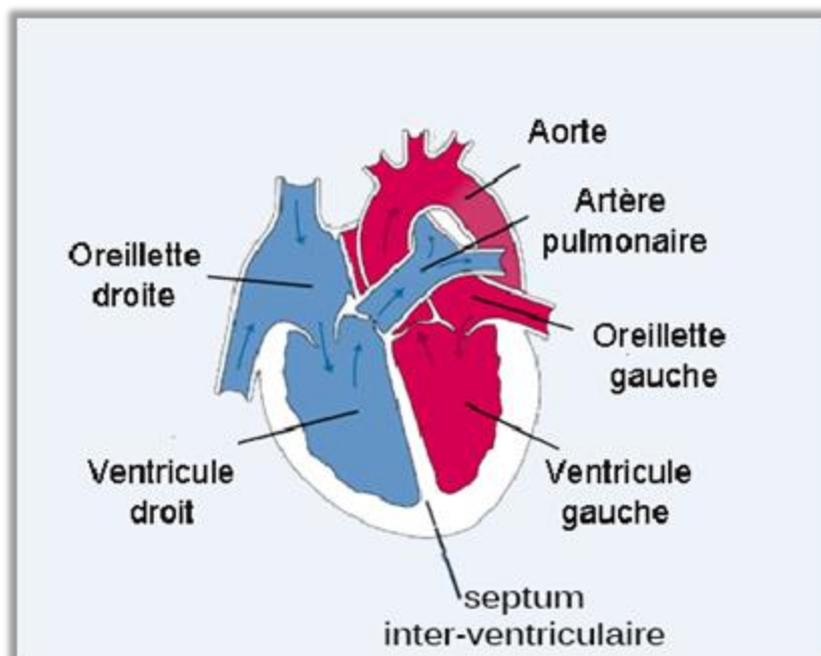


Figure I.2-1 Anatomie du cœur humain [2]

I.2.1 Fonctionnement électrique du cœur

La contraction du cœur est provoquée par la stimulation des impulsions électriques délivrées par le nœud sinusal définie comme le stimulateur cardiaque physiologique situé en haut et en arrière de l'oreillette droite à proximité de l'abouchement de la veine cave supérieur.

L'impulsion électrique arrive alors au nœud auriculo-ventriculaire, seul point de passage possible pour le courant électrique entre les oreillettes et les ventricules. A ce moment, l'impulsion électrique subie une courte pause permettant au sang de pénétrer dans les ventricules. Elle emprunte alors le faisceau de HIS, qui est composé de deux branches principales allant chacune dans un ventricule.

Les fibres constituant ce faisceau, complétées par les fibres PURKINJE grâce à leur conduction rapide, propagent l'impulsion électrique en plusieurs points des ventricules, et permettent ainsi une dépolarisation quasi instantanée de l'ensemble du muscle ventriculaire, malgré sa taille importante ; ce qui assure une efficacité optimale dans la propulsion du sang dans l'aorte ; cette contraction constitue la phase de systole ventriculaire. Puis la diastole ventriculaire; les fibres musculaire se re-polarisent et reviennent ainsi à leur état initial le cycle du battement cardiaque est alors terminé et le cœur est prêt pour un nouveau battement [4][5].

I.3 Électrocardiographie (ECG)

Le principe consiste à mesurer les impulsions électriques émises initialement par le nœud sinusal au moment de la contraction du muscle cardiaque. Cette technique est non-invasive mais nécessite de placer plusieurs électrodes sur la peau du patient. L'impulsion électrique se propage de l'électrode polarisée négativement à celle polarisée positivement et génère une déflexion positive sur l'électrocardiogramme. Au contraire, le signal évolue négativement lorsque le vecteur électrique est opposé à la polarisation des électrodes [1].

I.3.1 Génération du complexe QRS

Les principales ondes qui constituent l'ECG sont décrites comme suit:

- 1) L'onde P est une déflexion correspondant à la dépolarisation des oreillettes droite et gauche,
- 2) Le complexe QRS correspond à un ensemble de déflexions dues à la dépolarisation des ventricules
- 3) L'onde T est une déflexion correspondant à la repolarisation ventriculaire

Les valeurs normales des durées des déflexions sont :

- Onde P : inférieure ou égale à 0,12s
- Onde Q : inférieure à 0,04s
- Complexe QRS : compris entre 0,06 et 0,08s.
- Onde QT : sa durée varie selon la fréquence cardiaque, elle s'allonge quand la fréquence ralentit, et se raccourcit quand la fréquence accélère. Pour une fréquence de 60 cycles par minute, la durée de l'onde QT se situe dans un intervalle [0.35s, 0.43s] et pour une fréquence cardiaque de 80 cycles par minute, elle se situe plutôt entre 0.31s et 0.39s [3][5].

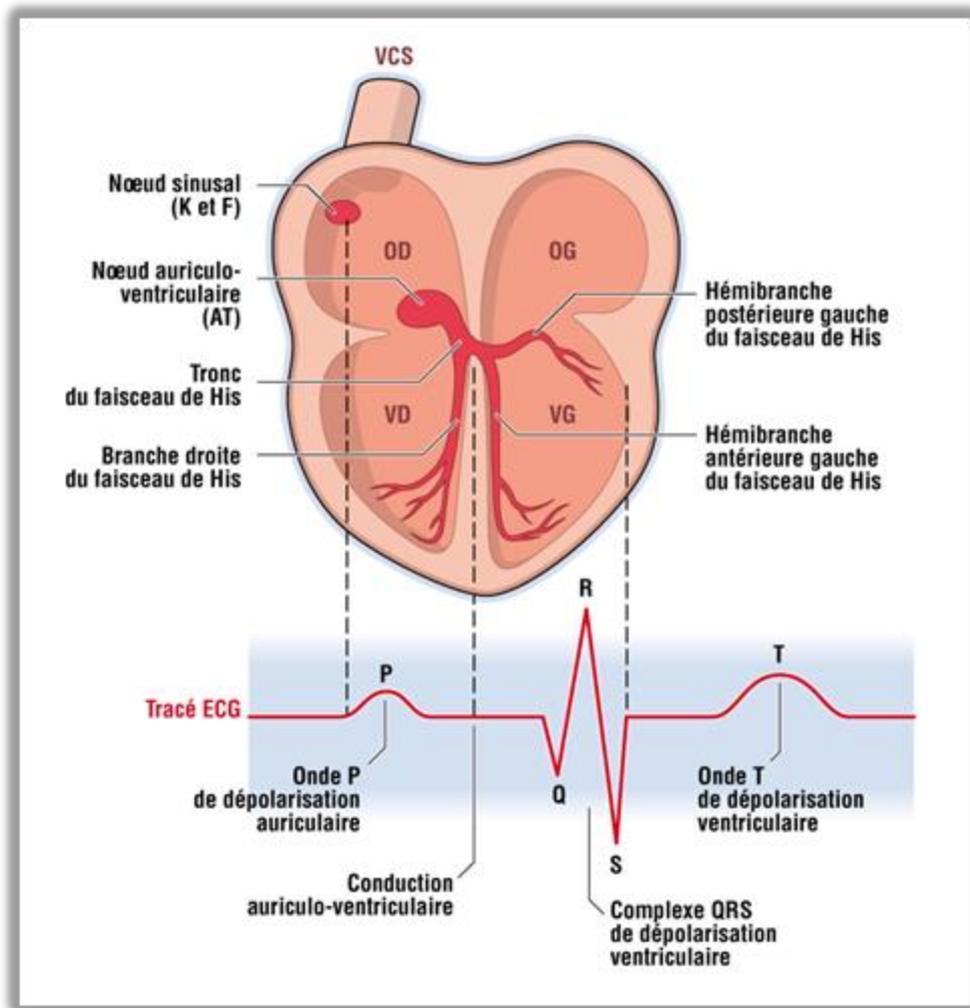


Figure I.3-1 Activité électrique du cœur et phases de l'ECG [2]

I.3.2 Différents segment du complexe QRS

- **Intervalle RR**

Mesuré entre deux sommets successifs de l'onde R, représente le temps qui sépare deux complexes QRS consécutifs. Si le rythme est régulier (intervalle RR constant), la fréquence cardiaque moyenne (exprimée en battements par minute) est égale à l'inverse de l'intervalle R-R (exprimé en secondes) [4].

$$\text{fréquence Cardiaque} = \frac{1}{RR} * 60 \quad (I.1)$$

- **Segment ST**

Correspond à la phase où les cellules ventriculaires sont dépolarisées ; il n'y a donc a priori pas de propagation électrique. Le segment ST va de la fin du complexe QRS jusqu'au début de l'onde T. sa durée est comprise entre 50 et 150 ms [4].

- **Segment QT**

Cet intervalle est mesuré depuis le début du complexe QRS jusqu'à la fin de l'onde T. il représente la durée totale de la dépolarisation et de la repolarisation ventriculaire. Sa valeur dépend de la fréquence cardiaque. Il est normalement compris entre 350 et 440 ms [4].

I.3.3 Le rythme cardiaque

La fréquence cardiaque ou le rythme cardiaque peut être mesuré par le pouls. C'est le nombre de pulsation (battement) cardiaques par minute. Il doit, théoriquement, être de 70 pulsations/min chez un homme au repos et doit être plus élevé lors de la pratique d'un sport, pendant la digestion,...etc. S'il apparaît trop élevé, on parle de tachycardie. S'il apparaît trop bas on parle de bradycardie. Le rythme cardiaque est un outil de diagnostic indispensable pour les troubles cardiaques, notamment en cas d'arythmie [6].

I.3.4 Rythme sinusal

Rythme normal du cœur selon l'électrocardiogramme. Il est avec une synchronisation correcte de la contraction du cœur suivant la séquence nœud sinusal. Il est caractérisé par un rythme cardiaque régulier, compris entre 60 et 80 battements/minute chez l'adulte au repos, avec un intervalle régulier entre chaque battements normaux [6].

I.4 Les pathologies cardiaques

Le meilleur outil pour diagnostiquer les arythmies est l'électrocardiogramme. Dans l'analyse de l'ECG, les pathologies sont détectées en fonction de leur différence avec le rythme idéal qui est le rythme sinusal. Chaque déviation visible sur l'ECG par rapport au rythme sinusal peut être attribuée à une anomalie physiologique [6].

I.4.1 Arythmies courantes

Les arythmies cardiaques sont une cause majeure de mortalité et constituent une partie très importante de la problématique des maladies cardiovasculaires. L'arythmie survient lorsque le

cœur bat de façon irrégulière ou s'il bat à moins de 60 pulsations ou plus de 100 pulsations / minute, sans que cela soit justifié par un effort physique par exemple [6].

L'arythmie est le trouble cardiaque le plus fréquent. Les impulsions électriques qui contrôlent les battements du cœur se produisent de façon désordonnée ou ne passent pas par les circuits électriques habituels. La contraction cardiaque qui résulte s'écarte du rythme normal. Le battement supplémentaire qu'elle provoque est appelé extrasystole, par opposition à la contraction normale appelée systole [6].

La durée d'une arythmie varie beaucoup d'un individu à l'autre et dépend aussi du type d'arythmie. Les arythmies sont classées selon le lieu où elles prennent naissance. Si le trouble se déclare dans l'oreillette ou l'AT, on parlera d'arythmie supra-ventriculaire ; s'il apparaît dans les ventricules, on parlera d'arythmies ventriculaires. Si une arythmie accélère le rythme cardiaque, on parle de tachycardie ; si elle ralentit, on parle alors de bradycardie [6].

I.4.2 Extrasystole

Les extrasystoles sont des battements du cœur provoqués, non par les impulsions du nœud sinusal, mais par des impulsions déclenchées ailleurs dans les oreillettes (extrasystoles supraventriculaires) ou dans les ventricules (extrasystoles ventriculaires). Ces battements perturbent le rythme sinusal et s'accompagnent d'une sensation de palpitation ou de pause cardiaque. La fréquence cardiaque est supérieure à 100bpm [6].

I.4.3 Bradycardie

C'est un battement lent du rythme cardiaque, sa fréquence est inférieure à 60bpm ; elle est dite d'origine sinusale, jonctionnelle, ou ventriculaire, selon l'origine de l'impulsion électrique au début des battements considérés [6].

- **Bradycardie sinusal :** Le rythme est de l'ordre de 40bpm
- **Bradycardie jonctionnelle :** Le rythme est de l'ordre de 35bpm.
- **Bradycardie ventriculaire :** Le rythme a une fréquence très lente, entre 15 et 40bpm [6].

I.4.4 Tachycardie

A l'inverse de la bradycardie, la tachycardie est caractérisée par le fait que la fréquence est supérieure à 100bpm. Elle peut être d'origine sinusale, auriculaire ou ventriculaire.

- **Tachycardie Sinusale** : Le rythme est compris entre 100 et 180bpm.
- **Tachycardie auriculaire** : Le rythme est compris entre 120 et 200bpm.
- **Tachycardie ventriculaire** : les battements sont très rapprochés [6].

I.5 Principe d'enregistrement

Le signal ECG peut être enregistré sur différents sites du corps d'un sujet par l'intermédiaire d'électrode. Selon leurs emplacements, on distingue deux types de dérivations :

- Les dérivations précordiales (appelées V) qui sont placées sur le thorax du patient pour explorer le plan transversal du cœur [5].
- Les dérivations périphériques ou frontales (appelées D) qui sont placées sur les 4 membres et qui explorent le plan frontal du cœur [5].

