

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE**
Domaine : **Sciences et Technologies**
Filière : **Génie électrique**
Spécialité : **Electronique Biomédicale**

Présenté par
**CHIOUKH ABDELAZIZ
CHOUGAR RACHID**

Thème
Conception et réalisation d'un stéthoscope Numérique avec Arduino

Mémoire soutenu publiquement le 11 juillet 2017 devant le jury composé de :

Mr Lazri M

Maître de Conférences à l'UMMTO Président

Mr Oualouche f

Maître de Conférences à l'UMMTO examinateur

Mr Laghrouche M

Professeur à l'UMMTO Promoteur

Mr Nachef M

Co-Promoteur

Promotion : 2016/2017

DEDICACE

Je rends grâce à dieu de m'avoir donné le courage et la volonté ainsi que la conscience d'avoir pu terminer mes études.

Je dédie ce modeste travail à :

- *A Ma très chère mère, et mon très cher père, à leur grand sacrifice et le dévouement pour mon bonheur.*
- *A Mes frères et sœurs, et toute ma famille.*
- *A mes amis et toute personne qui ma aidée de loin ou de près.*
- *A toute ma promotion: électronique*

CHIOUKH ABDELAZIZ



DEDICACE

Je rends grâce à dieu de m'avoir donné le courage et la volonté ainsi que la conscience d'avoir pu terminer mes études.

Je dédie ce modeste travail à :

- *A Ma très chère mère, et mon très cher père, à leur grand sacrifice et le dévouement pour mon bonheur.*
- *A Mes frères et sœurs, et toute ma famille.*
- *A mes amis et toute personne qui ma aidée de loin ou de près.*
- *A toute ma promotion: électronique*

CHOUGAR Rachid



Remerciement

Nous remercions tout d'abord, Allah qui nous a donné la force et le courage afin de parvenir à élaborer ce modeste travail.

Tout d'abord, ce pfe ne serait pas aussi riche et n'aurait pu voir le jour sans l'aide et l'encadrement de notre encadreur Mr LAGHROUCHE MOURAD nous le remercions pour la qualité de son encadrement, sa rigueur et sa disponibilité, ses remarques fructueuses et ses directives précieuses, qui ont contribué efficacement à l'avancement de ce travail.

Nous adressons de même nos remerciements aux membres du jury et au président du jury qui a accepté les rôles d'examineurs et surtout à cette circonstance se fit un grand honneur pour nous.

Nous adressons en particulier nos remerciements à Mr NACHEF de nous avoir accordé de leur temps avec gentillesse et sympathie permanente et nous en serons très reconnaissants.

Nous profitons à cette opportunité d'exprimer notre gratitude à tous nos enseignants qui ont contribué par leur collaboration, disponibilité et sympathie durant toute notre formation.

Sommaire

Introduction générale	1
-----------------------------	---

Chapitre 1 : Phonocardiographie et bruits cardiaques

Introduction	3
1.2 Anatomie du Cœur	3
1.3 Révolution cardiaque.....	6
1.3.1 Contraction ventriculaire.....	6
1.3.2 Ejection ventriculaire.....	6
1.3.3 Relaxation.....	7
1.3.4 Remplissage.....	7
1.4 Description du Signal Phonocardiographique.....	8
1.4.1. Bruit B1	9
1.4. 2. Bruit B2	10
1.4. 3. Bruit B3 et B4.....	10
1.4.4 Altération du fonctionnement normal du cœur	11
1.5caractéristique temporelles et fréquentielles du signal PCG.....	13
Conclusion	16

Chapitre 2 : Etude de stéthoscope électronique

Introduction.....	17
2 .1Le Capteur.....	18
2.1.1 Le Microphone a contact (accéléromètre piézoélectrique).....	18
2.1.2 Microphone Dynamique	19
2.1.3 Le Microphone à Condenseur.....	20
2.1.4 Le microphone Electret (ECM)	20
2.2.5 Capteur	21
2.2Le conditionnement du signal	23
2.3 Description de la carte Arduino Uno.....	23

2.4 Alimentation de la carte Arduino	24
2.5 Protection du port USB contre la surcharge en intensité	25
2.6 Gestion des memoires dans la carte Arduion	25
2.6.1 Les entrées/sorties numérique.....	26
2.6.2 Les entrées Analogiques	27
2.7 Arduino et la communication avec l'exterieur	28
2.7.1 Dimension de la carte.....	28
2.7.2 Le microcontrôleur ATmega 3	28
2.8 Le module EASY VR.....	29
2.8.1 Description de produit Easy vr	29
2.8.2 Fonctionnalités Easy vr.....	30
2.8.3 Numéro de connecteur nom type description	31
2.8.4 Caractéristique électrique.....	32
2.8.5 Caractéristique de Microphone	32
2.8.6 Le contact entre le module Easy vr et le microcontrolleur	36
2.9 Circuit de sélection du mode de démarrage	37
Conclusion.....	38

Chapitre 3 : Réalisation pratique du stéthoscope électronique

Introduction	40
3.1 Environnement de programmation Arduino	40
3.2 Description de logiciel ARDUINO	40
3.3 DESCRIPTION DE LA BARRE DES BOUTONS	42
3.4 DESCRIPTION DE LA BARRE DES MENUS	43
3.5 Les étapes de téléversement d'un programme vers la carte ARDUINO	43
3.6 Syntaxe du langage Arduino	44
3.6.1 Commandes de structure du programme	44
3.7 Le circuit réalisé	48
3.8 Circuit réalisée avec stéthoscope	48
3.9 Test de validation de notre réalisation	49
3.10 EASY VR Cammander	49
3.10.1 Vérification EASY VR Commander	50
3.10.2 Création de trois commandes dans un groupe.....	50

3.10.3 Le code de la commande pour arduino	51
3.10.4 Enregistrement de code comme arduino	51
3.11. Résultats et interprétation	51
Conclusion	53
Conclusion générale	54

Liste des figures

Chapitre 1 : Phonocardiographie et bruits cardiaques

1.1 Un stéthoscope (a) ,(b)	3
1.2 Description anatomique du myocarde	4
1.3 Coupes longitudinales et transversales d'une valvule cardiaque dans les positions (a) ouverte et (b)	5
1.4 Les Événements et les changements qui se produisent au cours d'un cycle cardiaque	8
1.5 Signal Phonocardiographique normal	9
1.6 Relation entre les bruits cardiaque et la contraction cardiaque	11
1.7 Signal PCG avec la présence d'un souffle diastolique	12
1.8 Valvule aortique dans les cas sténose et normal	12
1.9 Sténose au niveau de l'artère en (a) et rétrécissement de l'aorte en (b)	13
1.10 Spectre des bruits cardiaques	15

Chapitre 2 : Etude de stéthoscope électronique

2.1 Le schéma bloc du Stéthoscope Numérique.....	17
2.2 Un microphone piézoélectrique	19
2.3 Le microphone dynamique	19
2.4 Le microphone à condensateur	20
2.5 Le microphone à electret	21
2.6 Photo du capteur ECM	22
2.7 Les éléments nécessaires pour le capteur des bruits cardiaques.....	22
2.8 Capteur réalisé	22
2.9 Description de la carte ARDUINO.....	24
2.10 Brochage de la carte ARDUINO.....	27
2.11 Microcontrôleur ATmega328	29
2.12 Le module EASY VR.....	30
2.13 Datasheet EASY VR.....	31
2.14 EASY VR avec le microcontrôleur	36

Chapitre 3 : Réalisation pratique du stéthoscope électronique

3.1 Les différentes parties de la fenêtre principale de logiciel Arduino	40
3.2 Fenêtre associée au moniteur série	41
3.3 Structure de la barre des boutons.....	42
3.4 Branchement de module EASY VR avec la carte Arduino Uno	48
3.5 Circuit réalisée avec Stéthoscope	48
3.6 Test de validation de notre système	49
3.7 L'allure du signal obtenu visualisé sur oscilloscope.....	52
3.8 Signal PCG détecté par notre stéthoscope Numérique.....	52

Liste des tableaux

1.1. Les caractéristiques temporelles et fréquentielles des bruits cardiaques.....	14
2.1. Comparaison entre les caractéristiques des microphones.....	21

Introduction générale

Le signal Phono cardiographique (PCG) est riche en informations en relation avec l'hémodynamique cardiovasculaire moins perceptible par l'oreille humaine, notamment pour la détection des souffles diastoliques de faible intensité.

L'examen d'auscultation des bruits cardiaques est un outil fondamental dans le diagnostic des maladies cardiaques. Il est considéré comme la technique la plus couramment utilisée comme un outil de diagnostic primaire et dans les cas d'indisponibilité des équipements médicaux sophistiqués (comme dans les pays sous-développés).

Cependant, la détection des symptômes pertinents constituant un diagnostic basé sur les bruits cardiaques auscultés dans un stéthoscope est une tâche compliquée et difficile pour les médecins et qui demande des années d'entraînement pour à l'acquérir et l'affiner.

Une partie de cette difficulté provient du fait que les bruits cardiaques sont souvent séparés les uns des autres par moins d'une période de 30 millisecondes. En outre, les bruits cardiaques caractérisant les troubles cardiaques ont généralement beaucoup moins d'énergie que les sons cardiaques normaux. Cela rend la tâche de détection acoustique plus difficile pour le médecin.

De plus, même si la capacité d'effectuer l'auscultation est acquise par un médecin, il n'existe pas de manière organisée pour faire apprendre cette compétence aux autres médecins notamment les médecins stagiaires en apprentissage.

Cela constitue un défi supplémentaire pour apprendre à écouter les bruits du cœur.

Il serait intéressant que les avantages de l'auscultation puissent être illustrés sur un tracé graphique en utilisant un équipement qui est peu coûteux, robuste et facile à utiliser.

Les stéthoscopes souvent utilisés par les médecins sont des dispositifs acoustiques purement passifs qui permettent d'ausculter les bruits générés par le cœur. La simplicité de ces dispositifs est éclipsée par une qualité sonore peu convaincante. Ces dispositifs sont

Introduction générale

également difficiles à s'interfacer avec des technologies modernes telles que des ordinateurs pour enregistrer et analyser les bruits cardiaques.

Le but de ce travail est de concevoir et développer un stéthoscope numérique de moindre coût et qui a une meilleure réponse acoustique, et en plus peut s'interfacer avec les technologies modernes mieux que le stéthoscope acoustique ordinaire.

Le manuscrit de ce mémoire de projet de fin d'étude suit cette organisation :

Dans le Chapitre I, nous présentons un aperçu sur l'origine des bruits cardiaques en décrivant l'hémodynamique cardiaque. En vue de quantifier l'apport de chacune des valvules, et d'avoir une brève description sur la nature et l'allure du signal des bruits cardiaques connu par le signal phono cardiogramme PCG.

Dans le deuxième chapitre, nous présentons les différentes parties utiles pour la conception de notre stéthoscope numérique ainsi que les résultats de tests de chaque circuit pour un choix judicieux de chaque circuit.

Un troisième et dernier chapitre qui présente le circuit finale de notre stéthoscope numérique avec les différents programmes et logiciels que nous avons développé et qui permettent l'acquisition, l'enregistrement et le traitement numérique du signal phono cardiographique PCG acquis a travers notre circuit.

Chapitre 1

Phonocardiographie et bruits cardiaques

Introduction

L'auscultation des bruits générés par le cœur au moment de ses contractions est l'une des méthodes les plus couramment utilisées par les médecins pour diagnostiquer les maladies cardiovasculaires et respiratoires. L'outil d'auscultation le plus répandue dans le domaine cardiologique est **le stéthoscope**.

Un médecin expérimenté peut diagnostiquer un large éventail d'anomalies cardiovasculaires, y compris les sténoses mitrale et aortique ainsi que les souffles systoliques, mais de nombreuses anomalies sont souvent manquées en raison d'une incapacité à appliquer une écoute sélective parmi les différentes composantes qui accompagnent le cycle cardiaque, ou une incapacité naturelle à détecter les fréquences en dehors de la gamme de l'audition humaine.



(a)



(b)

Figure 1.1 : (a),(b) : Un Stéthoscope

La phonocardiographie offre une visualisation graphique des signaux d'auscultation, permettant l'observation clinique des bruits cardiaques caractérisés par des fréquences dehors de la plage normale de l'audition humaine.

1.2. Anatomie du Cœur

Le cœur est l'organe le plus important dans le corps humain. Il est considéré comme une pompe musculaire subdivisée en quatre cavités : deux oreillettes et deux ventricules (Figure 1.2), permettant de propulser le sang vers toutes les cellules du corps humain.

L'oreillette droite récupère le sang pauvre en oxygène des veines cave supérieure et inférieure, et le propulse après contraction vers le ventricule droit. Cette phase est la systole auriculaire. Cette phase systolique est suivie d'une autre diastolique, permettant d'envoyer le sang dans les poumons à travers les artères pulmonaires. Le dioxyde de carbone véhiculé par ce sang sera évacué par les poumons vers l'extérieur du corps, et par la suite sera rechargé en oxygène. Le sang revient ensuite au cœur dans l'oreillette gauche, puis passe dans le ventricule gauche là où il est envoyé vers les organes dans le réseau artériel à travers l'aorte.

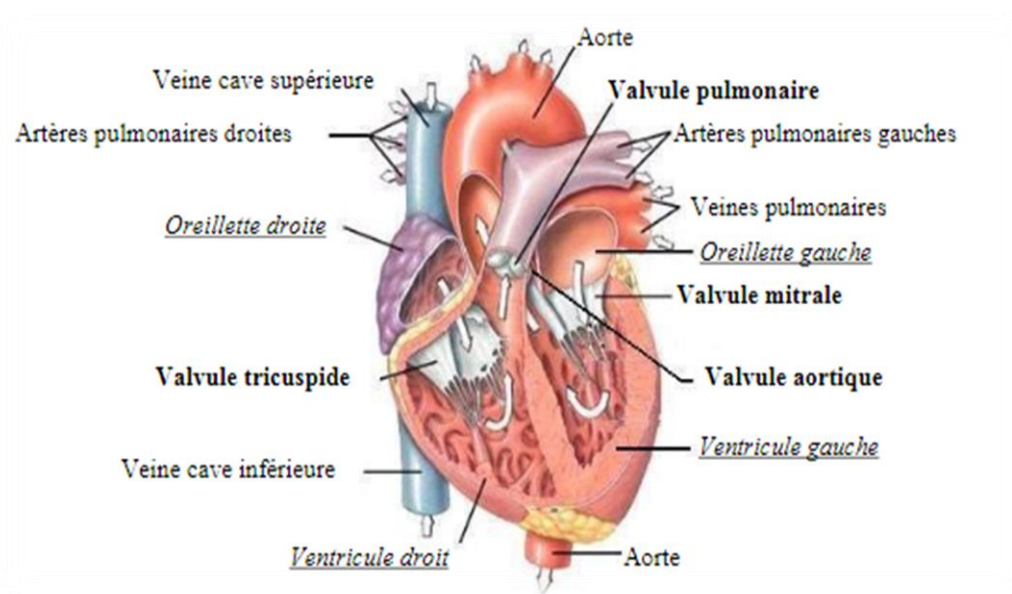


Figure 1.2 : Description anatomique du myocarde

Les valvules auriculo-ventriculaires assurent la liaison entre les oreillettes et les ventricules, tandis que les valvules sigmoïdes assurent la communication entre les ventricules et les gros vaisseaux.

Les valves mitrale et tricuspide permettent la communication entre les paires ventricule–oreillette gauche et droite respectivement. La sigmoïde aortique permet la circulation du sang du ventricule gauche vers l’aorte, et la sigmoïde pulmonaire permet d’assurer la liaison entre le ventricule droit vers l’artère pulmonaire

Les valves cardiaques jouent un rôle majeur dans le cycle cardiaque, empêchant le sang de couler en arrière.

Sur la **Figure 1.3** sont illustrées les coupes longitudinales et transversales d’une valve cardiaque dans les positions ouverte et fermée ;

- ✓ (a) *position ouverte* : les feuillets de la valve cardiaque sont éloignés afin de laisser passer le sang, cette ouverture engendre des turbulences sanguines qui font partie des bruits cardiaques.
- ✓ (b) *position fermée* : les feuillets se resserrent afin de ne pas permettre le passage du sang en cas d’insuffisance la valve ne se referme pas convenablement ce qui cause une régurgitation sanguine.

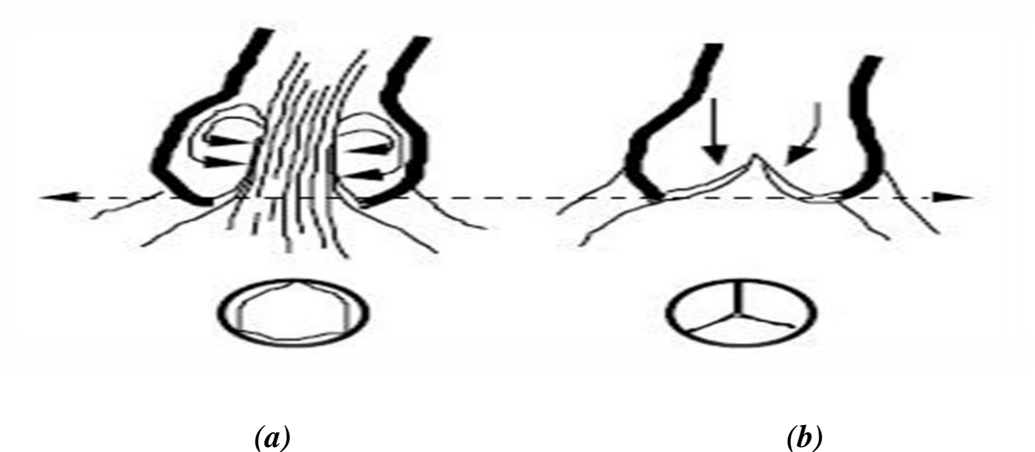


Figure. 1.3 : Coupes longitudinales et transversales d’une valve cardiaque dans les positions (a) ouverte (b) fermée

1.3. Révolution cardiaque

La révolution cardiaque se déroule en deux phases. La première phase dite systolique consiste à la contraction des deux ventricules ce qui provoque l'éjection du sang vers la périphérie du cœur à travers l'aorte, et vers les poumons par l'intermédiaire des artères pulmonaires. La deuxième phase dite diastolique permet la relaxation des deux ventricules en se remplissant de sang à travers les valvules auriculo-ventriculaires après accumulation dans les oreillettes.

L'étude de l'hémodynamique cardiaque peut être divisée en quatre phases essentielles la *contraction* et l'*éjection ventriculaire* constituant la *systole ventriculaire*, ensuite la *relaxation* et le *remplissage* lors de *diastole ventriculaire*.

1.3.1. Contraction ventriculaire

Cette phase est subdivisée en deux sous phases :

1. **Contraction pré- iso volumétrique** : la pression dans les ventricules augmente d'une façon rapide et devient supérieure à la pression des oreillettes, ce qui provoque la fermeture des valvules auriculo-ventriculaires.
2. **Contraction iso volumétrique**: les valvules auriculo-ventriculaires et sigmoïdes sont fermées. Les deux ventricules poursuivent à se contracter, et la pression devient supérieure à la pression dans les artères d'où l'ouverture des valvules sigmoïdes.

Cette phase correspond à l'apparition du premier groupe de vibrations acoustiques noté B1.

1.3.2. Ejection ventriculaire

L'éjection du sang ventriculaire vers les artères se réalise en trois phases :

1. **Phase d'éjection rapide** : le sang est propulsé rapidement avec une augmentation graduelle de la pression ventriculaire.
2. **Phase d'éjection lente** : le sang est éjecté légèrement selon la diminution de la pression ventriculaire.
3. **Protodiastole de Wiggers** : la pression ventriculaire diminue de façon trop rapide mais reste supérieure à la pression dans les artères.

Quand la pression dans les ventricules devient inférieure à la pression dans les artères, les valvules sigmoïdes se ferment, ce qui correspond à l'apparition du deuxième groupe de vibrations acoustiques, noté B2.

1.3.3. Relaxation

Pendant cette phase, la pression ventriculaire continue à diminuer jusqu'à ce qu'elle devient inférieure à la pression dans les oreillettes, ce qui induit l'ouverture des valvules auriculo-ventriculaires.

1.3.4. Remplissage

Durant cette phase, les ventricules se remplissent de sang par les oreillettes à travers les valvules auriculo-ventriculaires. Cette phase est subdivisée en quatre phases :

1. **Phase de remplissage rapide** : la pression ventriculaire est très réduite par rapport à la pression auriculaire ce qui résulte un remplissage rapide des ventricules.
2. **Phase de remplissage lent** : la pression dans les oreillettes commence à diminuer, par contre la pression ventriculaire continue à augmenter, le remplissage ventriculaire est alors moins rapide.
3. **Phase de Diastasis** : quand les pressions auriculaires et ventriculaires sont égales, le cœur est en phase de repos.
4. **Phase de remplissage actif** : les ventricules terminent leur remplissage par une augmentation de pression due à la systole auriculaire.

La figure 1.4 suivante illustre les événements successifs qui se réalisent pendant un cycle cardiaque au niveau du cœur avec l'évolution temporelle pendant un cycle cardiaque de plusieurs paramètres physiologiques en relation avec le fonctionnement du cœur comme le signal électrocardiogramme ECG et les pressions intracardiaques ainsi que le signal étudié qui est le phono cardiogramme PCG.