I.5.1 Dérivations précordiales unipolaires

Les électrodes exploratrices sont placées près du cœur, en des endroits précis comme indiquer dans la figure suivante. Ainsi, on reconnaît six Types de dérivations précordiales :

- V1 : électrode placée à droite du sternum
- V2 : électrode placée à gauche du sternum
- V3 : électrode placée à mi-distance entre V2 et le mamelon
- V4 : électrode placée sous le mamelon
- V5 : électrode placée à mi-distance entre le mamelon et la ligne axillaire
- V6 : électrode placée sur la ligne axillaire [7].

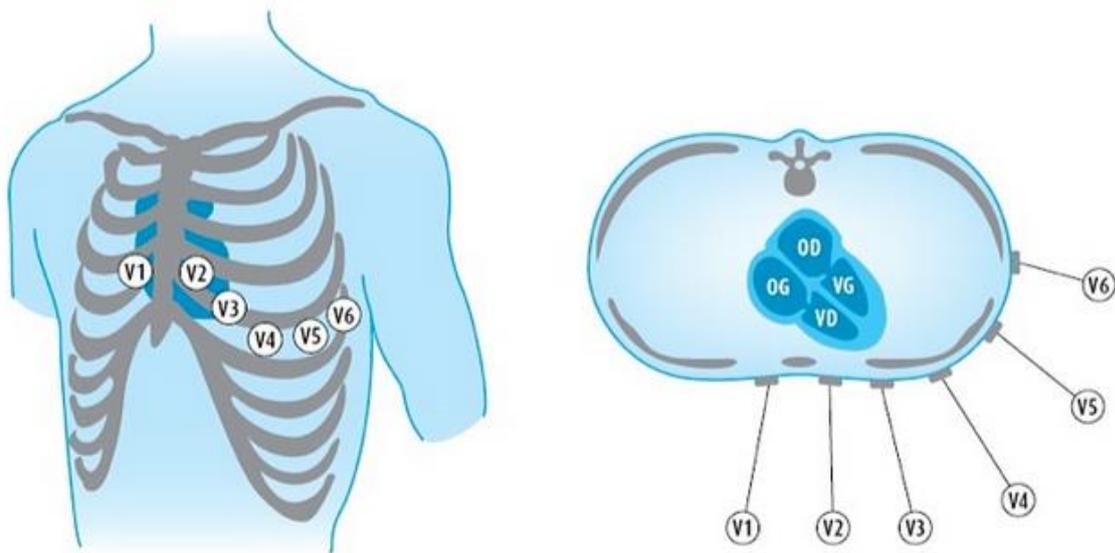


Figure I.5-1 Dérivations précordiales unipolaires [9].

I.5.2 Dérivations périphériques bipolaires

Elles sont au nombre de trois. Ces dérivations sont bipolaires. Elles permettent la mesure des différences de potentiel entre des points très éloignés au niveau du corps. On distingue :

- La dérivation I (DI) : Mesure bipolaire entre bras gauche et bras droit.
- La dérivation II (DII) : Mesure bipolaire entre jambe gauche et bras droit.
- La dérivation III (DIII) : Mesure bipolaire entre jambe gauche et bras gauche [7].

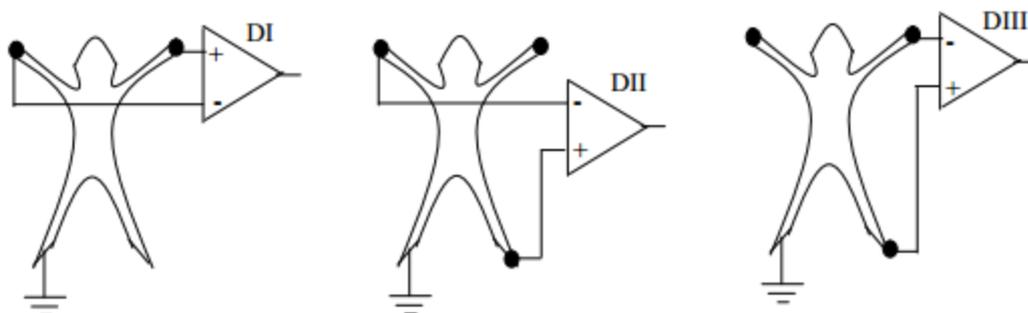


Figure I.5-2 Dérivations bipolaires périphériques standard [7]

I.5.3 Dérivations périphériques unipolaires

Elles permettent de mesurer les variations de potentiel aux extrémités du corps au moyen d'une électrode exploratrice (pôle positif). Ce potentiel étant comparé à celui d'une électrode de référence (pôle négatif) dont le potentiel est nul. Les dérivations sont connues par aVR, aVL et aVF ; où "a" ; indique que le potentiel recueilli par chacune de ces dérivations est amplifié.

Le "V" signifie qu'il s'agit d'une dérivation unipolaire. "R", "L", "F" est (Right, Left, Foot) [7].

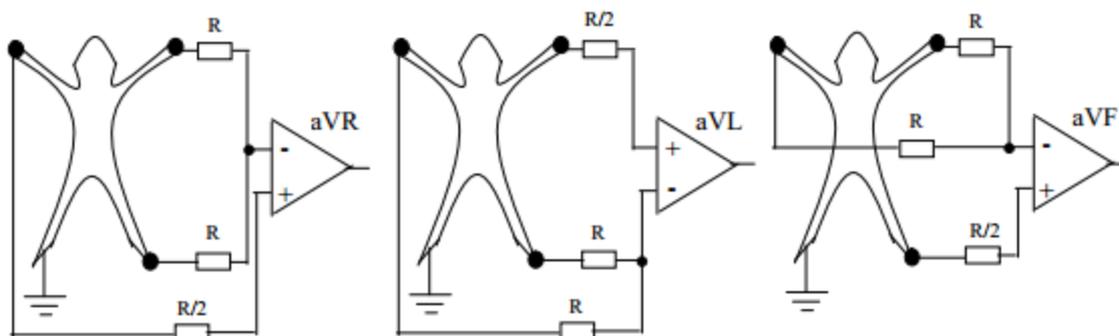


Figure I.5-3 Dérivations unipolaires [7]

I.6 Les sources de bruits dans les signaux électrocardiographiques

Le signal ECG obtenu lors de l'enregistrement est généralement contaminé avec différentes sources de bruits. Celles-ci peuvent perturber les caractéristiques de phase et d'amplitude du signal utile. Parmi ces bruits on peut citer [7][8][3] :

- **Electromyogramme (EMG)**

Il est dû aux variations de potentiel engendrées au sein des tissus musculaires. Son amplitude est la même que le signal ECG mais il se produit dans les hautes fréquences (25-100Hz).

- **Les bruits provoqués par la ligne de base**

La ligne de base est la ligne d'équilibre de l'activité cardiaque : Elle doit être isoélectrique. Des ondulations de très basses fréquences dues aux mouvements du sujet où au mauvais contact des électrodes, peuvent perturber cette ligne (fréquence < 5Hz).

- **Interférence du réseau 50 Hz**

Le bruit 50Hz est un bruit qui provient de l'alimentation par le réseau de distribution électrique. Il contamine le signal électrocardiographique ECG avec des oscillations dont l'harmonique fondamentale est à 50 Hz. Généralement, ce bruit est présent dans tous les enregistrements et il peut être assez fort, cependant il s'élimine facilement avec une opération de filtrage sélectif.

- **Artefacts de mouvements**

C'est la forme de bruit la plus difficile à extraire des signaux ECG, et ceci en raison du chevauchement de son spectre avec celui de l'ECG ainsi que sa morphologie qui ressemble à celles des ondes P, QRS, T (fréquence 1-10Hz).

- **Autres bruits**

Parmi les autres bruits souvent rencontrés, nous pouvons citer les artefacts dus aux :

- Mouvements des câbles électriques.
- La saturation des instruments de mesure
- Mauvaise qualité du câblage
- Port de vêtements synthétiques.
- Ondes RF émises par les équipements électro-chirurgicaux.

I.7 Techniques de réduction du bruit [9]

Comme vu précédemment . Il y a des bruits de hautes fréquences, provoqués par l'activité musculaire extracardiaque et les interférences dues aux appareils électriques, et des bruits de basses fréquences provoqués par les mouvements du corps liés à la respiration, les changements physicochimiques induits par l'électrode posée sur la peau et les micro variations du flux sanguin.

Pour réduire ces bruits, il faut avoir une respiration calme et éviter les mouvements ou le contact avec du métal. Il faut bien préparer la peau avant de placer une électrode (rasage des poils, lavage simple et frottement pour améliorer le flux capillaire pour les électrodes périphériques, éviter l'utilisation d'alcool). Il faut aussi éviter les chevauchements des fils d'enregistrement (boucle).

On peut recourir à plusieurs types de filtre en cas de parasites :

- **Filtre passe-bas**

Pour supprimer les bruits à très basse fréquence, on utilise un filtre passe-haut qui supprime en mode réel les bruits en dessous du seuil de 0,05 Hz.

- **Filtre passe-haut**

Pour supprimer les bruits à haute fréquence, on utilise un filtre passe-bas qui supprime en mode réel les bruits au-dessus de 150 Hz. La bande passante recommandée se situe entre 0,05 Hz et 150 Hz. Mais la majorité des appareils du marché proposent des filtres pré réglés entre 0,5 et 40-50 Hz, car le tracé est plus stable et moins parasité.

I.8 Conclusion

Les notions évoquées dans ce chapitre notamment l'anatomie cardiaque et les différents aspects relatifs à l'ECG nous permettent de mieux comprendre la problématique et de mieux aborder la suite de ce travail.

Algorithmes de détections

II.1 Introduction

Le traitement automatisé des signaux ECG a gagné un large intérêt au cours des dernières années. Le but est de fournir des méthodes d'aide au diagnostic en développant divers

algorithmes afin de compléter ou même de substituer l'apport du médecin. La plupart de ces méthodes suivent les étapes ci-après :

- L'acquisition des signaux ECG et leur transformation dans un format qui soit utilisable pour le traitement automatique.
- Le prétraitement : comme vu dans le chapitre I lors du processus d'acquisition, les Signaux ECG sont affectés par de multiples sources de bruits. L'étape de prétraitement doit réduire le plus possible leurs effets.
- La détection des pics : étape importante et délicate du traitement Automatique, cette Étape permet d'isoler les ondes P et T et le complexe QRS.
- L'extraction des paramètres : il s'agit de trouver, à partir des ondes élémentaires, Les paramètres qui puissent prédire le mieux l'état du patient.
- La classification : en utilisant les paramètres, la classification doit être capable de donner le diagnostic, en se substituant au rôle du médecin.

Chaque étape est particulièrement délicate et nécessite une qualité quasi parfaite. Le succès de la chaîne automatique est assuré seulement dans le cas d'un fonctionnement irréprochable de chacune de ses composantes [10].

Dans ce chapitre nous allons exposer deux algorithmes de traitement du signal ECG à savoir :

- Algorithme de Pan et Tompkins.
- Algorithme de la machine à état.

Nous ferons une présentation et une documentation de l'algorithme de la machine à état tout en expliquant les choix pris dans son développement. Notre conclusion se fera par l'explication des raisons du choix de cet algorithme pour notre application.

II.2 Algorithme de Pan J. et Tompkins W.J [11]

L'algorithme comprend six étapes. En dehors de la dernière (6-seuillage adaptatif), il se décompose de la manière suivante :

- 1) Filtrage passe-bande: D'après les données physiologiques, les ondes R ont une énergie maximale dans la bande 5-15 Hz. Il commence donc par un filtrage du signal dans cette bande.
- 2) Dérivation: Le complexe QRS étant bref (entre 0.02 et 0.2 seconde), et de forte amplitude, la dérivée du signal à ce niveau présente des valeurs maximales élevées. On continue donc le traitement du signal par l'application d'un filtre dérivateur numérique.
- 3) Transformation non linéaire: Afin d'accentuer le contraste mis en évidence par la dérivation, on élève au carré le signal obtenu, ce qui permet, en outre, d'éliminer le signe et d'assurer la symétrie de détection entre les ondes R et les ondes QS.
- 4) Intégration: L'intégration du signal suivi du filtrage passe bas permet d'obtenir un maximum unique pour chaque complexe. La taille de la fenêtre d'intégration doit être adaptée à la largeur moyenne d'un complexe QRS. Si elle est trop grande, le maximum est décalé en temps par rapport à la position du R (influence de l'onde T) ; si, au contraire, elle est trop petite, on obtient plusieurs pics pour une même onde R. La taille, choisie ici de manière empirique, est de 0.15 seconde ce qui correspond au double de la largeur moyenne d'un complexe QRS.
- 5) Filtrage passe bas: Le signal obtenu est filtré passe-bas afin d'enlever le bruit haute fréquence résiduel, et ainsi obtenir un maximum unique pour chaque complexe.
- 6) Seuillage adaptatif : À l'issue du traitement précédent, le signal disponible possède un maximum absolu pour Chaque complexe QRS ; il possède en outre d'autres maxima locaux, de plus faible amplitude en général, qui correspondent soit à du bruit, soit aux ondes T. Cette étape consiste donc en une recherche de maxima capable de ne pas prendre en considération les maxima trop faibles qui peuvent correspondre au bruit en effectuant un seuillage en amplitude et ceux trop proches qui risquent d'être introduits par les ondes T en effectuant un seuillage temporel ; ces deux types de seuillage sont décrits aux paragraphes suivants.

En outre, l'amplitude des battements pouvant varier fortement au cours d'un même enregistrement et d'un enregistrement à l'autre, les seuils utilisés ne peuvent pas être fixés a priori : ils sont donc calculés tout au long de l'analyse.

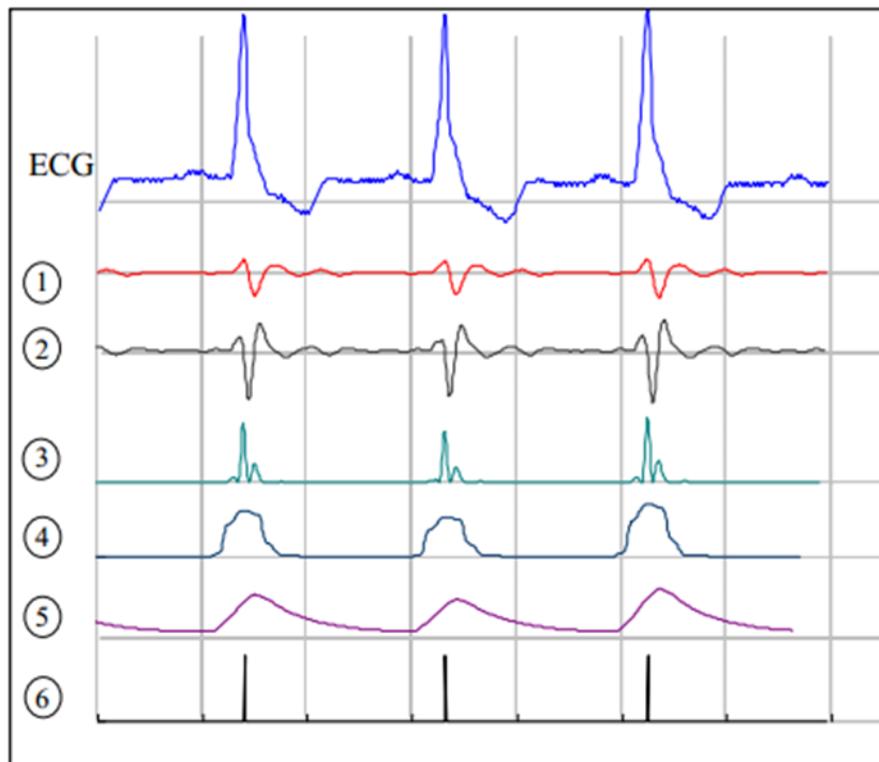


Figure II.2-1 Décomposition des six étapes de l'algorithme.[11]

6. a) Seuil adaptatif en amplitude

Le seuillage en amplitude permet de distinguer les maxima correspondants aux ondes R de ceux correspondants aux ondes T qui sont en général beaucoup plus faibles. L'algorithme calcule un seuil qui vaut 30 % de l'amplitude moyenne des 5 dernières ondes R détectées, et recherche le prochain maximum. Si ce maximum est d'amplitude supérieure au seuil, il est

considéré comme correspondant à un complexe QRS et est ainsi validé ; sinon, l'algorithme continue jusqu'au maximum suivant.

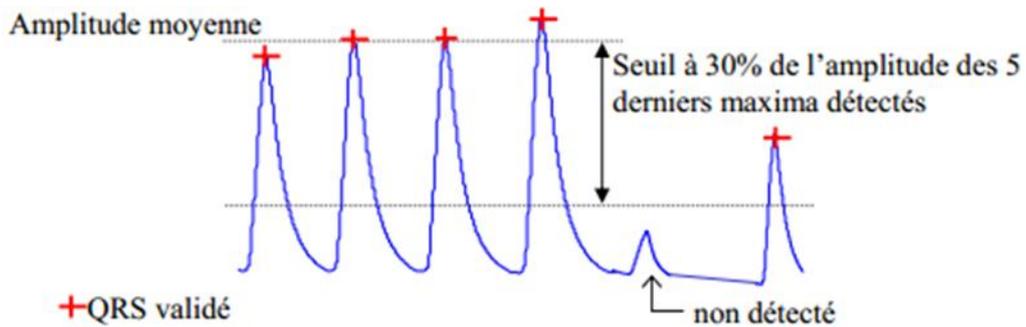


Figure II.2-2 Validation de détection du complexe QRS par seuillage [11]

Autrement dit, on recherche ici les maxima du signal qui sont supérieurs à un seuil ajusté tout au long de l'algorithme. Mais cette simple adaptation du seuil en amplitude ne suffit pas : dans plusieurs cas (changement de position, extrasystole ventriculaire, ...) on peut observer une baisse soudaine de l'amplitude des ondes R donc de l'amplitude des maxima associés ; dans ce cas, l'algorithme est mis en défaut car les maxima sont tous en dessous du seuil.

6.b) Seuil adaptatif en temps :

Pour surmonter cette difficulté, on tient compte de la forte probabilité d'observer un QRS dans un certain intervalle de temps : si aucun nouveau complexe n'est détecté dans un intervalle de 166% de la moyenne des sept derniers intervalles RR validés, l'algorithme recommence la

recherche de maximum avec un seuil abaissé à 10% de la moyenne des 5 derniers maxima, au lieu des 30% précédents.

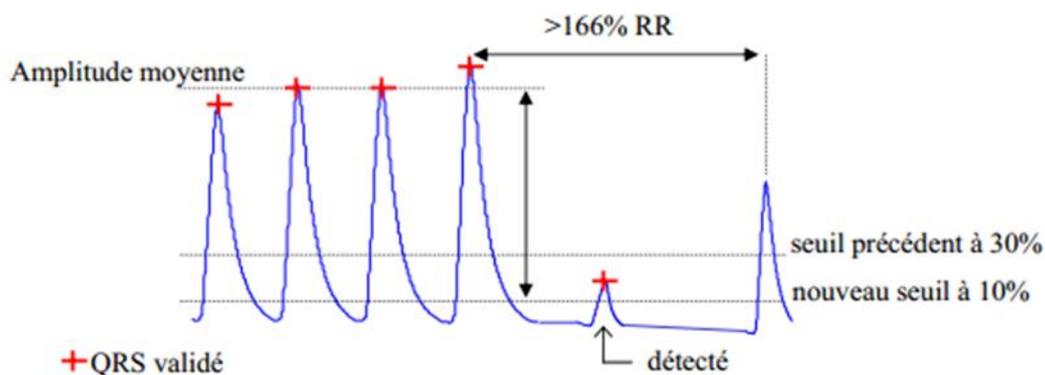


Figure II.2-3 Si aucun QRS n'est détecté 166% de l'intervalle RR moyen des 5 derniers complexes validés -, l'algorithme recommence la recherche de maximum avec un seuil abaissé à 10% de la moyenne des amplitudes des 5 derniers maxima validés [11].

Ce système permet dès lors la détection de petits complexes parmi d'autres de grande Amplitude, tout en ne détectant pas les petites ondes trop proches de celles déjà validées.

II.3 Algorithme de la machine a état [12]

Initialisation : Sur un intervalle de deux secondes qui correspond en moyenne à trois cycles de battement cardiaque consécutif on définit deux seuils adaptatifs S_T et S_{DC} .

Seuil S_T : Moyenne des N premiers échantillons ecg_i qui correspondent à 2 secondes d'enregistrement ($N = 2fs$). Ce seuil, qui sera utilisé pour la détection de l'onde T, est défini comme suit :

$$S_T = \frac{1}{N} \sum_{i=1}^N ecg_i \quad (\text{II.1})$$

Seuil S_{DC} : Moyenne des distances entre les N premiers échantillons et le seuil S_T (moyenne des écarts absolus).

$$S_{DC} = \frac{1}{N} \sum_{i=1}^N |ecg_i - S_T| \quad (\text{II.2})$$

La moyenne des écarts absolu est un paramètre de dispersion absolue qui indique de combien les valeurs d'une distribution s'écartent en général de la valeur de référence. ici le S_T . Ce seuil sera utilisé pour la détection du pic R ; d'après les données physiologiques nous savons que le pic R a une forte amplitude en comparaison du reste des points du signal qui ont une distribution plus faible vis avis du S_T , donc le pic R aura une amplitude supérieure à l'écart absolu moyen S_{DC} du signal.

Vue que L'amplitude des battements varie fortement au cours d'un même enregistrement ou d'un enregistrement à l'autre, les seuils utilisés ne peuvent pas avoir des valeurs constantes ils sont donc recalculés toutes les deux secondes tout au long de l'analyse.

Afin de lisser le signal ECG pour éliminer les fluctuations, On utilise une moyenne glissante avec une fenêtre de n échantillons avec $n = 0.04fs$. Le signal obtenu, notée ECG_m sera utilisé pour le reste du traitement.

Le diagramme des transitions de la machine à états utilisés est donné par la figure (II.3-1).

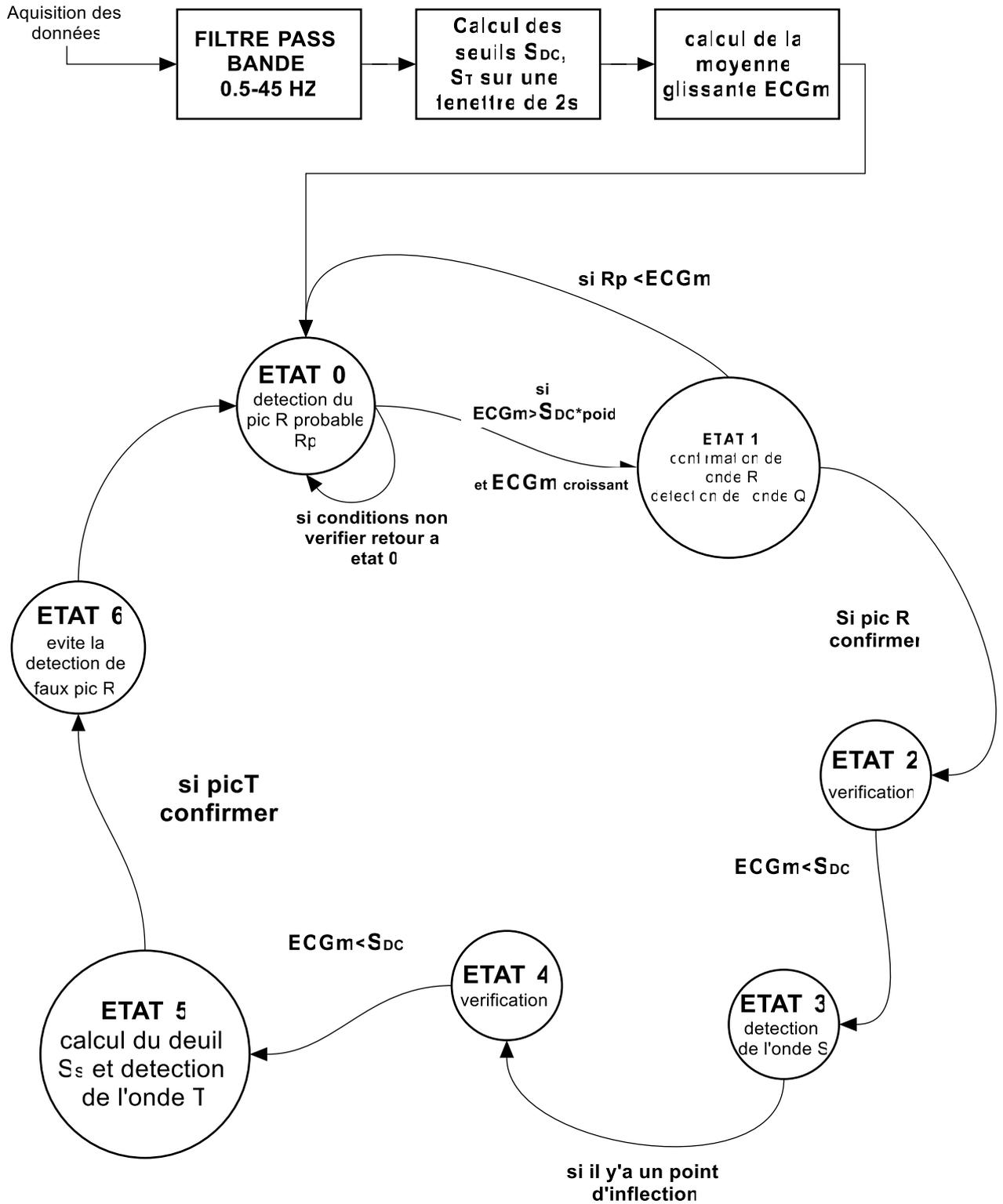


Figure II.3-1 Diagramme de la machine à état [12].

Etat 0 : (Détection de l'onde R)

L'estimation de la localisation du pic R se fait en considérant les deux conditions suivantes nommées **condition 1**, **condition 2** :

- **Condition 1:** Si la moyenne $ECGm_i$ est supérieure au seuil S_{DC} pondéré avec un poids p .

$$ECGm_i > S_{DC} \times p \quad (\text{II.3})$$

- **Condition 2:** Et si la moyenne mobile $ECGm_{(n-1)}$ est supérieure à la moyenne glissante $ECGm_{(n-2)}$.

$$ECGm_{(n-1)} > ECGm_{(n-2)} \quad (\text{II.4})$$

Si ces deux conditions sont vérifiées, alors on suppose que $ECGm_{(n-2)}$ est le pic R et on passe à l'état 1 pour confirmation. Dans le cas où l'une des conditions n'est pas vérifiée, on reste à l'état 0 la fenêtre se décale d'une position et la moyenne $ECGm_{(n-1)}$ est calculée.