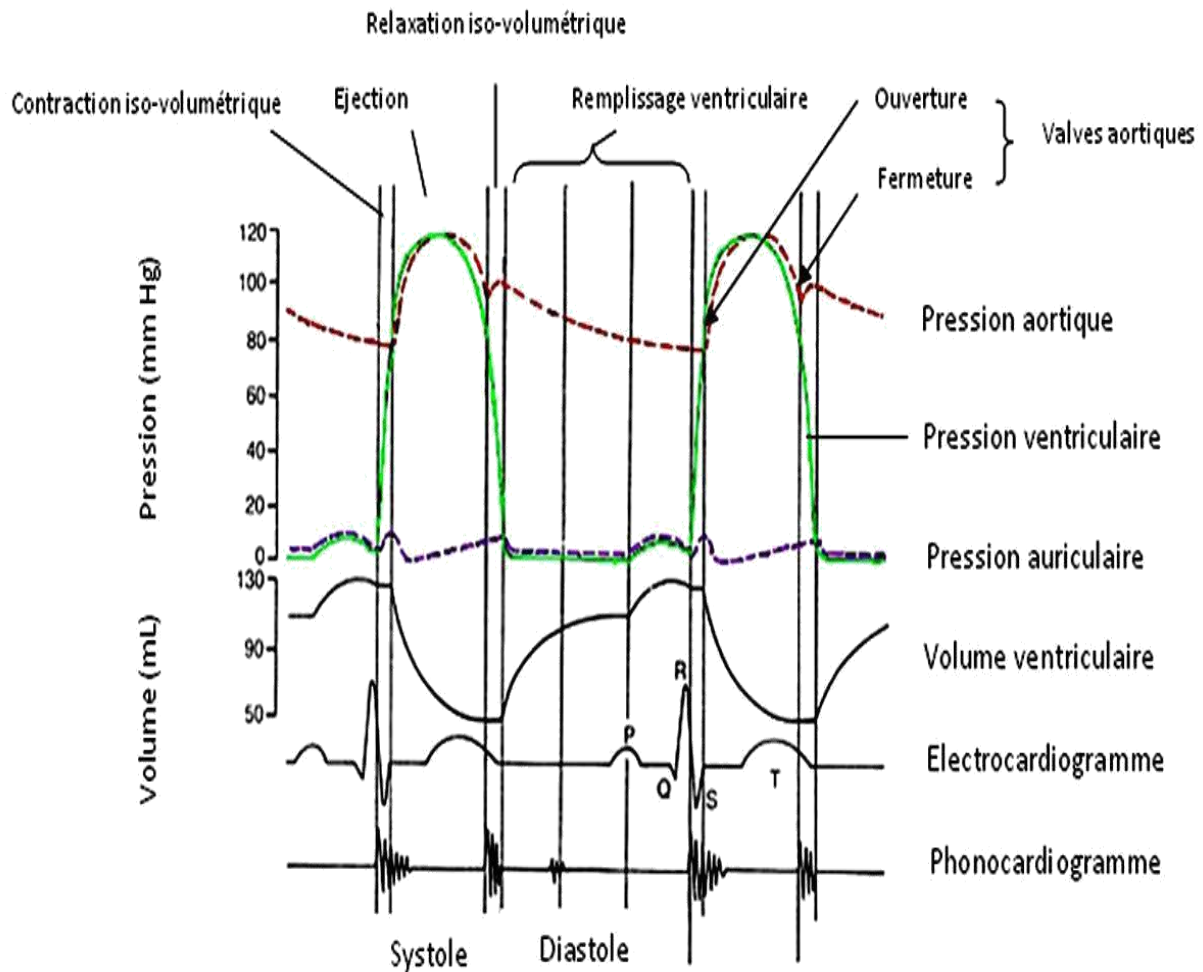
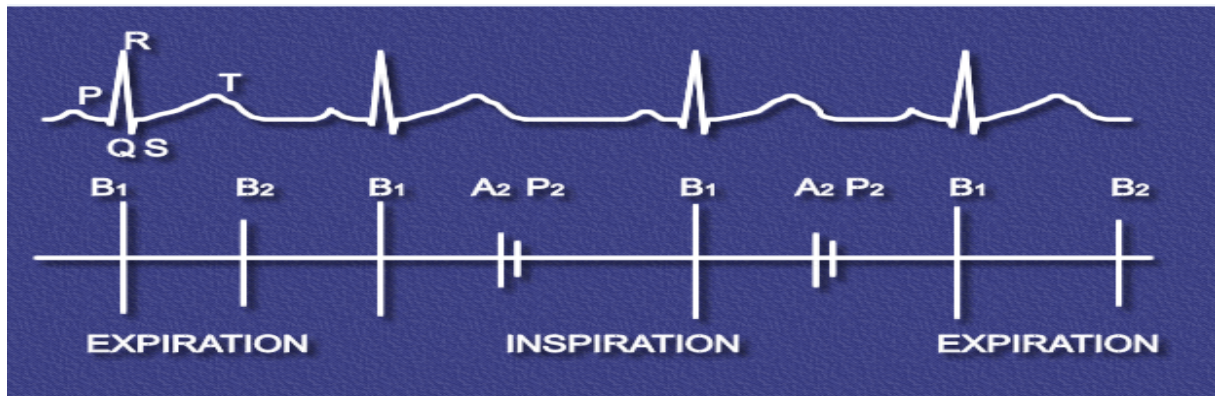


Figure 1.4 : Les évènements et les changements qui se produisent au cours d'un cycle cardiaque.

1.4. Description du Signal Phono cardiographique

Les bruits cardiaques sont des ondes acoustiques résultantes des battements du cœur ainsi que du flux sanguin qui coule à travers ses cavités. Le signal enregistré est appelé phono cardiogramme PCG.

Ces bruits cardiaques sont des vibrations mécaniques 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 entourent le cœur.



Temps, sec.

Figure 1.5 : Signal Phono cardiographique normal

Il existe quatre types de bruits cardiaques connus par B1, B2, B3 et B4, qui peuvent être auscultés et visualisés sur un signal phonocardiographique PCG.

Les bruits B1 et B2 sont les deux principaux bruits entendus dans une auscultation d'un sujet sain. L'intervalle entre le premier et le deuxième bruit définit la systole (éjection ventriculaire), tandis que l'intervalle entre le deuxième et le premier bruit suivant définit la diastole (remplissage ventriculaire).

1.4.1 Bruit B1

Ce bruit cardiaque est caractérisé par son contenu fréquentiel de basses fréquences par rapport au bruit B2 du même cycle cardiaque. Marquant le début de la systole ventriculaire, ce bruit se trouve, en sa grande proportion en terme d'énergie, généré par la fermeture des valvules auriculo-ventriculaires mitrale et tricuspide.

Le bruit cardiaque B1 est constitué par quatre groupes de vibrations, comme illustré dans la figure 1.6 :

1. Oscillations de faible amplitude et de basse fréquence, non audibles, elles correspondent au début de la contraction ventriculaire.
2. Oscillations de grande amplitude, audibles correspondant aux fermetures des valvules auriculo-ventriculaires mitrale (M1) et tricuspide (T1).
3. Oscillations de grande amplitude, audibles correspondant à l'ouverture des sigmoïdes.

4. Oscillations de faible amplitude et de basse fréquence, non audibles, dues à l'écoulement turbulent du sang dans l'aorte et l'artère pulmonaire, brusquement dilatée au début de l'éjection ventriculaire.

1.4.2 Bruit B2

Ce bruit cardiaque est caractérisé par son contenu acoustique de fréquences aiguës. Il est plus sec et plus claquant que le bruit B1, et de tonalité plus élevée et peut atteindre des fréquences jusqu'à 200 Hz []. Il correspond aux fermetures des sigmoïdes aortique et pulmonaire. Ce deuxième bruit marque le début de la diastole ventriculaire, sa durée ne dépasse pas généralement 100 ms.

Le bruit B2 est essentiellement composé de deux groupes de vibrations comme illustré dans la figure 1.6.

1. vibrations de faible amplitude non audible à cause des tourbillons sanguins précédant la fermeture des sigmoïdes.
2. vibrations audibles de hautes fréquences dues aux fermetures des sigmoïdes aortique et pulmonaire, notées respectivement A2 et P2. Chacune des composantes A2 et P2 dure moins de 50 ms, et sont généralement séparées par un intervalle de temps de 3 à 4 ms qui augmente lors de l'inspiration.

1.4.3 Bruit B3 et bruit B4

Il y a également un troisième et un quatrième bruit cardiaque (B3 et B4) qui peuvent parfois être visibles sur un signal phono cardiogramme PCG.

Le troisième bruit B3 survient à la fin de la phase de remplissage ventriculaire rapide, et son apparition est après le bruit B2, comme illustré dans la figure 1.6. Il est habituellement de basse fréquence et peut s'entendre normalement chez l'enfant ou chez l'adulte au-dessous de 30 ans. Au-delà de cet âge, sa présence témoigne le plus souvent d'un état pathologique.

Par contre, le quatrième bruit B4 débute juste avant le premier bruit B1, comme illustré dans la figure 1.6. Il est habituellement de basse fréquence, il répond à la contraction auriculaire. Il n'est jamais audible normalement, et son accentuation témoigne d'une hyperactivité. Auriculaire, droit ou gauche (galop pré systolique).chez les personnes âgées de plus de 50 ans, le signal phono cardiogramme enregistre consommant (même si l'oreille ne le perçoit pas toujours) un 4ème bruit B4.

La figure 1.6 illustre un schéma qui montre les différentes composantes des bruits cardiaques B1, B2, B3 et B4 ainsi une illustration de leurs causes physiologiques. Uniquement le côté gauche du cœur est affiché dans cette figure.

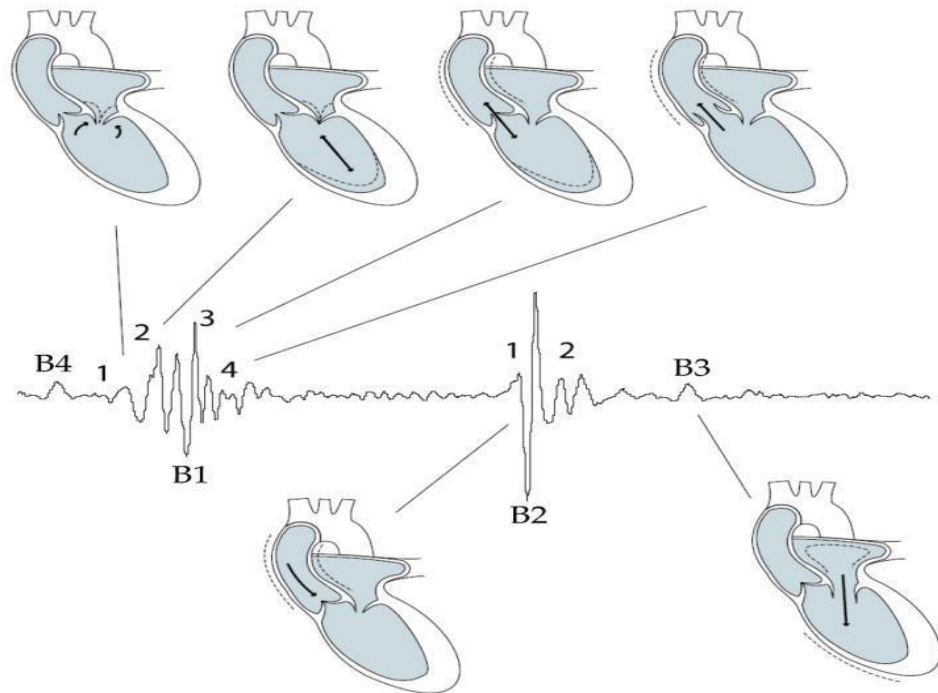


Figure 1.6 : Relation entre les bruits cardiaques et la cardiographie

1.4.4 Altération du fonctionnement normal du cœur

Le rôle des valvules cardiaques est l'aiguillage de la circulation du sang dans le cœur les valvules mitrale et tricuspide permettent l'acheminement du sang des oreillettes aux ventricules pendant la phase diastolique, tandis que les valvules aortique et pulmonaire permettent de véhiculer le sang des ventricules vers les gros vaisseaux.