Etat 1 : (confirmation du pic R et détection de l'onde Q)

Dans l'état 1, le pic R supposé (R_p) dans l'état 0 est comparé aux quatre moyennes glissantes suivantes, pour s'assurer que c'est bien l'amplitude maximale. Chaque moyenne dans cette fenêtre doit être inférieure à la moyenne qui la précède. Si cette condition (**Condition 3**) est vérifiée alors le pic R est confirmé.

$$R_p > ECGm_{(n+1)} > ECGm_{(n+2)} > ECGm_{(n+3)} > ECGm_{(n+4)} \quad (\text{II.5})$$

Pour détecter l'onde Q, on définit un intervalle qui s'étale de 0.04s avant l'emplacement du pic R, jusqu'à l'emplacement du pic R considéré ; cet intervalle correspond à la durée maximale d'une onde Q normale ($< 0.04s$)[5] ; Le minimum des valeurs de cet intervalle est considéré comme pic Q.

L'adaptation de la valeur du poids, utilisé pour la détection du pic R, s'effectue tous les huit pics R en utilisant l'expression suivante :

$$p = 0.3 \left(\frac{1}{8} \sum_{m=1}^8 R_m \right) \frac{1}{S_{DC}} \quad (\text{II.6})$$

Etat 2 : (Vérification)

Dans cette étape nous allons vérifier si la moyenne ECG_m est inférieure au seuil S_{DC} (**Condition 4**) puisque l'onde S est une onde négative

$$\text{Condition 4:} \quad ECG_m < S_{DC} \quad (\text{II.7})$$

Etat 3 : (Détection de l'onde S)

Dans cette étape, la phase de recherche ne doit pas dépasser 0.2 s. Dans un premier temps l'algorithme recherche le point d'inflexion où la pente devient ascendante (**condition5**) ; Pour vérifier si le point est bien le pic S l'algorithme compare les trois moyennes suivant le point S présumé (S_p) et si chaque moyenne est supérieure à la moyenne précédente le pic S est confirmé (**Condition 6**) et on passe à l'état 4.

$$\text{Condition 5:} \quad ECG_{m_n} > ECG_{m_{(n-1)}} \quad (\text{II.8})$$

$$\text{Condition 6:} \quad S_p < ECG_{m_{(n+1)}} < ECG_{m_{(n+2)}} < ECG_{m_{(n+3)}} \quad (\text{II.9})$$

Dans le cas où aucun pic S n'est pas détecté et que la période de recherche n'est pas terminée on passe directement à l'état 4. Le période de 0.2s correspond en moyenne à la période entre le pic R et l'onde T.

Etat 4 : (Vérification)

Dans cette étape nous allons vérifier si la moyenne ECG_m est inférieure au seuil S_{DC} puisque l'onde S est une onde négative (**Condition 7**).

$$\text{Condition 7:} \quad ECG_m < S_{DC} \quad (\text{II.10})$$

Etat 5 :

En premier lieu nous définissons un seuil S_S basé sur le dernier pic S détecté ; ce seuil nous permettra de trouver le début de l'onde T il est défini comme suit :

$$S_S = \frac{3}{4} (|S_T| - |S|) + S \quad (\text{II.11})$$

Pour trouver le début de l'onde T, il faut avoir au minimum 3 moyennes $ECGm$ successives qui sont supérieures au seuil S_S (**Condition 8**).

$$\text{Condition 8 : } \begin{cases} ECGm_n > S_S \\ ECGm_{(n+1)} > S_S \\ ECGm_{(n+2)} > S_S \end{cases} \quad (\text{II.12})$$

Pour trouver le pic T il nous faut chercher le point d'inflexion c.-à-d. le point où l'allure des moyennes ECG_M devient descendante (**Condition 9**).

$$\text{Condition 9 : } ECGm_n < ECGm_{(n+1)} \quad (\text{II.13})$$

Lorsque le pic T probable (T_p) est détecté on compare les huit moyennes suivantes et si chaque moyenne est inférieure à la précédente (**Condition 10**) alors T_p est détecté comme pic T.

$$\text{Condition 10 : } T_p > ECGm_{(n+1)} > ECGm_{(n+2)} > \dots > ECGm_{(n+8)} \quad (\text{II.14})$$

Etat 6 : (Vérification)

Cet état est utilisé pour éviter la détection d'un bruit ou d'un autre pic, cela évite la détection de deux pics d'ondes R à moins de 0.4 s d'intervalle.

Nous pouvons remarquer que cet algorithme ne prend pas en compte la détection des pic P en conséquence il nous permet seulement de détecter des pathologies ventriculaires qui sont liées au complexe QRS et nous permet pas de détecter les pathologies auriculaires qui sont liées au pic P.

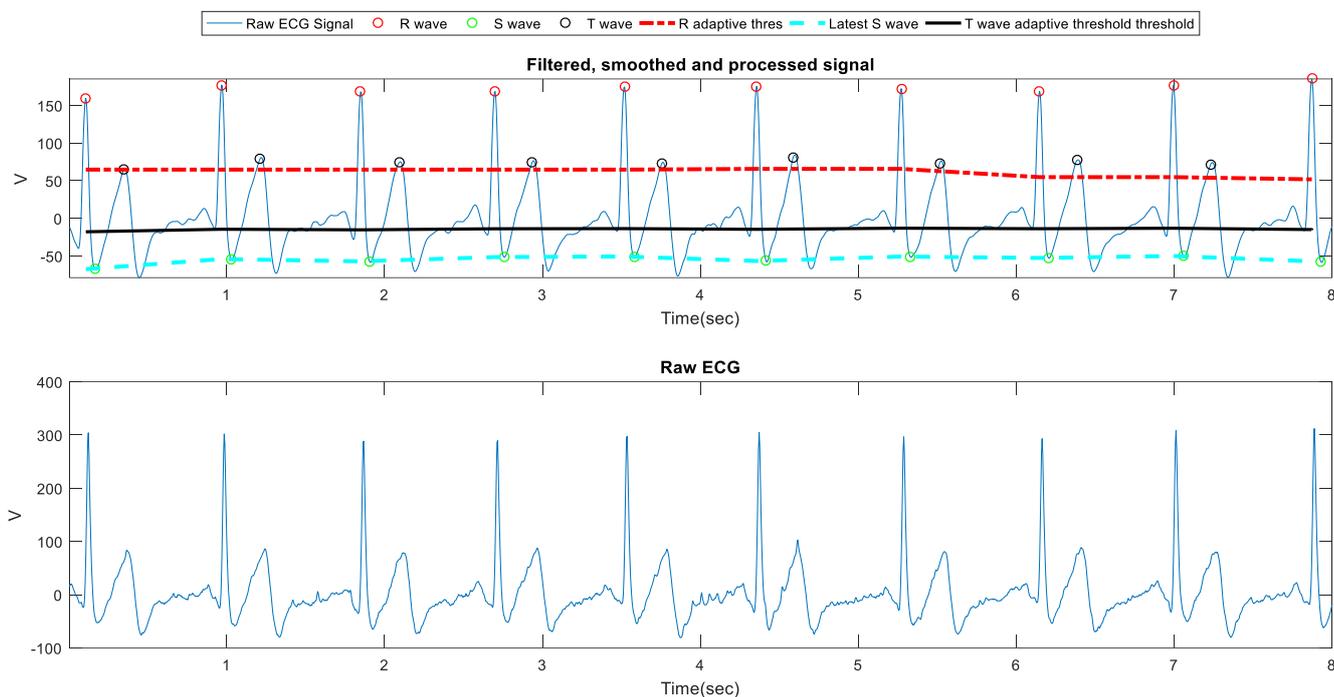


Figure II.3-2 résultat de l'exécution du programme de la machine à état[12].

II.4 Conclusion

Le choix de l'algorithme de la machine à état a été fait selon les critères suivants :

- L'algorithme est basé sur le principe de la machine à état cela permet de réduire le temps d'exécution.
- Il utilise des fonctions non bloquantes qui ne bloquent pas l'exécution du programme durant son déroulement. Ce type de fonction est très utile pour les applications en temps réels.
- L'algorithme est basé sur des éléments relativement faciles à comprendre donc il est un atout majeur si on veut l'améliorer

Implémentation

III.1 Introduction

Dans ce chapitre nous allons aborder les différents aspects qui se rapportent à l'implémentation matérielle de l'algorithme de machine à états pour la détection du complexe QRS. Ce chapitre est divisé en trois parties qui sont :

- Matériel utilisé.
- Adaptation de l'algorithme à l'implémentation matérielle avec une carte Arduino.
- Résultats obtenus après l'exécution du programme.

III.2 Matériel utilisé

Le montage que nous avons réalisé ce compose essentiellement des modules illustrés par le schéma de la figure (III.2-1).

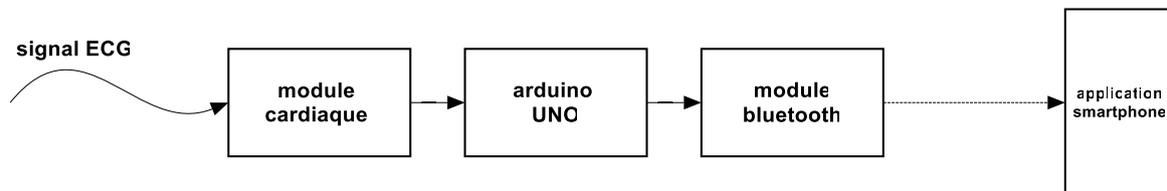


Figure III.2-1 schéma simplifier du système réalisé

L'acquisition du signal ce fait avec le moniteur cardiaque AD8232, pour être analysée par la carte arduino UNO, ensuite les résultats sont envoyés avec le module Bluetooth HC-50 à l'application androïde Bluetooth Terminal/Graphics V2.02. Cette application a plusieurs fonctionnalités notamment la possibilité de tracer plusieurs graphes et d'enregistrer les données reçus en fichier de format .CSV. La figure (III.2-2) montre les interconnexions entre les différents éléments.

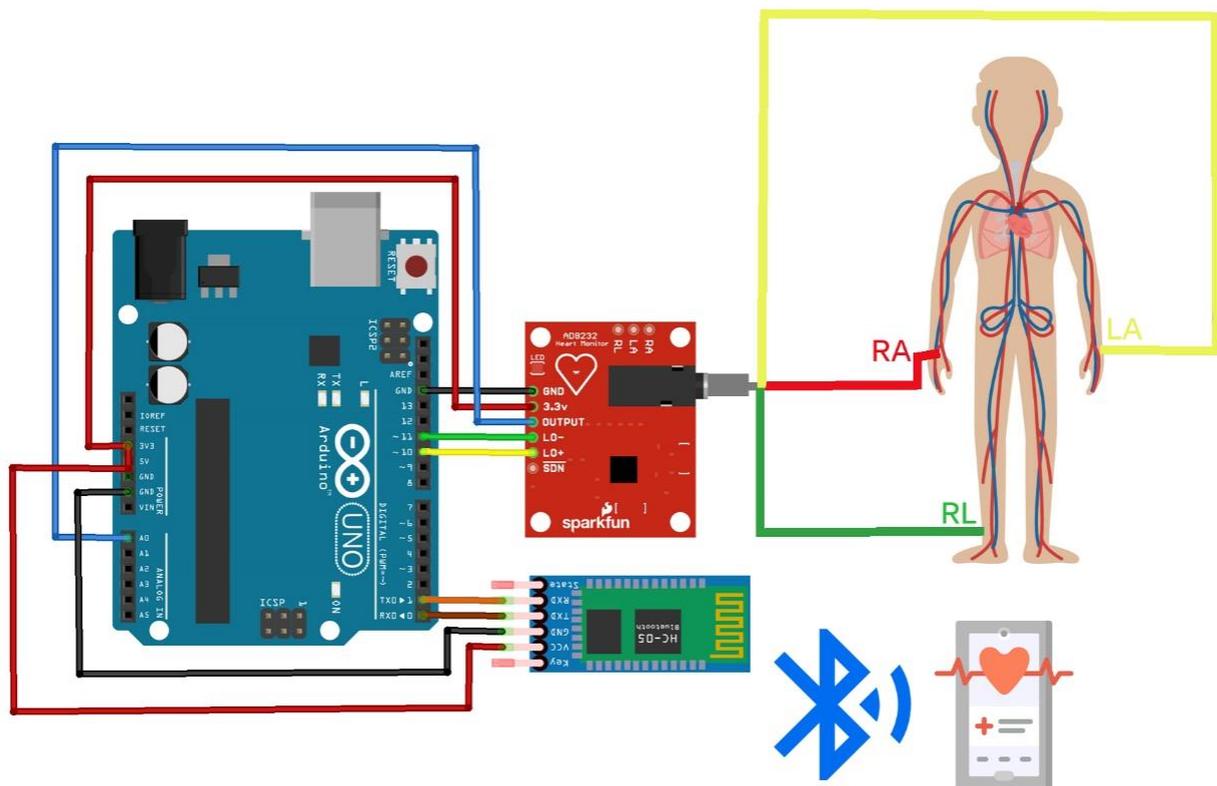


Figure III.2-2 Schéma du montage final

Remarque : puisque la carte arduino est alimentée avec le port USB d'un pc portable il est important de débrancher le pc de son alimentation pour réduire le bruis du secteur.

Le choix des composants a été établi selon plusieurs critères notamment le coût et leur disponibilité.

III.2.1 Carte Arduino UNO

L'Arduino UNO est une carte de développement et de prototypage, basée sur le microcontrôleur ATmega328P, sa configuration compacte la rend très intéressante pour les systèmes embarqués. Son rôle principal est de stoker le programme et de l'exécuter ; comme toutes les autres cartes Arduino, la UNO nous permet de rajouter différents modules externes pour la réalisation de tous types de projet.