Les valvulopathies sont des déformations des valvules cardiaques en rétrécissements ou insuffisance. Ces anomalies contraignent le fonctionnement normal du cœur et génèrent des souffles surajoutés sur le tracé Phono cardiographique (PCG) comme illustré dans le signal PCG présenté dans la figure 1.7.

Les souffles cardiaques sont des sons de plus longue durée que les bruits normaux déjà décrits. Ils sont dus à des vibrations induites par un flux turbulent de sang. Ces turbulences ont diverses :

- ✓ Des insuffisances de fermeture des valves (comme illustré dans la figure 1.8 qui présente les cas d'ouverture et fermeture d'une valve aortique normale et pathologique)
- ✓ Des réductions de calibre des passages empruntés par le sang ou sténoses (comme illustré dans la figure 1.9)
- ✓ Des communications anormales entre les diverses chambres du cœur.
- ✓ Des variations de la vitesse et de la viscosité du sang.

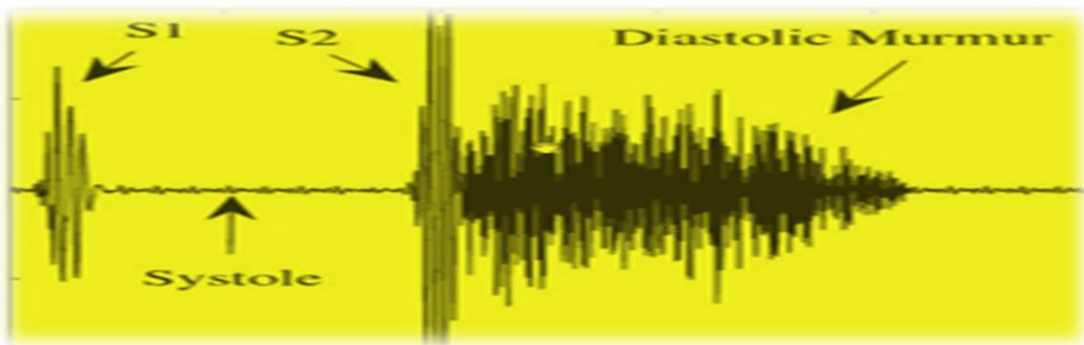


Figure 1.7 : signal PCG avec la présence d'un souffle diastolique

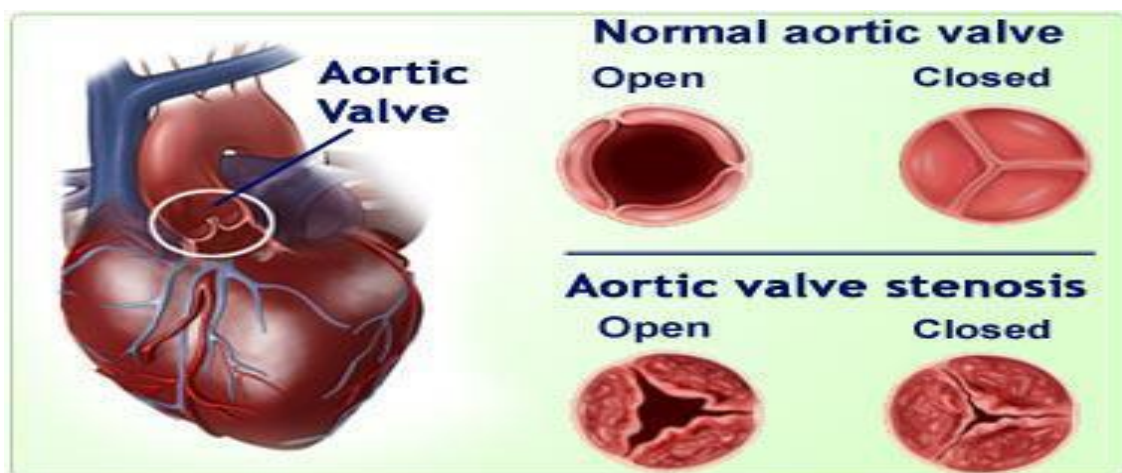


Figure 1.8 : Valvule aortique dans les cas sténose et normal

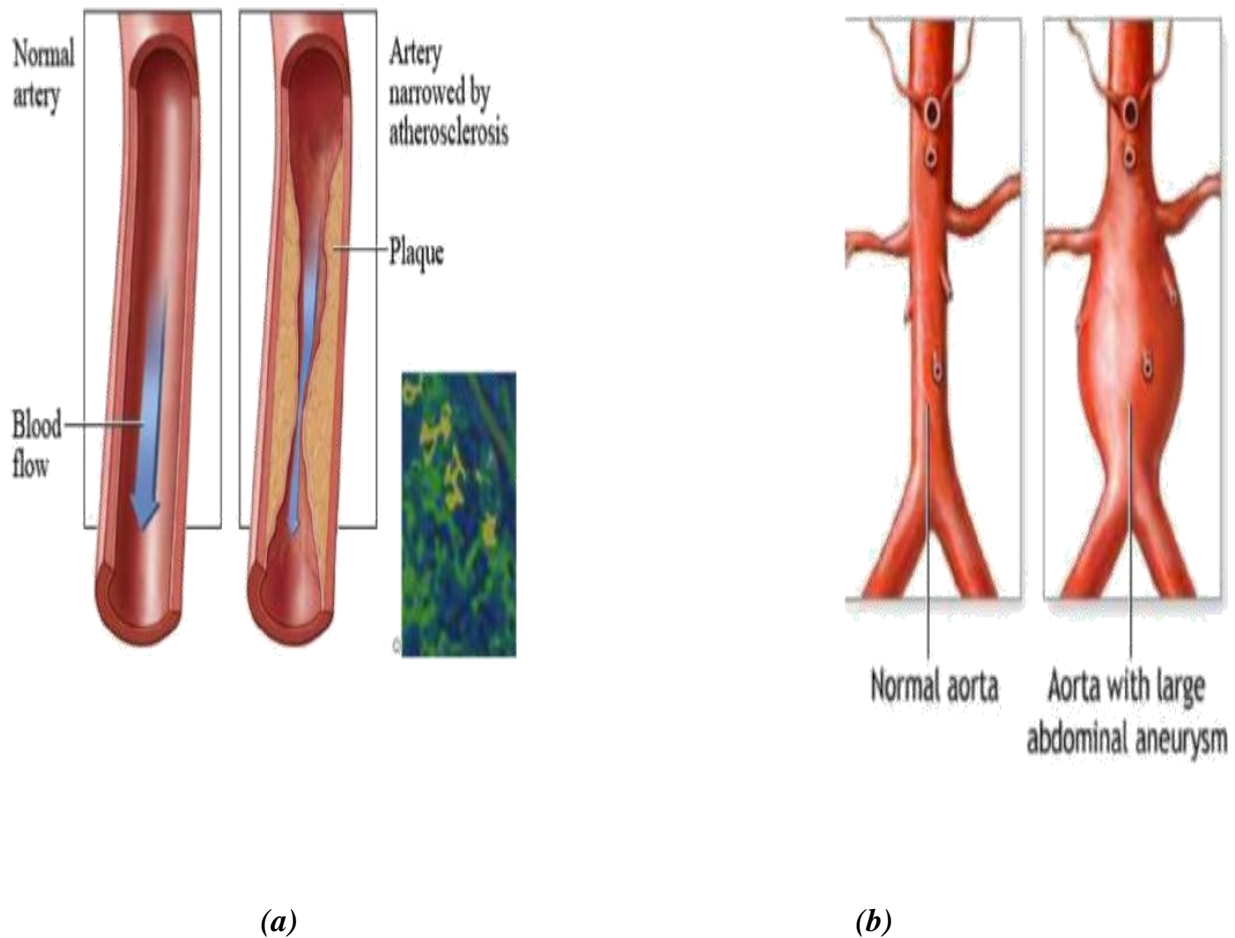


Figure 1.9 : (a) Sténose au niveau de l'artère ; (b) rétrécissement de l'aorte

1.5 Caractéristiques temporelles et fréquentielles du signal PCG

Comme déjà vu précédemment, pour chaque anomalie cardiaque correspond un signal PCG pathologique bien spécifique, où les caractéristiques fréquentielle et temporelle différent d'un cas à un autre.

Les caractéristiques temporelles et fréquentielles des différents bruits sont affichées dans le tableau ci-dessous.

Les bruits	Localité (ms)	La durée (ms)	Bande de fréq.
B1	10-50 après le pic R dans l'ECG	100-160	10-140
B2	280-360 après le pic R dans l'ECG	80-140	10-400
B3	440-460 après le pic R dans l'ECG ou 120-180 après la cloture des valves sigmoïdes	40-80	15-60
B4	40-120 après l'onde P dans l'ECG	30-60	15-45

Tab. 1.1 : Les caractéristiques temporelles et fréquentielles des bruits cardiaques

Les bruits cardiaques sont essentiellement des sons de fréquences basses à faible niveau. Comme le montre la figure 1.10, seule une petite partie des sons cardiaques se trouvent dans la plage d'audibilité humaine. Cela signifie que nos oreilles sont incapables d'entendre correctement ces sons et donc nous ne serons pas en mesure d'analyser les moindres détails répandus afin de diagnostiquer le patient correctement. Donc, nous devons amplifier ces sons pour entendre et interpréter correctement.

Pour cette raison l'utilisation d'un stéthoscope s'avère une nécessité pour une bonne auscultation des bruits cardiaques.

Pression
acoustique
en dynes
par cm^2

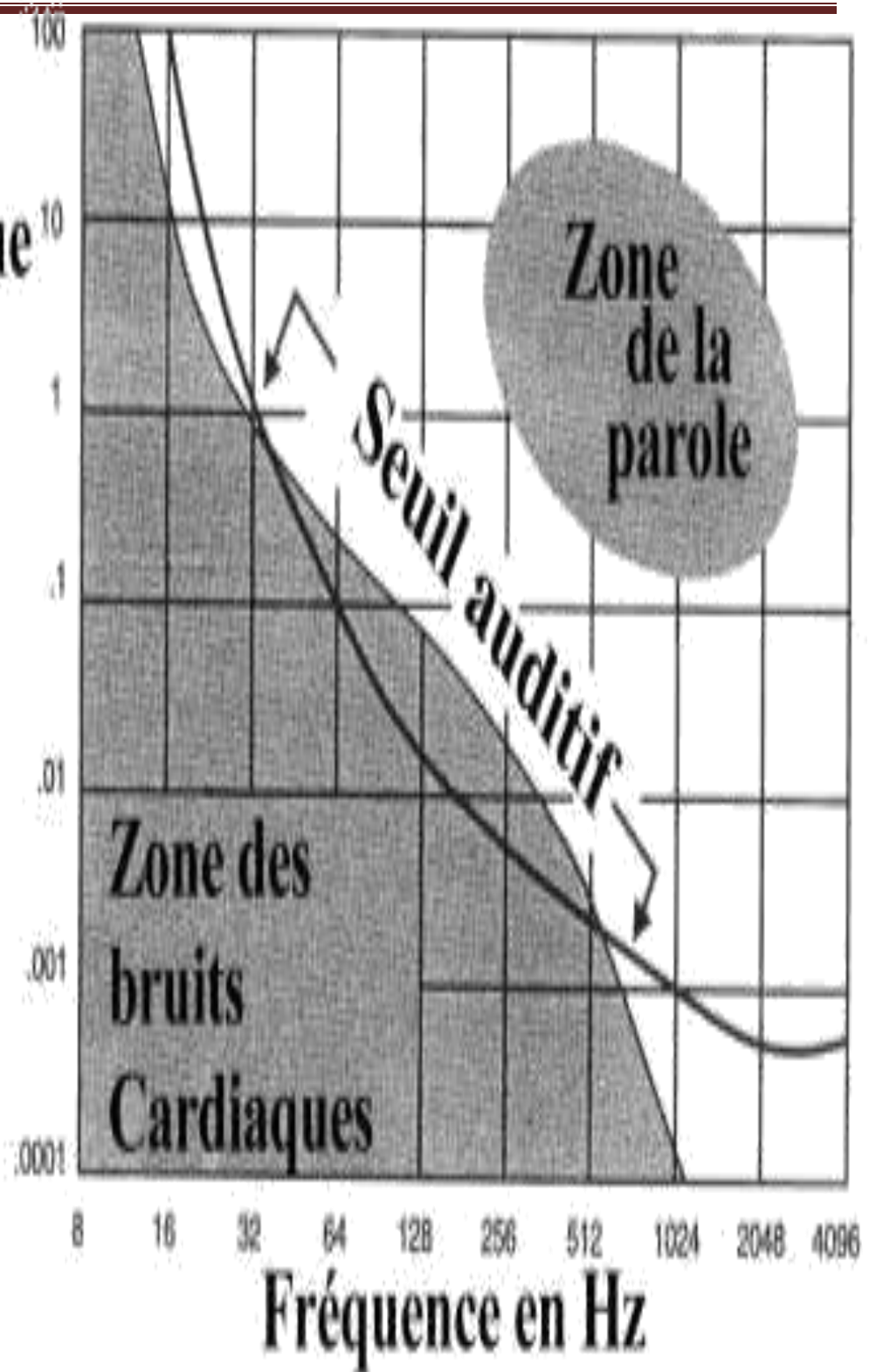


Figure 1.10 : Spectre des bruits cardiaques

Conclusion

L'étude de l'hémodynamique intracardiaque menée dans ce chapitre nous a renseignés sur les différents bruits cardiaques, leur forme et leurs origines de formation, Comme elle nous a permis de savoir les caractéristiques temporelles et fréquentielle de ces bruits.

Le signal enregistré aux cours d'une auscultation des sons cardiaques est le signal phono cardiogramme PCG. Ce signal nous permet de diagnostiquer certaines pathologies cardiaques en relation avec le mécanisme cardiaque d'où la nécessité de développer un système d'auscultation électronique qui permettra aux médecins d'enregistrer ces bruits cardiaques sous forme d'un signal phonocardiogramme PCG.

Le prochain chapitre présente une étude des différentes parties de notre stéthoscope numérique dont les deux parties numérique et analogique qui sont nécessaires pour la conception de notre système réaliser.

Chapitre 2

Etude de stéthoscope numérique

Introduction

Ce chapitre est consacré à l'étude des différents blocs de notre stéthoscope numérique dédié à la détection et l'acquisition des bruits cardiaques sous forme d'un signal électrique connu du signal phono cardiogramme PCG.

Ce stéthoscope digital est constitué principalement de deux parties Analogique et Numérique, dont la partie numérique elle-même se décompose en deux parties (le module EASY VR et la carte ARDUINO UNO) et la partie Analogique d'un microphone.

Le schéma bloc illustré dans la figure 2.1 présente la plateforme à partir de laquelle notre circuit a été réalisé. Il est constitué principalement :

- ✓ Capteur : Un microphone (partie Analogique).
- ✓ Module Easy vr.
- ✓ Acquisition du signal (Arduino Uno)

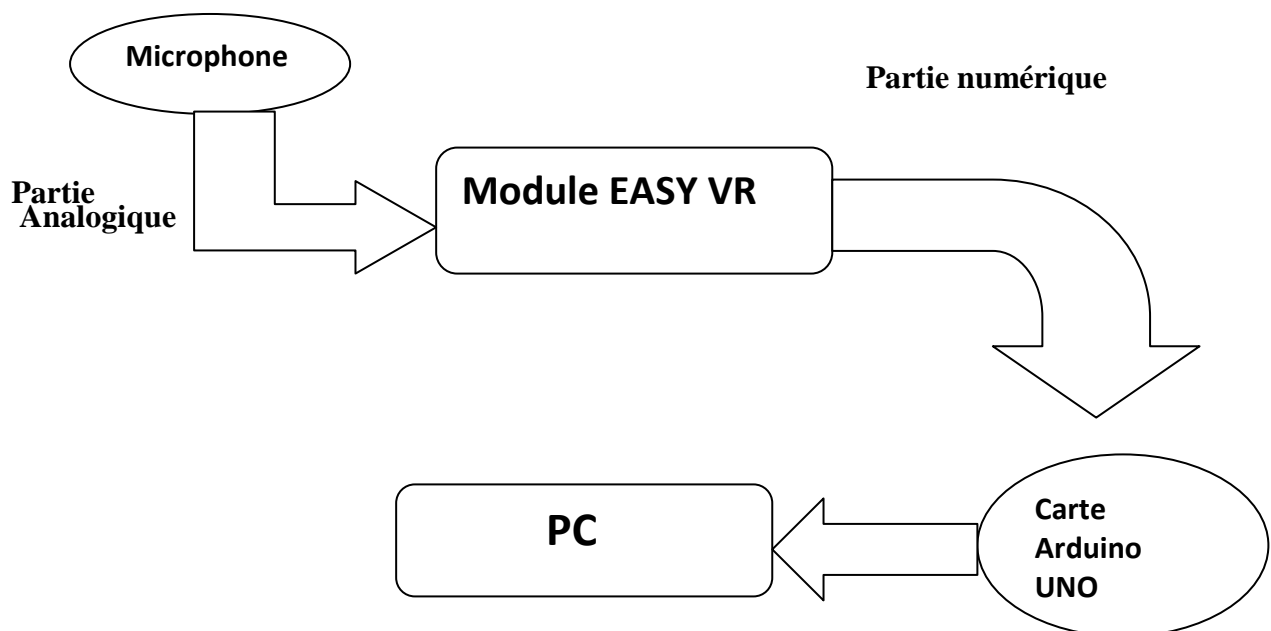


Figure.2.1 : Le schéma bloc du Stéthoscope numérique

2.1 Le Capteur

Le capteur est l'élément principal d'un instrument de mesure auquel est directement appliquée la grandeur à mesurer. Son rôle est de transformer les valeurs de cette grandeur appelée mesurande, en signaux électriques exploitables par les autres éléments de la chaîne de mesure. Pour cela, ce dernier doit être choisi avec minutie. Le capteur utilisé dans notre projet est un microphone.

Le microphone est un capteur qui convertit l'énergie du son (sous forme de pression) en énergie électrique.

Lors d'un enregistrement sonore, les capteurs les plus souvent utilisés sont les microphones ou les accéléromètres (appelé aussi microphone de contact ou microphone piézoélectrique).

Ces capteurs ont une bonne réponse dans la bande des hautes fréquences qui les rendent tout à fait adéquates pour les sons du corps humains. Mais, le principe c'est la réponse dans les basses fréquences qui est le principal inconvénient de ces capteurs ultrasonores.

Le microphone est un capteur qui mesure les ondes vibratoire de pression induites par les mouvements de différentes composantes du cœur, tandis que les accéléromètres sont des capteurs de contact qui mesurent directement les mouvements de la paroi thoracique.

2.1.1 Le Microphone a contact (accéléromètre piézoélectrique)

Ce type de microphone exploite la propriété piézoélectrique d'un Crystal qui va se polariser

Électriquement lorsqu'il est soumis à une contrainte mécanique. Le principe est de détecter la variation des vibrations d'un solide (dans ce cas, la paroi thoracique) et non pas les variations de la pression de l'air. Ce capteur doit être collé à la cage thoracique près du cœur pour pouvoir détecter les mouvements des vibrations dû au battements cardiaques [Oma'09]

Les vibrations détectées déforment la membrane piézoélectrique et changent la valeur de la capacité entre cette membrane et une autre plaque métallique fixe, induisant un changement dans la tension de condensateur dans des conditions de charge constante.

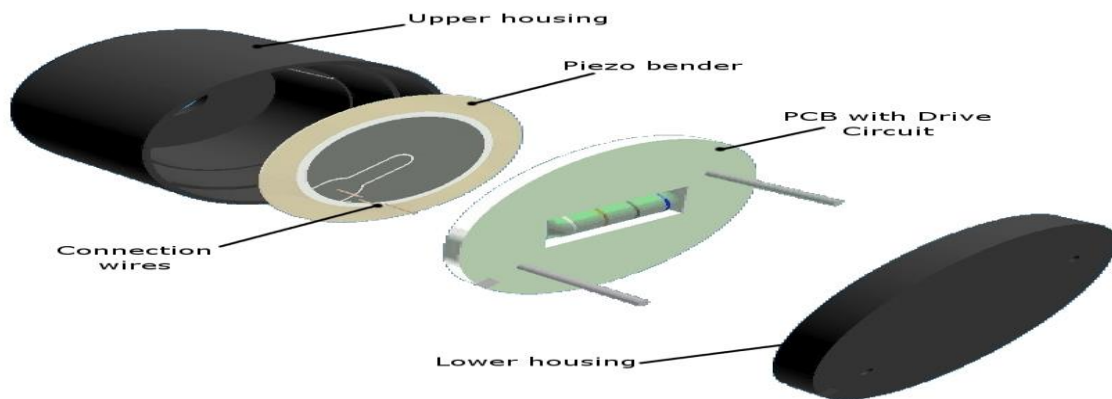


Figure 2.2 : Un microphone piézoélectrique.

Les microphones à leurs tours se divisent en trois types les plus couramment utilisés :

2.1.2 Microphone Dynamique

Ce type de microphone est constitué par un diaphragme en plastique ou en métal, ce dernier est fixé à une bobine en cuivre, située dans un champ magnétique créé par un aimant. Les ondes de pression sonore qui percutent le diaphragme provoquent son déplacement, qui entraîne à son tour celui de la bobine située dans le champ magnétique. Les variations magnétiques qui en résultent se traduisent par des variations électriques correspondant généralement aux variations physiques de l'onde sonore d'origine. Les microphones dynamiques sont extrêmement robustes, offrent une réponse en fréquence lisse et étendu, ne nécessitent pas une source DC externe pour leur fonctionnement. Les micros dynamiques sont largement utilisés dans le discours public, et pratiquement dans toutes les applications d'enregistrement.