Figure III.2-3 Carte Arduino UNO[13].

Ces principales caractéristiques techniques sont résumées dans le tableau suivant :

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
E / S numériques	14 (dont 6 fournissent une sortie PWM)
E / S numériques PWM	6
Pointes d'entrée analogiques	6
Courant CC par broche d'E / S	20 Ma
Courant CC pour la broche de 3.3V	50 Ma
Mémoire flash	32Ko dont 0,5 Ko utilisé par bootloader
SRAM	2 Ko
EEPROM	1 Ko
Vitesse de l'horloge	16 MHz
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

Tableau III.2-1 caractéristiques techniques Arduino UNO [13]

III.2.2 Moniteur de fréquence cardiaque AD8232

Pour la mesure du signal ECG, Nous utilisons un module Arduino destiné à la réalisation de moniteur de fréquence cardiaque (Figure III.2-4). Ce module est basé sur le circuit intégré

AD8232 de conditionnement de signal ECG et d'autres signaux physiologiques. Il est conçu pour amplifier et filtrer de petits signaux [14].

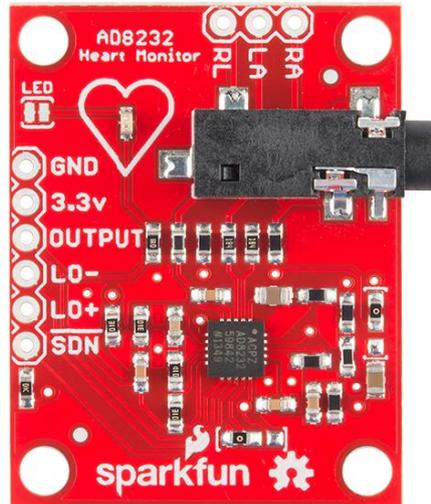


Figure III.2-4 La carte de monitoring de fréquence cardiaque AD8232[14].

L'AD8232 contient principalement un amplificateur d'instrumentation et un étage qui prend en charge le filtrage. La fonction clé dans la chaîne du signal est l'amplificateur d'instrumentation. Les électrodes du corps sont directement connectées aux nœuds d'entrée haute impédance de cet amplificateur. Il est couplé à un filtre passe-haut à 2 pôles, qui filtre les artefacts de mouvement. Le deuxième étage consiste en un AOP qui peut être configuré comme un filtre passe-bas à 2 pôles pour éliminer le bruit de ligne et les autres interférences. Le signal de sortie représente une version filtrée et amplifiée du signal ECG[14].

L'AD8232 possède plusieurs configurations d'utilisation ; celle implémentée par le module utilisé est donné par la figure (III.2-5) :

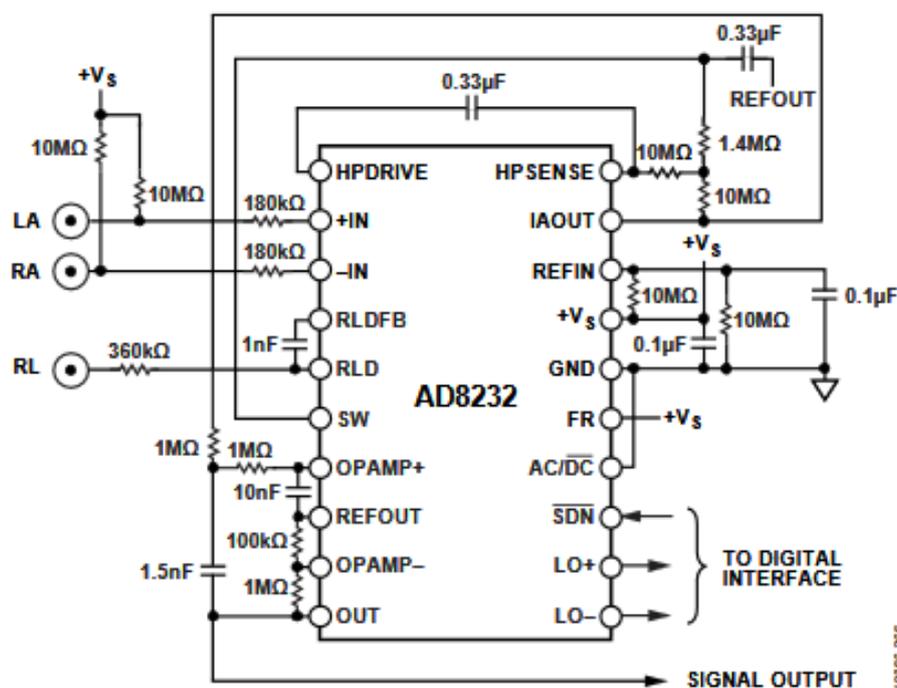


Figure III.2-5 configuration du moniteur de fréquence cardiaque AD8232 [15].

Pour obtenir une forme d'onde ECG avec une distorsion minimale, l'utilisateur doit rester relativement immobile pendant la mesure, et donc, les artefacts de mouvement sont moins importants. l'AD8232 est configuré avec un filtre passe-haut de 0,5 Hz suivi d'un filtre passe-bas de 40 Hz. Une troisième électrode est utilisée pour un rejet optimal en mode commun [15].

Nous connecterons cinq des neuf broches de la carte à la Arduino de la manière suivante :

Nom de la broche	Connexion Arduino
GND	GND
3.3v	3.3v
Output	A0
LO-	11
LO +	10

Tableau III.2-2 raccordement du moniteur cardiaque à la carte Nano [14].

Les mesure son prise en utilisant les câbles indiquer dans la figure (III.2-6), ces câbles dispose d'un code couleur pour qui indique l'emplacement de chaque électrodes



Figure III.2-6 électrode et câble pour moniteur de fréquence cardiaque AD8232 [14]

On place les électrodes Selon le tableau suivant :

Couleur du câble	Signal
Rouge	RA (bras droit)
Jaune	LA (Bras Gauche)
Vert	RL (jambe droite)

Tableau III.2-3 emplacement des électrodes sur le corps [14].Module bluetooth HC-05

le module Hc-05 est un émetteur/récepteur Bluetooth qui permet d'établir une liaison serie entre la carte arduino et un autre appareil (smartphone ; tablette ; ordinateur).

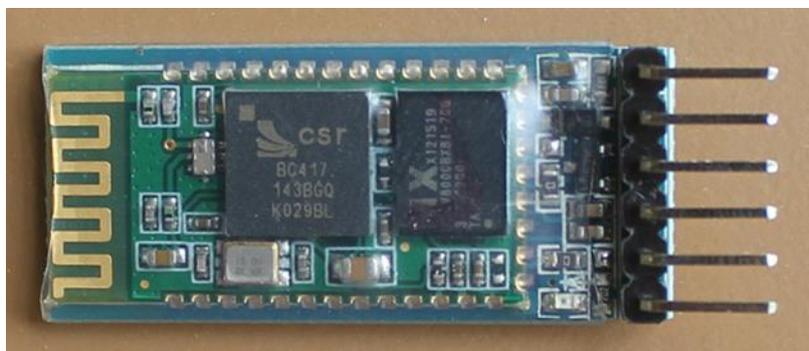


Figure III.2-7 Module bluetooth HC-05[16]

Ces principales caractéristiques techniques sont résumées dans le tableau suivant:

Fonctions	maître / esclave
Tension de fonctionnement	3.3V
Tension d'entrée	3.3 à 6V
Baudrate par défaut	9600 baud
Taille	39 x 15 mm
Brochage 6 pins	Vcc, Gnd, TxD, RxD

Tableau III.2-4 caractéristiques techniques du HC-05[16]

Le raccordement à la carte Arduino est résumé dans le tableau suivant :

Carte Nano	Module HC-05
5V	Vcc
GND	GND
10	RXD
11	TXD

Tableau III.2-5 raccordement module HC-05 à la carte Uno[16].

III.3 Adaptation de l'algorithme de la machine a état pour l'implémentation matérielle

III.3.1 Algorithme en temps réel

Dans cette partie nous exposons le programme réalisé en adaptant l'algorithme de la machine à état présenté dans le chapitre 2 à une utilisation dans un système embarqué et en temps réel. Certaines modifications ont été apportées au programme pour s'adapter aux exigences d'un système embarqué. L'organigramme du programme est présenté par la figure III.6 où les conditions notées (**cond**) font références aux conditions décrites dans le chapitre 2.

III.3.2 Validation des données ECG mémorisées

Dans notre programme l'acquisition et le traitement se font en même temps. Ceci peut engendrer des problèmes notamment la perte de l'information

Nous avons réglé ce problème en utilisant des tableaux qui stockent l'information acquise pour leurs traitements avec un décalage de 2s.

Dans la première phase de test nous utilisons des données utilisées dans le programme sous matlab dans la carte Arduino en stockant les données dans la mémoire programme en utilisant l'instruction PROGMEM.

La figure(III.3-1) illustre le signal ECG mémorisé dans la carte Arduino.

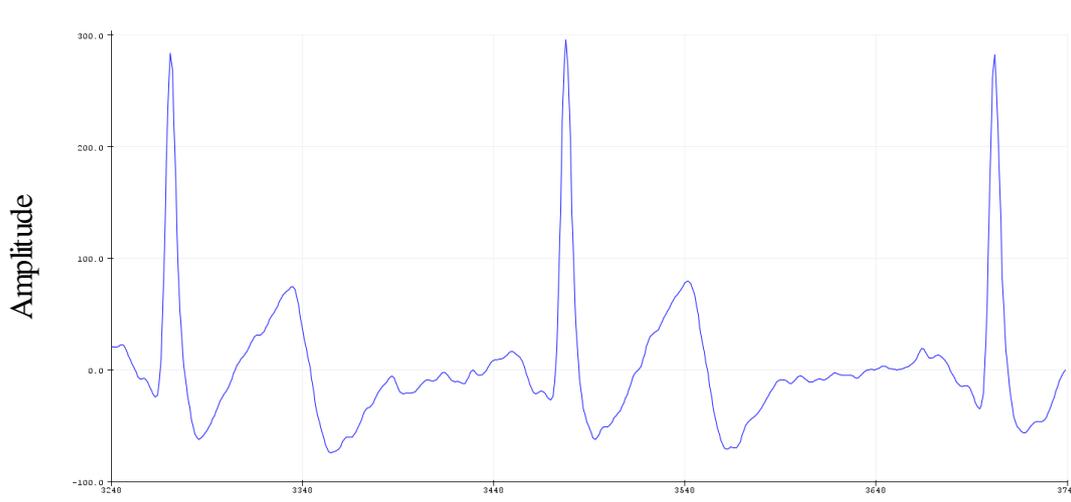


Figure III.3-1 signal ECG généré à partir de la mémoire de l'Arduino[12].

Dans le système à temps réel le temps d'exécution est très important, l'utilisation de l'algorithme de la machine à état nous permet de réduire efficacement celui-ci. En effet vu que la période d'échantillonnage choisie est de 250 Hz, cela veut dire que le temps de traitement doit être inférieur ou égale à 0.04s.

Le temps d'exécution moyen de chaque étape est résumé dans le tableau suivant :

Etape	Temps d'exécution moyen
Calcule des seuils S_DC et S_T	1149 μ s
ETAT0	20 μ s
ETAT1	44 μ s
ETAT2	8 μ s
ETAT3	12 μ s
ETAT4	8 μ s
ETAT5	58 μ s
ETAT6	24 μ s
Le programme	1103 μ s

Tableau III.3-1 temps d'exécution moyen des étapes du programme

En observant ces données ; on remarque que la phase de calcul des seuils est la plus gourmande en terme de temps d'exécution cela s'explique par l'utilisation des opérations arithmétiques (addition et division) avec des tableaux de valeurs.

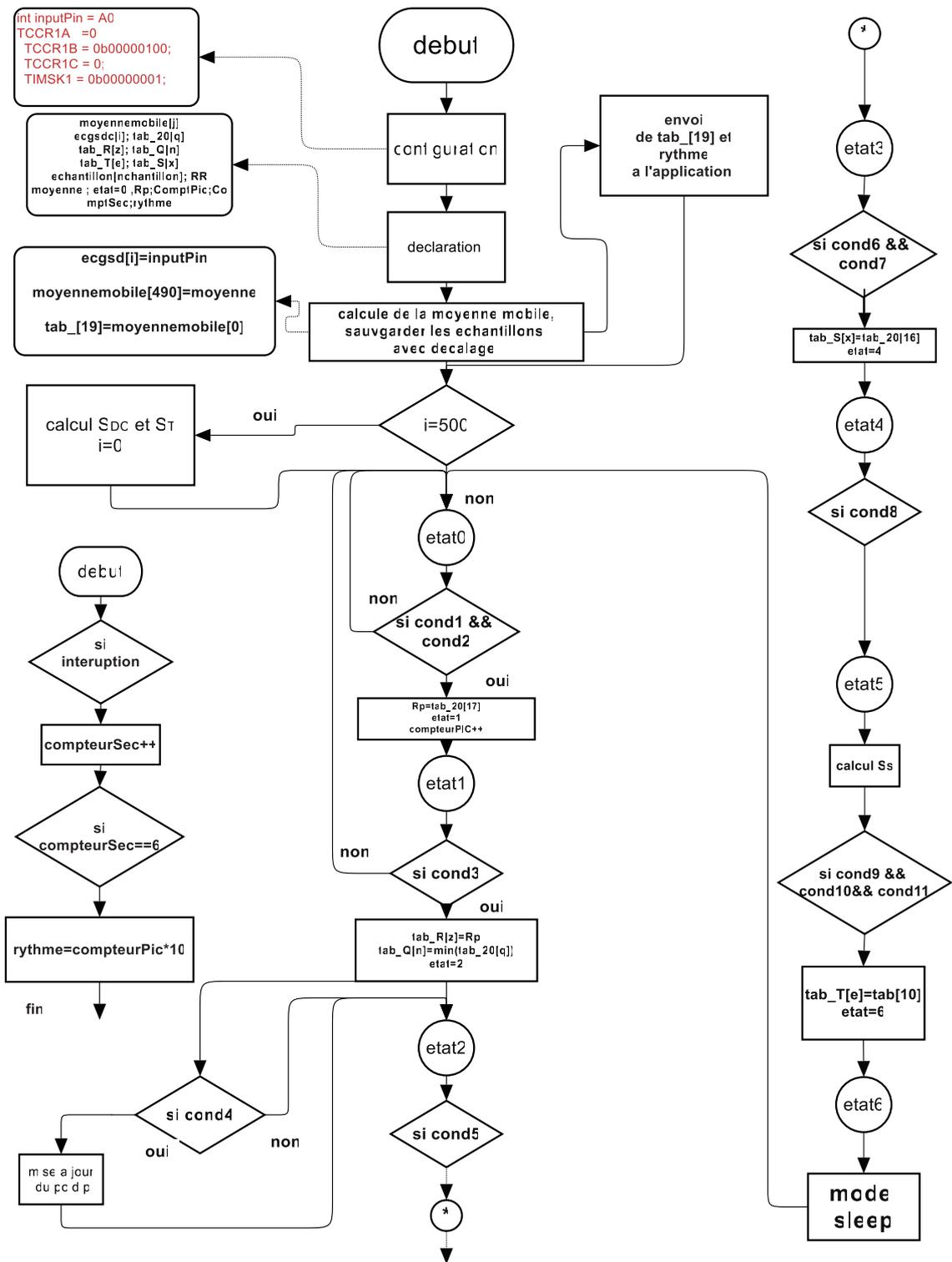


Figure III.3-2 organigramme du programme

III.4 Comparaison des résultats

III.4.1 Comparaison entre les résultats entre Matlab et Arduino

III.4.1.1 ECG réel mémorisé :

Dans cette partie nous comparons entre les résultats obtenus par le traitement de signaux ECG (fréquence d'échantillonnage= 250Hz) avec Matlab et ceux obtenus par le traitement du même signal avec la carte Arduino.

La figure (III.4-1) illustre la variation des états par rapport au signal ECG pendant l'exécution du programme sous Arduino.

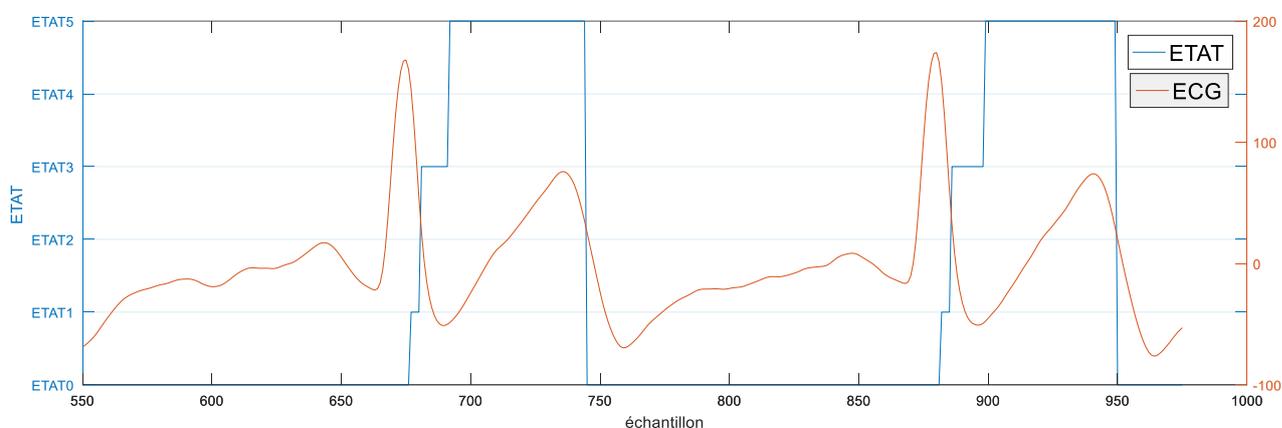


Figure III.4-1 Variation des états par rapport au signal ECG.

La figure (III.4-2) présente les amplitudes du pic R obtenus avec Matlab et Arduino. On remarque que les valeurs des pics R sont très proches.

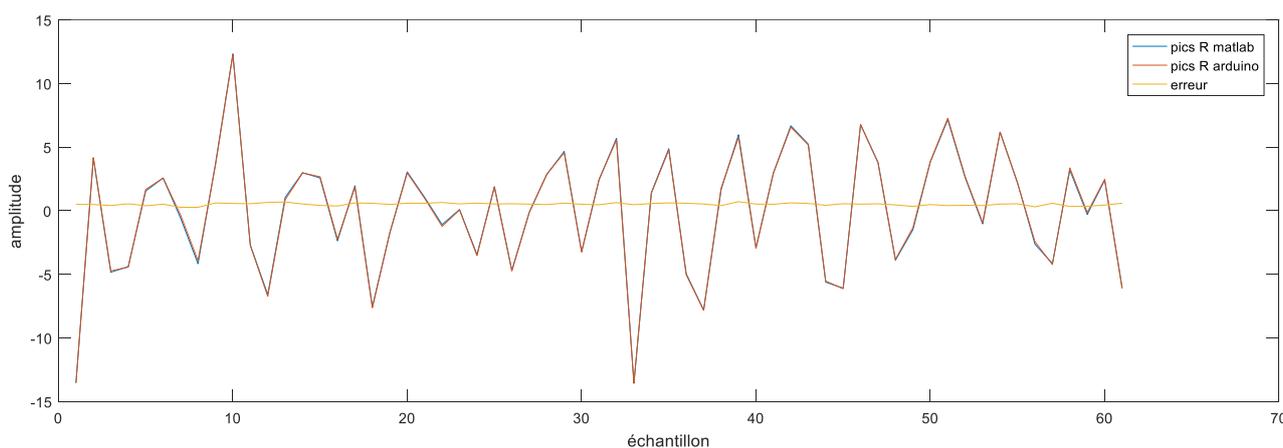


Figure III.4-2 Comparaison entre les pics R détectés avec le programme Arduino et les résultats obtenus sous Matlab

la figure(III.4-3) est une comparaison entre les pics Q détecté avec le programme Arduino et les résultats de Matlab. On remarque que les valeurs des pics ont une erreur relativement grande même si les deux se suivent dans l'allure.

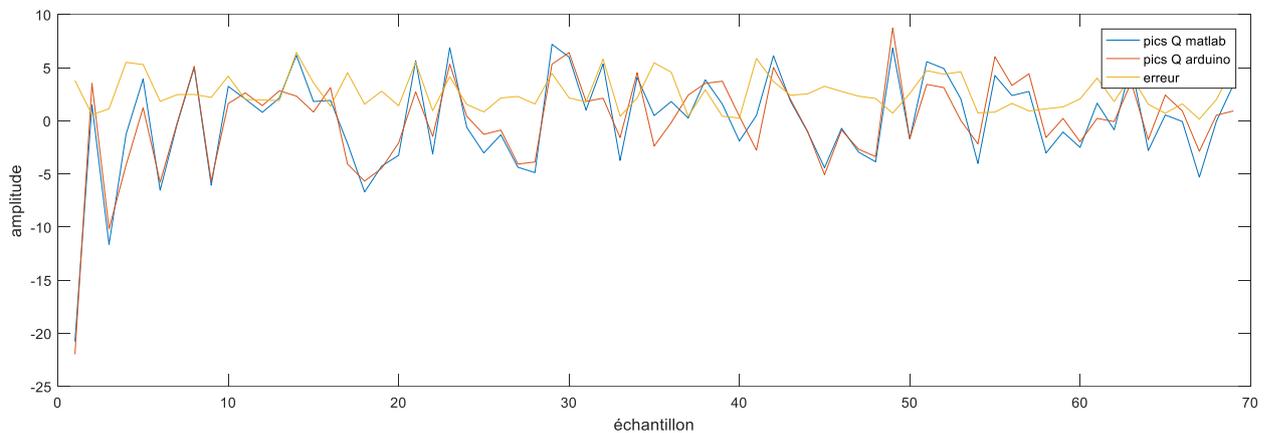


Figure III.4-3 Comparaison entre les pics Q détectés avec le programme Arduino et les résultats de simulation sous Matlab[12]

la figure (III.4-4) est une comparaison entre les pics S détectés avec le programme Arduino et les résultats de Matlab. On remarque que les valeurs des pics S sont très proches

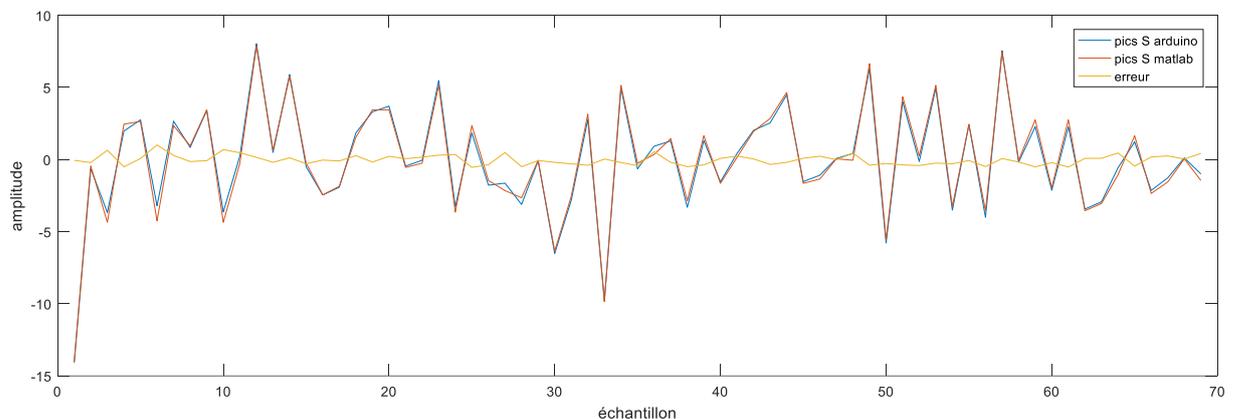


Figure III.4-4 Comparaison entre les pics S détectés avec le programme Arduino et les résultats de simulation sous Matlab

La figure (III.4-5) est une comparaison entre les pics T détectés avec le programme Arduino et les résultats de Matlab. On remarque que les valeurs des pics ont une erreur relativement grande même si les deux se suivent dans l'allure.

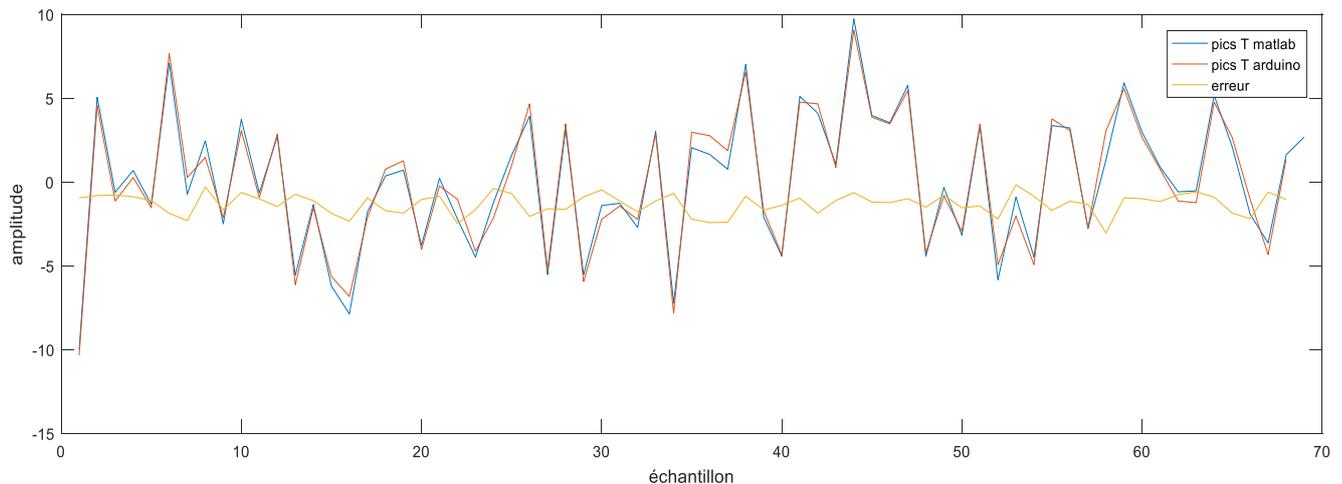


Figure III.4-5 Comparaison entre les pics T détectés avec le programme Arduino et les résultats de simulation sous Matlab

Ces différents résultats nous informent que le programme que nous avons implémenté sous Arduino est relativement efficace pour la détection de pics R et S, cependant pour les pics Q et T, le programme implémenté présente une marge d'erreur assez importante. Dans la suite du travail nous nous intéresserons uniquement à la détection des pics R.