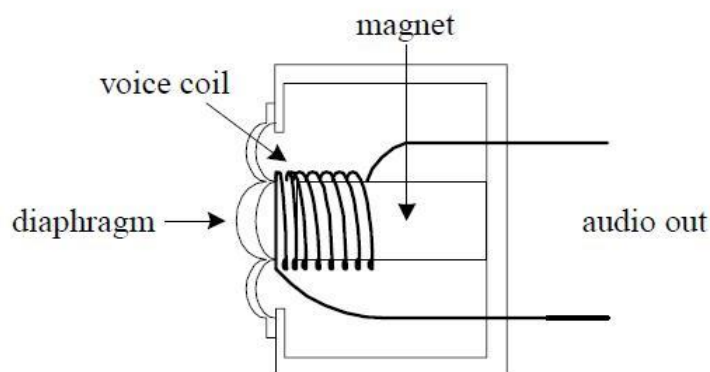


Figure. 2.3 Le microphone dynamique

2.1.3 Le Microphone à Condenseur

Ce type de microphone se compose d'une paire de plaques qui peuvent se rapprocher et s'éloigner par l'impact de la pression atmosphérique. En effet, les plaques agissent comme un condensateur sensible au son. L'une des plaques est en métal rigide fixé avec la masse.

L'autre plaque est en métal ou en plastique métallisé flexible qui est chargé positivement par une source de tension externe. Les microphones à condensateur sont caractérisés par un son de faible bruit et par un l'enregistrement de haute qualité [Oma'09].

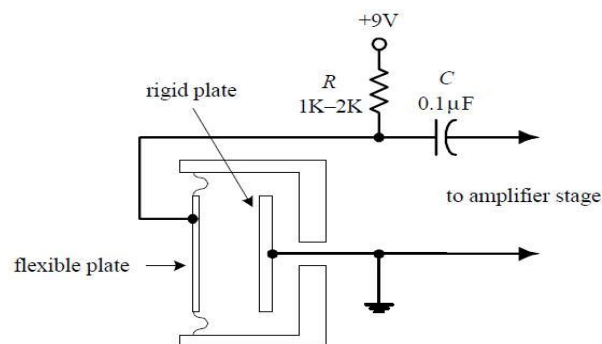


Figure 2.4 : Le microphone à condensateur

2.1.4 Le microphone Electret (ECM)

Un microphone à condensateur électret (ECM : Electret Condenser Microphone) est une variante du microphone à condensateur.

Au lieu d'utiliser une source de tension externe pour charger le diaphragme, il utilise un élément en plastique chargé de manière permanente (électret) placés en parallèle avec une plaque de métal conducteur.

La plupart des microphones électret ont un amplificateur FET intégré dans leur petit amplificateur. Cela nécessite une alimentation externe pour fonctionner, généralement une tension DC comprise entre +4 et +10 V. Cette tension alimente le microphone à travers une résistance (1-10 K) (voir figure 2.5). Le microphone électret répond bien aux fréquences moyennes, cependant il a une mauvaise réponse aux basses fréquences. Pour cette raison, son utilisation se limite aux communications vocales. En outre, la performance des microphones électret diminue au fil des ans à cause de la diminution de la charge d'électret

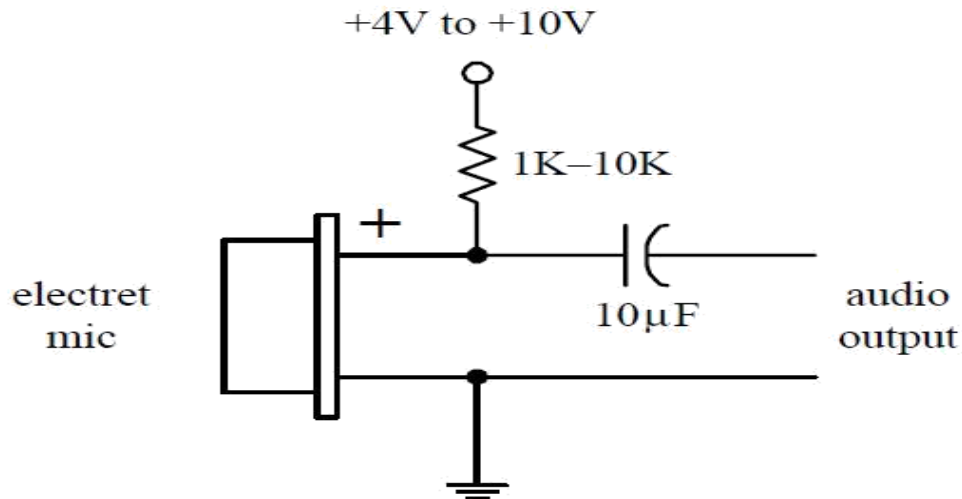


Figure 2.5 : Le microphone à électret

Le tableau suivant (tableau 2.1) montre une étude comparative des caractéristiques des trois types de microphones :

	<i>Microphone</i>		
	<i>ECM</i>	<i>dynamique</i>	<i>piézoélectrique</i>
Cout	faible	faible	important
Sensibilité	bonne	faible	bonne
Dimension	petit	grand	petit
Bande de fréquence	[20Hz-20kHz]	[50Hz-20kHz]	[10Hz-20kHz]

Tab.2.1 : Comparaison entre les caractéristiques des microphones

D'après ces caractéristiques présentées dans le tableau précédant, notre choix est vite tourné vers l'utilisation d'un microphone à électret ECM. Ce choix est aussi consolidé par la grande disponibilité de ce type de capteur dans les magasins des composants électroniques.

2.1.5 Capteur

Un Microphone à électret (ECM) a été utilisé pour la détection des bruits cardiaques celui illustré dans la figure 2.6.

Ce microphone ECM a été inséré dans l'extrémité de la tubulure d'un stéthoscope pour acquérir sons cardiaques comme illustré dans les figures 2.7 et 2.8.

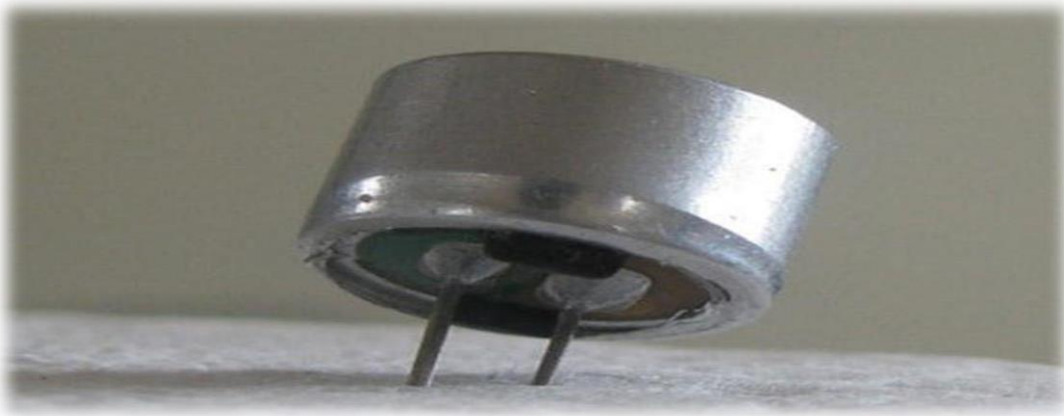


Figure 2.6 Photos du capteur ECM

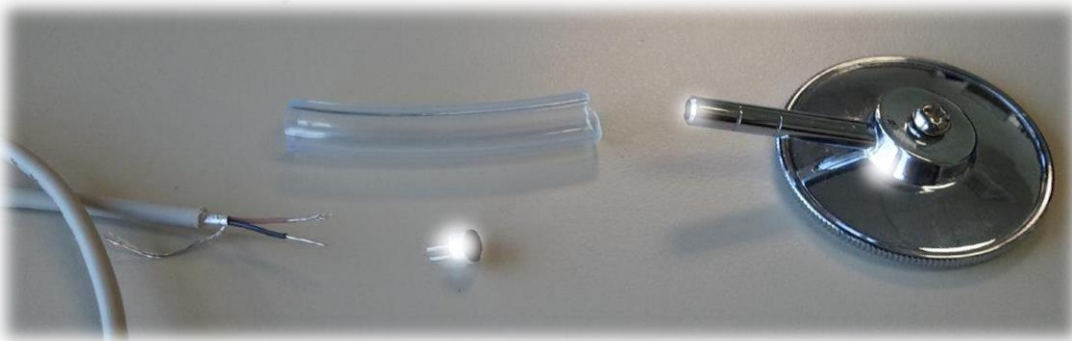


Figure 2.7 : les éléments nécessaire pour le capteur des bruits cardiaques



Figure 2.8 : capteur réalisé

2.2 Le conditionnement du signal

En instrumentation électronique, Les signaux issus directement des capteurs sont souvent des signaux incompatibles avec l'étage d'acquisition de ces données. Pour surmonter cette incompatibilité, le signal du capteur doit être conditionné. De plus, le type de conditionnement de signaux requis dépend du capteur utilisé.

Sachant que le signal sonore des mouvements cardiaques est de très faible amplitude et ne peut pas être détectée clairement donc, la sortie du capteur nécessite premièrement une amplification et ensuite traitée par des filtres appropriés.

2.3 Description de la carte ARDUINO UNO

Le modèle UNO(figure 2) de la société ARDUINO est une carte électronique dont le noyau est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est une microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C. Cette carte possède 14 bits entrées/sorties numérique (dont 6 peuvent être utilisées comme étant des sorties PWM(pulse width modulation)), 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits de résolution, 1 résonateur céramique(quartz) de 16MHz, 1 connecteur ICSP(In circuit Serial programming) qui permet d'injecter le boot loader à l'intérieur du microcontrôleur, 1 connecteur jack pour une alimentation extérieure, un bouton de reset pour mettre le processus à zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de pilote pour faire la conversion FTDI USB/Série, elle a juste un petit microcontrôleur ATmega8 (pour la version 2) programmé comme convertisseur USB/Série.

Elle contient tout ce que le microcontrôleur a besoin pour fonctionner, il faut seulement la connecter avec un câble USB à un ordinateur ou avec une alimentation externe pour commencer.