III.4.1.2 ECG réel mesuré

L'acquisition du signal ECG Le signal ECGF représente le signal acquis en utilisant le moniteur de fréquence cardiaque. ce fait selon le programme suivant :

```
void setup() {
  Serial.begin(9600);
  pinMode(10, INPUT); //LO +
  pinMode(11, INPUT); // LO -
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() { // en cas ou une électrode est débrancher
  if ((digitalRead(10) == 1) || (digitalRead(11) == 1)) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  else {
    Serial.println(analogRead(A0));
    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(4); //fréquence 250Hz
}
```

Le signal ECG est acquis toutes les 4ms soit une fréquence d'échantillonnage de 250Hz. La sélection « if » nous permet de vérifier si les électrodes sont bien placées : si l'une des électrodes « + » ou « - » est débranchée l'acquisition est suspendu ; et la LED « L » s'allume pour signaler le défaut. La figure (III.4-6) présente un exemple de signal ECG acquis avec le circuit réalisé. On y distingue clairement les complexes QRS.

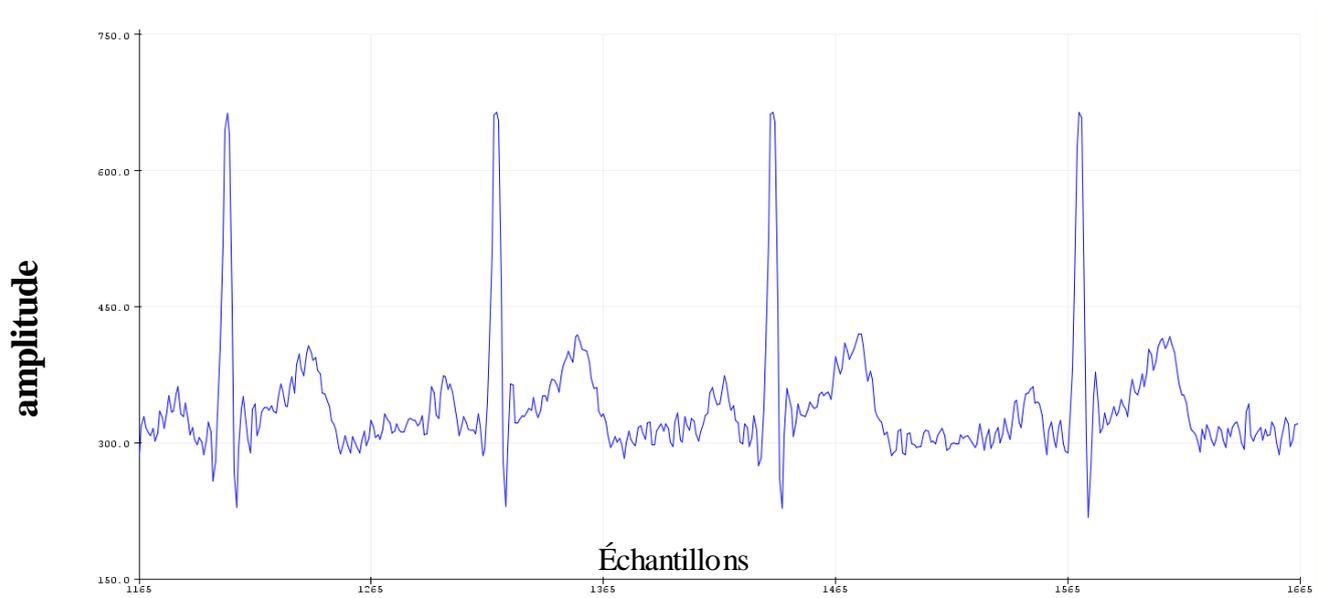


Figure III.4-6 signal acquis par la carte Arduino

La figure (III.4-8) donne les pics R détecté avec le programme Arduino. On remarque que le programme est efficace pour la détection des pics R.

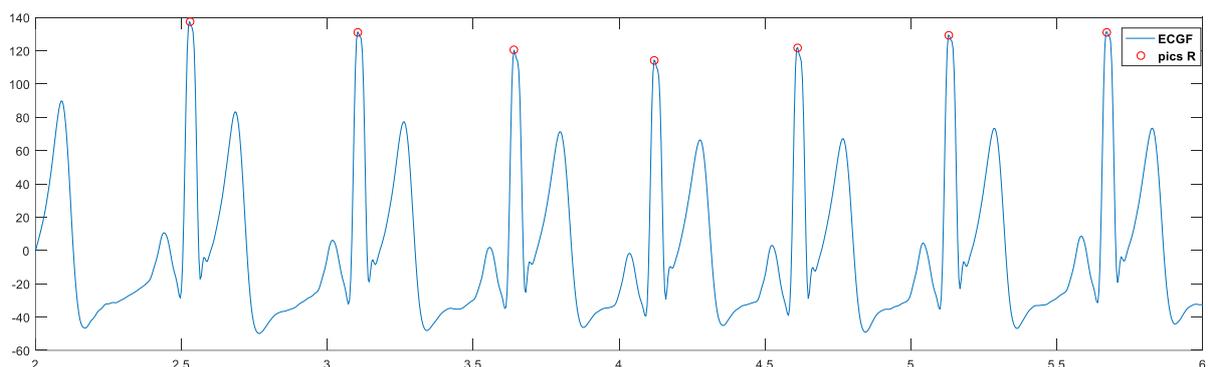
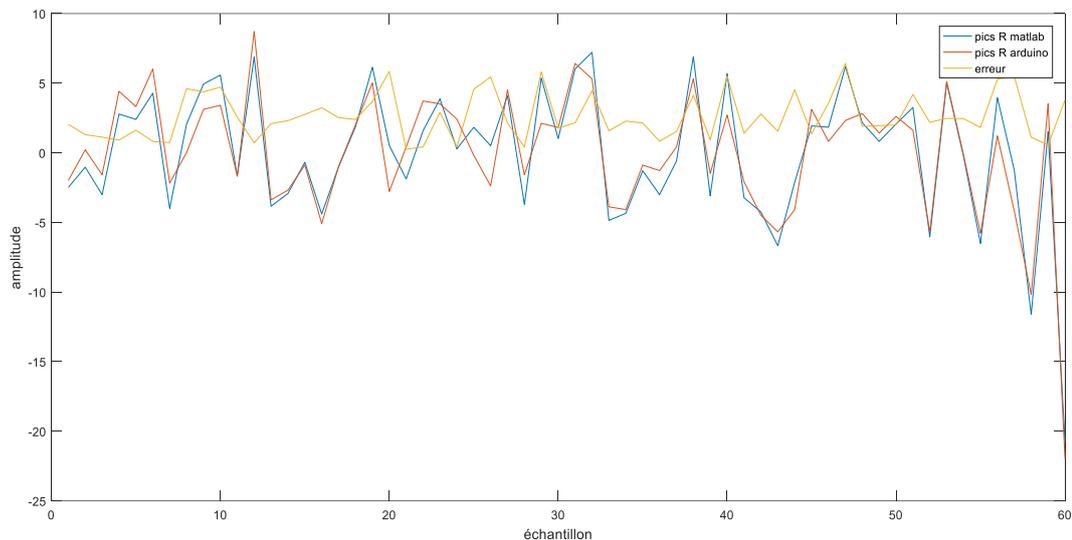


Figure III.4-7 pics R du signal mesuré détecté par le programme Arduino

La figure (III.4-9) est une comparaison entre les pics R détectés avec le programme Arduino et les résultats obtenus par le programme Matlab appliqué au signal acquis. On remarque que même si les allures se suivent la marge d'erreur demeure importante.

Figure III.4-8 Comparaison entre les amplitudes des pics R détectés avec le programme



Arduino et celui de Matlab pour le signal mesuré.

Cette différence impacte la valeur du rythme cardiaque affiché par la carte (70Bpm) et celui calculé avec Matlab (68Bpm).

III.5 Conclusion

Après avoir fait la synthèse des différents résultats abordés dans ce chapitre nous arrivons aux conclusions suivantes :

- Le programme implémenté présente des résultats peu concluants vis-à-vis de la détection des pics Q et T en comparaison aux résultats obtenus avec le programme sous Matlab.
- Le signal acquis par le moniteur cardiaque reste très bruité à cause notamment des bruits de mouvement qui ne sont pas éliminés.
- Les résultats obtenus pour la détection des pics R et S sont acceptables.
- Le rythme cardiaque peut être calculé en utilisant les pics R.

- Le temps d'exécution du programme est très satisfaisant vue qu'il respecte le temps d'échantillonnage.
- Ce travail peut servir pour la visualisation du signal ECG et de la fréquence cardiaque.
- Vue le manque de précision de la détection des pics l'étape de la classification reste impossible à ce stade.



Conclusion Générale

Dans ce travail, nous disposons d'un programme disponible sur le site MATWORK qui est une contribution de Hooman Sedghamizh, après avoir documenté son travail en cherchant à expliquer les choix pris dans son développement, nous avons essayé de l'adapter à une utilisation en temps réel dans un système embarqué ; en comparant les résultats obtenus à ceux du programme de la machine à état nous arrivons aux conclusions suivantes :

Vu le manque de précision de la détection des pics, l'étape de la classification reste impossible à ce stade et par conséquent le diagnostic automatique des pathologies.

Cependant vu que la détection des pics R est acceptable ce travail peut servir pour la visualisation du signal ECG et le calcul de la fréquence cardiaque.

à travers ces constats nous pouvons entrevoir quelques Perspectives futures afin de l'améliorer :

- Ajuster le programme pour avoir de meilleurs résultats pour la détection des pics Q et T, et rajouter la détection des pics P et U.
- Améliorer le signal d'entrée notamment en ajoutant un gyroscope pour éliminer les bruits du au mouvement.
- Faire une étude sur la consommation d'énergie en vu d'intégrer une batterie et améliorée sa portabilité.
- Utiliser des brassards au lieu de patchs pour les électrodes, ils ont l'avantage d'être réutilisables et peu intrusifs
- Développer une application pour smartphone qui permettra d'envoyer toutes les informations directement au médecin pour le suivi du patient
- Réaliser le boîtier externe du dispositif et rajouter des fonctionnalités notamment des boutons de sélection pour changer la fréquence d'échantillonnage ou bien l'information à afficher sur l'écran.

- Intégrer la partie classification au programme principal pour avoir Un pré-diagnostic de l'état du patient

Annexe

Programme de la machine a état[12] :

```

R_i = []; %save index of R wave
R_amp = []; %save amp of R wave
S_i = []; %save index of S wave
S_amp = []; %save amp of S wave
T_i = []; %save index of T wave
T_amp = []; %save amp of T wave
thres_p = []; %for plotting adaptive threshold
buffer_plot = [];
buffer_long = []; % buffer for online processing
state = 0 ; % determines the state of the machine in the algorithm
c = 0; % counter to determine that the state-machine doesnt get stock in T
wave detection wave
T_on = 0; % counter showing for how many samples the signal stayed above T
wave threshold
T_on1=0; % counter to make sure its the real onset of T wave
S_on = 0; % counter to make sure its the real onset of S wave
sleep = 0; % counter that avoids the detection of several R waves in a
short time
S_amp1 = []; % buffer to set the adaptive T wave onset
buffer_base = []; %buffer to determine online adaptive mean of the signal
dum = 0; %counter for detecting the exact R wave
window = round(fs/25); % averaging window size
weight = 1.8; %initial value of the weight
co = 0; % T wave counter to come out of state after a certain time
thres2_p = []; %T wave threshold indices
thres_p_i = []; %to save indices of main thres
S_amp1_i = []; %to save indices of S thres
thres2_p_i = []; %to save indices of T threshold
Q_i = []; % vectors to store Q wave
Q_amp = []; %vectors to store Q wave
n=[];

ecg = ecg (:); % make sure its a vector
ecg_raw = ecg; %take the raw signal for plotting later
time_scale = length(ecg_raw)/fs; % total time;
%Noise cancelation(Filtering)
f1=0.5; %cutoff low frequency to get rid of baseline wander
f2=45; %cutoff frequency to discard high frequency noise
Wn=[f1 f2]*2/fs; % cutt off based on fs
N = 3; % order of 3 less processing
[a,b] = butter(N,Wn); %bandpass filtering
ecg = filtfilt(a,b,ecg);

```

```

buffer_mean=mean(abs(ecg(1:2*fs)-mean(ecg(1:2*fs)))); % adaptive threshold
DC corrected (baseline removed)
buffer_T = mean(ecg(1:2*fs));%second adaptive threshold to be used for T
wave detectio

for i = 1 : length(ecg)
buffer_long = [buffer_long ecg(i)] ; % save the upcoming new samples
buffer_base = [buffer_base ecg(i)] ; % save the baseline samples

if length(buffer_base) >= 2*fs
    buffer_mean = mean(abs(buffer_base(1:2*fs)-mean(buffer_base(1:2*fs))));
    buffer_T = mean(buffer_base(1:2*fs));
    buffer_base = [];
end

    if length(buffer_long)>= window % take a window with length 15 samples
for averaging
        mean_online = mean(buffer_long); % take the mean
        buffer_plot =[buffer_plot mean_online]; % save the processed signal

if state == 0
    if length(buffer_plot) >= 3 %added to handle bugg for now
        if mean_online > buffer_mean*weight && buffer_plot(i-1-window) >
buffer_plot(i-window) %2.4*buffer_mean
            state = 1; % entered R peak detection mode
            currentmax = buffer_plot(i-1-window);
            ind = i-1-window;
            thres_p = [thres_p buffer_mean*weight];
            thres_p_i = [thres_p_i ind];
        else
            state = 0;
        end
    end
end
end
if state == 1 % look for the highest peak

        if currentmax > buffer_plot(i-window)
            dum = dum + 1;
            if dum > 4
                R_i = [R_i ind];%save index
                R_amp = [R_amp buffer_plot(ind)]; %save index
                n=R_amp;
                % Locate Q wave

%
                [Q_tamp Q_ti] = min(buffer_plot(ind-
round(0.040*fs):(ind)));
                Q_ti = ind-round(0.040*fs) + Q_ti -1;
                Q_i = [Q_i Q_ti];
                Q_amp = [Q_amp Q_tamp];