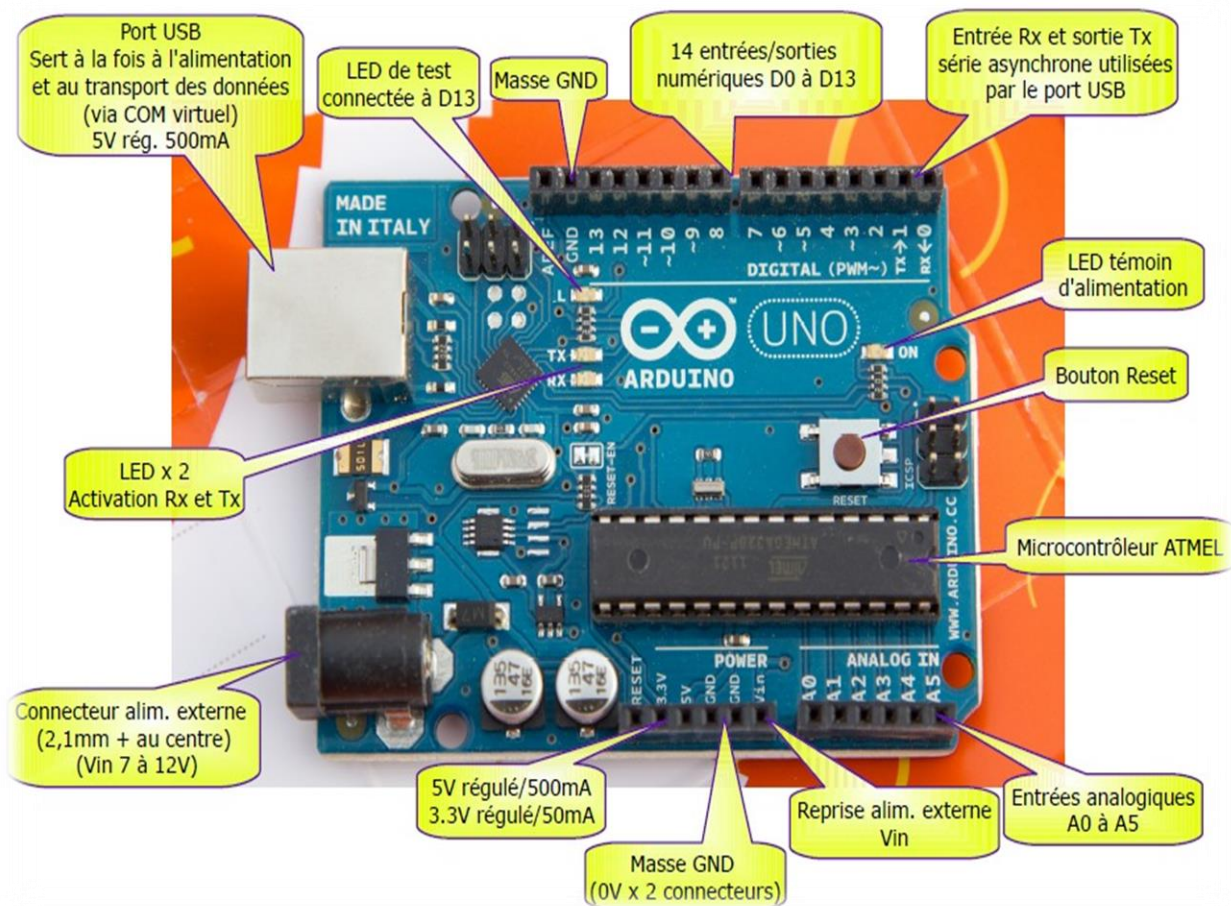


Figure 2.9 : Description de la carte ARDUINO UNO

L'intérêt principal des cartes ARDUINO UNO est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open-source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions « Clé » en main sont également fournies pour l'exploitation d'entrées/sorties courantes : gestion des E/S ? gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C, exploitation de servomoteurs, émission/réception série...

2.4 Alimentation de la carte Arduino

La carte Arduino Uno peut être alimentée soit via la connexion USB (qui fournit 5v jusqu'à 500 mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3v à 12 v sous 500mA) ou des piles (ou des accus).

L'adaptateur secteur peut être connecté en branchant une prise de 2,1mm, dédié du pôle positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de

pires ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0 v) et vin (tension positive en entrée) du connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20v. Cependant, si on utilise plus de 12 v, le régulateur de la tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7v et 12v.

Il est noté qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jack et d'avoir le câble USB connecté (risque de destruction de la carte). Les broches d'alimentation sont les suivantes :

- ✓ **VIN** : (à distinguer du 5v de la connexion USB ou autre source 5v régulée). on peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, on peut accéder à la tension d'alimentation sur cette broche.
- ✓ **5v** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite « tension régulée » obtenue à l'aide d'un composant appelé un régulateur et qui est intégré dans la carte arduino). Le 5v régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5v régulé) ou de toute autre source d'alimentation régulée.
- ✓ **3V3** : Une alimentation de 3,3v fournie par le régulateur de 3,3v de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5v). L'intensité maximale disponible sur cette broche est de 50mA.

2.5 Protection du port USB contre la surcharge en intensité :

La carte Arduino Uno intègre un poly-fusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliquées au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

2.6 Gestion des mémoires dans la carte ARDUINO

L'ATmega 328 a 32ko de mémoire FLASH pour stocker le programme (dont 0,5ko également utilisés par le boot loader). L'ATmega 328 a également 2ko de mémoire SRAM

(volatile) et 1ko d'EEPROM (non volatile-mémoire qui peut être lue à l'aide de la librairie EEPROM).

2.6.1 Les entrées/sorties numérique

Chacune des 14 broches numériques de la carte UNO(numérotées des 0à13)peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions PIN Mode(),DigitalWrite() et Digital Read() du langage Arduino.Ces broches fonctionnent en 5v.chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne « résistance de rappel »(pull-up)(déconnectée par défaut)de 20-50kohms.Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digital Write (broche, HIGH). Il Ya entre ces broches celles qui ont des fonctionnalités en plus :

- ✓ **Communication série** : broches 0(RX) et 1(TX). Utilisées pour recevoir (RX)et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers –série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel à la transmission série a travers ces broches avec l'instruction Serial.print(), à condition que le câble USB soit déconnecté, il va y avoir un chevauchement.
- ✓ **Interruptions Externes** : broches 2et 3, ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt pour plus de détails.
- ✓ **Impulsion PWM (largeur d'impulsion modulée)** :Broches3,5,6,9,10 et 11.Fournissent une pwm8-bits à l'aide de l'instruction analogWrite.
- ✓ **SPI (Interface série périphérique)** : Broche 10(SS), 11(MOSI), 12(MISO), 13(SCK). Ces broches supportent la communication SPI (interface série périphérique) disponible avec la librairie pour communication SPI.LES BROCHES SPI sont également connectées sur le connecteur ICSP.
- ✓ **I2C** : Broches 4(SDA) et 5(SCL), supportent les communications de protocole I2C, disponible en utilisant la librairie Wire/I2C.
- ✓ **LED** : Broches13, Il Ya une LED incluse dans la carte connectée à la broche 13.lorsque la broche est au niveau HAUT , la LED est allumée, lorsque la broche est au niveau BAS ,la LED est éteinte

2.6.2 Les entrées analogiques

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. par défaut, ces broches mesurent entre 0 V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

NB : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19, aux cas où le nombre de broches numériques n'est suffisant.

2.6.3 Autres broches

- ✓ Il Ya deux autres broches disponibles sur la carte :
- ✓ **AREF :** Tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction `analogReference()`. Elle s'utilise pour réduire.
- ✓ **Reset :** Mettre cette broche au niveau BAS entraîne la réinitialisation (=le redémarrage) du microcontrôleur. typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

On peut voir les différentes broches de la carte et leurs fonctions à travers la figure suivante :

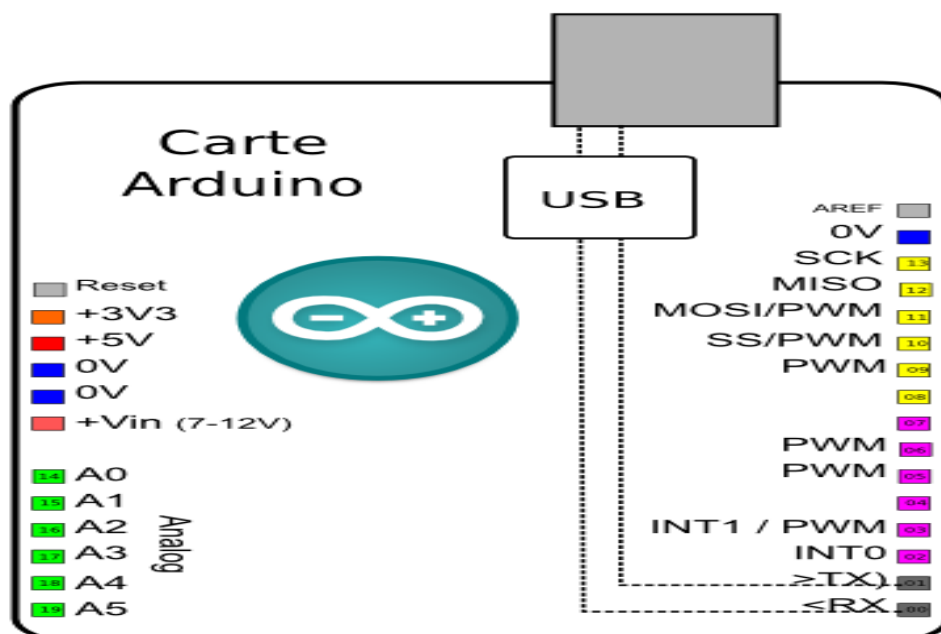


Figure 2.10 : Brochage de la carte ARDUINO UNO

2.7 ARDUINO et la communication avec l'extérieur

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec une autre carte Arduino, ou avec d'autre microcontrôleur.

L'ATmega328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur – récepteur universel asynchrone en français) pour communication série de niveau TTL(5v) et qui est disponible sur les broches 0(RX) et 1(TX).

Un certain intégré ATmega 8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel Arduino (dont l'on va parler dans le chapitre suivant) inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré USB vers série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1). Une librairie série logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO. L'ATmega328 supporte également la communication par protocole I2C et SPI :

- ✓ Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- ✓ Pour utiliser la communication SPI (Interface série périphérique), la librairie pour communication SPI est disponible, il suffit de la faire inclure dans le programme au niveau du logiciel Arduino lors de la programmation.

2.7.1 Dimensions de la carte

Les longueurs et largeurs maximales de la Uno sont respectivement 6,86cm, avec le connecteur USB et le connecteur d'alimentation jack s'étendant au-delà des dimensions de la carte. Quatre trous de vis permettent à la carte d'être fixée sur une surface ou boîtier (pour l'embarquer sur un système). noter que la distance entre les broches 7 et 8 est de 0,16 pouces, et 0,1 pouces séparant les autres broches.

2.7.2 Le microcontrôleur ATmega328

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. Ceci un microcontrôleur ATMEL de la famille AVR sur 8 bits.