```

```

        if length(R_amp) > 8
            weight = 0.30*mean(R_amp(end-7:end)); %calculate the 35% of
the last 8 R waves
            weight = weight/buffer_mean;
            end
            state = 2; % enter S detection mode state 2
            dum = 0;
            end
        else
            dum = 0;
            state = 0;
            end
    end

    end
if state == 2
    if mean_online <= buffer_mean % check the threshold
        state = 3; %enter S detection
    end
end
if state == 3
    co = co + 1;

    if co < round(0.200*fs)
        if buffer_plot(i-window-1) <= buffer_plot(i-window) % see when
the slope changes
            S_on = S_on + 1; % set a counter to see if its a real change
or just noise
            if S_on >= round(0.0120*fs)
                S_i = [S_i i-window-4];%save index of S wave
                S_amp = [S_amp buffer_plot(i-window-4)];%save index
                S_amp1 = [S_amp1 buffer_plot(i-window-4)]; %ecg(i-4)
                S_amp1_i = [S_amp1_i ind]; %index of S_amp1_i
                state = 4; % enter T detection mode
                S_on = 0;
                co = 0;
            end
        end
    else
        state = 4;
        co = 0;
    end
end
if state == 4
    if mean_online < buffer_mean % see if the signal drops below mean
        state = 6; % confirm
    end
end
if state ==6
    c = c + 1; % set a counter to exit the state if no T wave detected
after 0.3 second
    if c <= 0.7*fs
        % set a double threshold based on the last detected S wave and
% baseline of the signal and look for T wave in between these
% two threshold
        thres2 = ((abs(abs(buffer_T)-abs(S_amp1(end))))*3/4 +
S_amp1(end));
        thres2_p =[thres2_p thres2];
        thres2_p_i =[thres2_p_i ind];

```

```

        if mean_online > thres2
            T_on = T_on +1; % make sure it stays on for at least 3
samples
            if T_on >= round(0.0120*fs)
                if buffer_plot(i-window-1)>= buffer_plot(i-window)
                    T_on1 = T_on1+1; % make sure its a real slope change
                    if T_on1 > round(0.0320*fs)
                        T_i = [T_i i-window-11];%save index of T wave
                        T_amp = [T_amp buffer_plot(i-window-11)];%save index
                        state = 5; % enter sleep mode
                        T_on = 0;
                        T_on1 = 0;
                    end
                end
            end
            end
            end

        else
            state= 5; % enter Sleep mode
        end

    end

    if state==5
        sleep =sleep+c+1;
        c = 0;
        if sleep/fs >= 0.400
            state = 0;
            sleep = 0;%look for the next peak
        end
    end

    % update the online buffer by removing the oldest sample
    buffer_long(1)=[];
end
end
R_R = diff(R_i); % calculate the distance between each R wave
heart_rate=length(R_i)/(time_scale/60); % calculate heart rate
msgbox(strcat('Heart-rate is= ',mat2str(heart_rate)));
% compute the min max R-R wave
max_R_interval = max(R_R);
min_R_interval = min(R_R);
% detect arhythmia if there is any irregularity
% if (max_R_interval/fs)-(min_R_interval/fs) > 0.16
%     msgbox('Irregular Rhythm','Arhythmia Detected');
% end

```

Programme arduino implementer :

```

const int nEchantillons = 10;
float S_T = 0, S_DC = 0, S_S = 0, as = 0, tau = 0, tot = 0, Rp = 0, poids =
1.8 , somlast8R = 0, ECGi = 0;
float moyennemobile[491], ecgsdc[500], tab_20[20], tab_R[70], tab_Q[70],
tab_T[70], tab_S[70], minn = 0, Fc = 0;
int echantillon[nEchantillons];
int indice = 0, taille = 0, etat = 0, sleep = 0, first = 0, rythme = 0;
float total = 0, p = 0;
float moyenne = 0;
int i = 0, j = 0, m = 500, z = 0, h = 0, ir = 0, n = 0, co = 0, so = 0, to
= 0, x = 0, y = 0, g = 0, g1 = 0, e = 0, q = 0, u = 0, d = 0, lo = 0,
ComptPic = 0, ComptSec = 0, pp = 0;
double RR = 0;
unsigned wx = 0, wx1 = 0;
int inputPin = A0;
void setup() {
    TCCR1A = 0; //default
    TCCR1B = 0b00000100; // clk/256 est incrémenté toutes
les 16uS
    TCCR1C = 0;
    TIMSK1 = 0b00000001; // TOIE2
    sei();
    Serial.begin(9600);
    for (int v = 0; v < nEchantillons; v++) {
        echantillon[v] = 0;
    }
}
void loop() {
    wx = micros(); wx1 = wx;
    total = total - echantillon[indice];
    echantillon[indice] = inputPin ;
    ecgsdc[i] = inputPin;
    ECGi = ecgsdc[i];
    //////////////////////////////////////////calcul de S_T et
S_DC//////////////////////////////////////
    ///
    if (i < m) {
        tot = tot + ECGi;
        as = ECGi - S_T;
        tau = tau + abs(as);
    }
    else {
        S_T = tot / m;
        S_DC = tau / m;
        tot = 0;
        tau = 0;
        i = 0;
    }
    total = total + echantillon[indice];
    indice++;
    i++;
    if (indice >= nEchantillons) {
        indice = 0;
        first = 1;
    }
    moyenne = total / nEchantillons;
}

```

```

if (first == 1) {
  for (j = 1; j <= 490; j++) {
    moyennemobile[j - 1] = moyennemobile[j];
  }
  moyennemobile[490] = moyenne;
  ////////////////////////////////////////envoi des
données a l'application
  Serial.print("E");
  Serial.print(rythme);
  Serial.print(",");
  Serial.print(moyennemobile[0]);
  Serial.print("\n");
  ////////////////////////////////////////
  for (q = 1; q <= 19; q++) {
    tab_20[q - 1] = tab_20[q];
  }
  tab_20[19] = moyennemobile[0];
}
//////////////////////////////////////
//////////////////////////////////////ETAT0:detection du pic
R//////////////////////////////////////
//////////////////////////////////////
  if (etat == 0) {
    if (k >= 500) {
      if ((tab_20[19] > S_DC * poids) && tab_20[17] > tab_20[18]) { //le
premier et le deuxieme //
        Rp = tab_20[17];
//*****
*****
        etat = 1;
      } else {
        etat = 0;
      }
    }
  }
  ////////////////////////////////////////
  ////////////////////////////////////////ETAT1:detection du pic Q et confirmation du pic
R//////////////////////////////////////
  if (etat == 1) {
    if (Rp > tab_20[18]) { //confirmation du pic R//
      h++;
      if (h > 4) {
        tab_R[z] = Rp;
        pp++; z++;
        if (pp >= 8) {
          for (int i = z; i != z - 8; i--) {
            somlast8R = somlast8R + tab_R[i];
          }
          poids = 0.3 * (somlast8R / 8) / S_DC;
          somlast8R = 0;
          pp = 0;
        }
        ComptPic++;
        h = 0;
        etat = 2;
        minn = tab_20[0]; //detection pic Q//
        for (d = 1; d <= 19; d++) {
          if (minn > tab_20[d]) {

```

```
        minn = tab_20[d];
    }
}
tab_Q[n] = minn;

    n++;
}
}
else {
    etat = 0;
    h = 0;
}
}
//
//
//ETAT2:verification//
//
if (etat == 2) {
    /
    if (tab_20[19] <= S_DC) {
        etat = 3;
    }
}
//
//
//ETAT3:detection du pic
S//
//
if (etat == 3) {
    co++;
    if (co < 100) {
        if (tab_20[18] <= tab_20[19]) {
            so++;
            if (so >= 3) {
                tab_S[x] = tab_20[16];
                etat = 4;
                so = 0;
                co = 0;
                x++;
            }
        }
    } else {
        etat = 4;
        co = 0;
    }
}
//
//
//ETAT4:verification//
//
if (etat == 4) {

    if (tab_20[19] < S_DC) {
        etat = 5;
    }
}
}
```

```

//ETAT5:detection du pic T
if (etat == 5) {
  to++;
  if (to <= 175) {
    S_S = ((abs(abs(S_T) - tab_S[x])) * 3 / 4 + tab_S[x]);
    if (tab_20[19] > S_S) {
      g++;
      if (g > 3) {
        if (tab_20[18] >= tab_20[19]) {
          g1++;
          if (g1 > 8) {
            tab_T[e] = tab_20[10];
            e++;
            etat = 6;
            g = 0;
            g1 = 0;
            to = 0;
          }
        }
      }
    }
  } else {
    etat = 6;
    to = 0;
  }
}
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////ETAT6:sleep//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
if (etat == 6) {
  //wx = micros();
  sleep = sleep + to + 1;
  to = 0;
  etat = 0;
  if (sleep / m >= 0.4) {
    sleep = 0;
    etat = 0;
  }
}
wx = micros() - wx;
delayMicroseconds(4000 - wx);
}
ISR (TIMER1_OVF_vect)
{ //Serial.println(ComptSec);
  ComptSec++;
  if (ComptSec == 6)
  {
    rythme = ComptPic * 10;
    ComptPic = 0;
    ComptSec = 0;
  }
}
}

```

Liste des références

- [1] F. BOUSEFSAF « mesure sans contact de l'activité cardiaque par analyse du flux vidéo issu d'une caméra numérique » THÈSE de doctorat, UNIVERSITÉ DE LORRAINE, 2014.
- [2] D. FARGE, J. DERANLOT « e-semio le livre de la semiologie médicale », [<http://www.e-semio.org/>], 2012 [<http://www.e-semio.org/Relation-entre-activite-electrique>] (date de consultation 15/04/2018)
- [3] N. DOBIGEON « Projet Traitement du Signal : Analyse de Signaux ECG » cour module d'application MODAP Signal/Image, UNIVERSITÉ DE Toulouse, 2011
- [4] A. LOUCHICHE, M. KERBOUB « étude et classification des signaux ECG » mémoire d'ingénieur. UNIVERSITÉ DE BEJAIA .2009
- [5] D. ATTIAS, N. LELLOUCHE « ikb CARDIOLOGIE VASCULAIRE » Edition VG, 7^e édition 2016.
- [6] H. DEGHILOUS « Méthodes Supervisées et Non Supervisées en Classification des Signaux ECG » mémoire master. UNIVERSITÉ DE BEJAIA 2016.
- [7] M. HADJ SLIMANE « analyse et synthèse de méthodes de détection du complexe QRS du Signal ECG » THÈSE de doctorat, UNIVERSITE TLEMCEM, 2006.
- [8] R. BENALI « Analyse du signal ECG par réseau adaptif d'ondelettes en vue de la reconnaissance de pathologies cardiaques » THÈSE de doctorat, UNIVERSITE TLEMCEM, 2013
- [9] P. TABOULET « ECG de A à Z », [<http://www.e-cardiogram.com>] , 2013, [http://www.e-cardiogram.com/Technique/filtre_ECG] (date de consultation 15/04/2018) .
- [10] M. OLTEAN « TRAITEMENT AUTOMATIQUE DES SIGNAUX ECG », RAPPORT DE STAGE, 2005, Ecole Nationale Supérieure de Télécommunications de Bretagne
- [11] J. PAN, W. J. TOMPKINS, « A real time QRS detection algorithm », IEE Trans. Biomed. 1985444
- [12] H. SEDGHAMIZ « ECG Q R S wave online detector » ; [www.mathworks.com]; 2014.
- [13] ARDUINO [<https://www.arduino.cc/>] , 2018 (date de consultation 28/05/2018) .
- [14] SPARKFUN ELECTRONICS [<https://www.sparkfun.com/products/12650>] (date de consultation 28/05/2018)
- [15] ANALOG DEVICES « AD8232 DATA Sheet » 2013.
- [16] GOTRONIC [<https://www.gotronic.fr>] 2015 (date de consultation 28/05/2018)

Le Résumé

Ce travail vise à réaliser un d'un dispositif portable dont la fonction est de détecter le signal ECG qui est le signal physiologique lié à l'activité cardiaque afin de le traiter et d'extraire des informations pertinentes concernant les paramètres du rythme cardiaque qui permettront de détecter certaines pathologies.

Mots clés : système embarqué ; signal ECG ; complexe QRS ; rythme cardiaque ; pic R ; carte arduino

Abstract

This work aims to realize one of a portable device whose function is to detect the ECG signal which is the physiological signal related to the cardiac activity in order to treat it and to extract relevant information concerning the parameters of the cardiac rhythm which will allow to detect certain pathologies.

Keywords: embedded system; ECG signal; QRS complex; heartbeat ; peak R; arduino card.

الخلاصة

يهدف هذا العمل إلى تحقيق جهاز محمول وظيفتها للكشف عن إشارة ECG وهو إشارة الفسيولوجية المرتبطة بالنشاط القلب لعلاج واستخراج المعلومات ذات الصلة المتعلقة المعلومات من معدل ضربات القلب التي من شأنها تكشف عن بعض الأمراض.

كلمات البحث: نظام مضمن؛ إشارة تخطيط القلب ; مجمع QRS ; معدل ضربات القلب ذروة R ؛ بطاقة اردوينو.