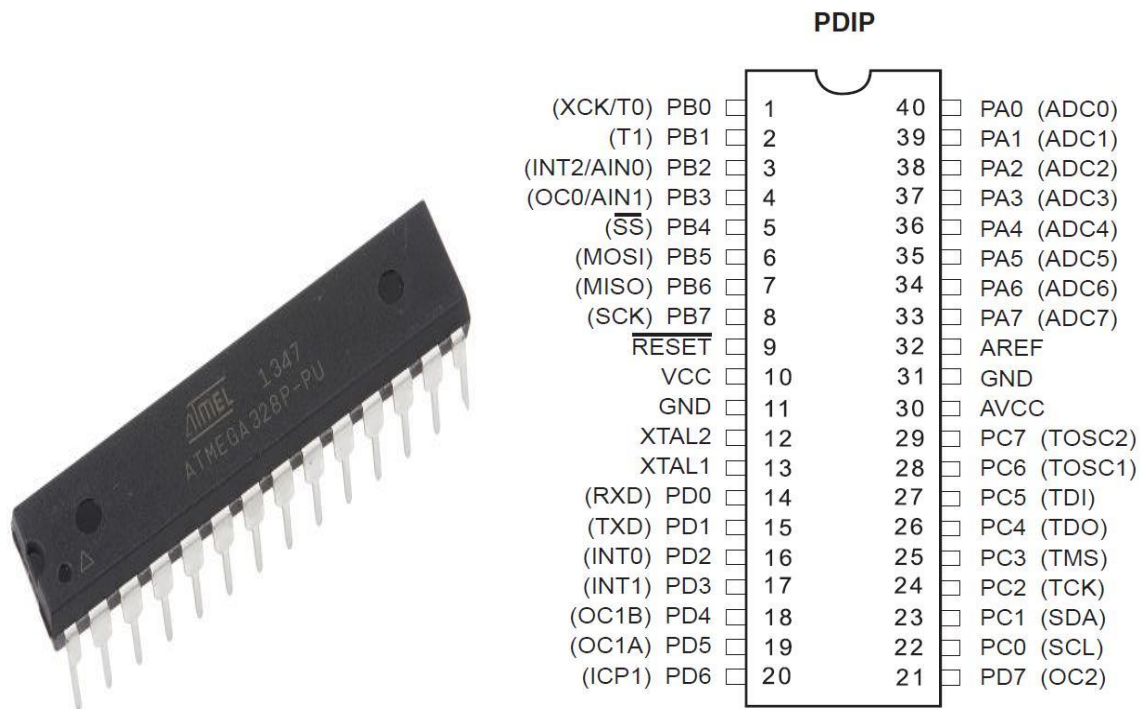


Figure2.11 : Le microcontrôleur ATMega328

2.8 Module Easy VR

2.8.1 Description du produit EasyVR

EasyVR est la version de deuxième génération du module VRbot réussi. Il s'agit d'un module de reconnaissance vocale polyvalent conçu pour ajouter facilement des fonctionnalités de reconnaissance vocale polyvalentes, robustes et rentables à presque toutes les applications.

Le module EasyVR peut être utilisé avec n'importe quel hôte avec une interface UART alimentée à 3.3V - 5V, comme les cartes PIC et Arduino. Certains exemples d'application incluent la domotique, tels que les interrupteurs d'éclairage, les serrures ou les lits commandés par la voix, ou l'ajout d'une «audition» aux robots les plus populaires sur le marché.

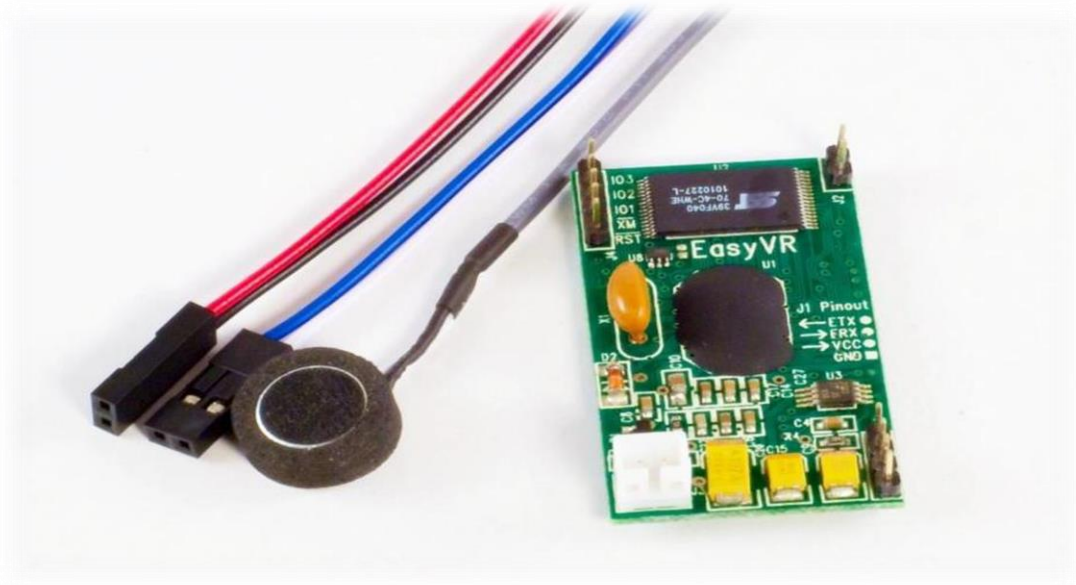


Figure 2.12 : Module EASY VR

2.8.2 Fonctionnalités EasyVR

- ✓ Une foule de commandes intégrées Speaker Independent (SI) pour les contrôles de base prêts à exécuter, dans les langues suivantes:
 - _ English (US)
 - _ Italian
 - _ allemand
 - _ français
 - _ Espagnol
 - _ japonais
- ✓ Prend en charge jusqu'à 32 déclencheurs ou commandes dépendants (SD) définis par l'utilisateur ainsi que des mots de passe vocaux. Les commandes personnalisées SD peuvent être parlées dans n'importe quelle langue.
- ✓ Interface utilisateur graphique facile à utiliser et simple pour programmer les commandes vocales et audio.
- ✓ Le module peut être utilisé avec n'importe quel hôte avec une interface UART (alimenté à 3.3V - 5V)

- ✓ Protocole série documenté simple et robuste pour accéder et programmer à travers le panneau hôte
- ✓ 3 lignes GPIO (IO1, IO2, IO3) qui peuvent être contrôlées par de nouvelles commandes de protocole.
- ✓ Sortie audio PWM prenant en charge les haut-parleurs de 8 ohms.
- ✓ Fonction de lecture sonore.
- ✓ Spécifications techniques.
- ✓ Dimensions physiques et affectation des broches.

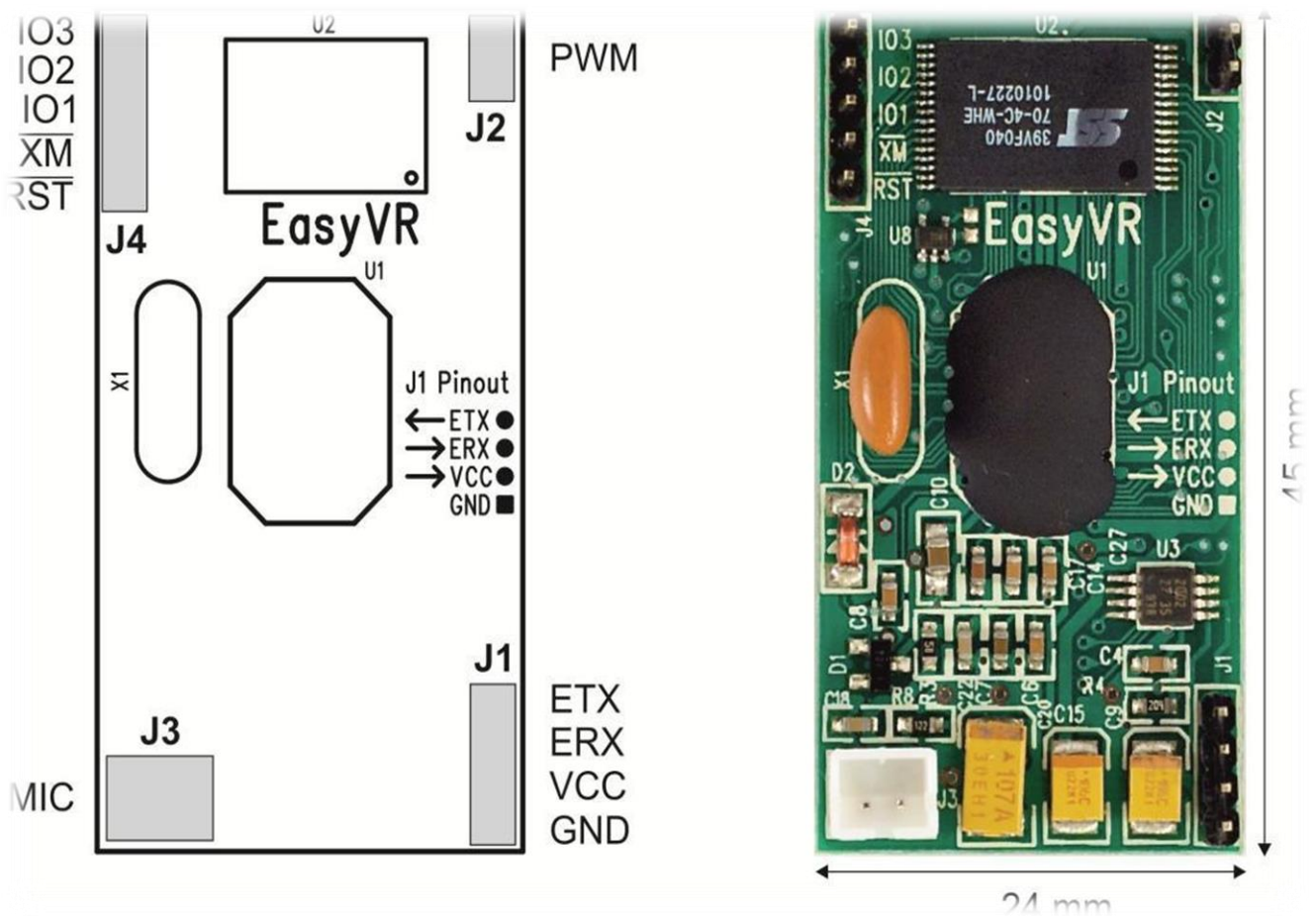


Figure 2.13 : Datasheet de module EASY VR

2.8.3 Numéro de connecteur Nom Type Description

- _ 1 GND - Ground
- _ J1 2 VCC I Entrée DC de tension
- _ 3 ERX I Serial Data Receive (niveau TTL)
- _ 4 ETX O Serial Data Transmit (niveau TTL)

- _ J2 1-2 PWM O Sortie audio différentielle (peut conduire directement un haut-parleur de 8Ω)
- _ J3 1 MIC_RET - Terre de référence du microphone
- _ 2 MIC_IN I Signal d'entrée du microphone
- _ 1 / RST I Réinitialisation asynchrone basse active (pull-up 100K interne)
- _ 2 / XM I Boot select (1K déroulant interne)J4
- _ 3 E / S IO1 E / S à usage général (niveau 3,0 VDC TTL)
- _ 4 IO2 E / S E / S à usage général (niveau 3,0 VDC TTL)
- _ 5 IO3 E / S E / S à usage général (niveau 3,0 VDC TTL)
- _ Symbole Paramètre Min Typ Max Unité
- _ VCC Tension entrée DC 3.3 5.0 5.5 V
- _ La Température de fonctionnement ambiante 0 25 70 ° C
- _ Données de réception du port série ERX 0 - VCC V
- _ Données de transmission du port série ETX 0 - VCC V

2.8.4 Caractéristiques électriques

- _ Ceux-ci s'appliquent uniquement aux broches J4, y compris IO1-3, / XM et / RST.
- _ Symbole Paramètre Min Typ Max Unité
- _ VIH Entrée haute tension 2.4 3.0 3.3 V
- _ VIL Entrée basse tension -0,1 0,0 0,75 V
- _ IIL Input Leakage Current (0 < VIO < 3V, entrée Hi-Z) <1 10 µA
- _ Résistance pull-up forte 10 kΩ
- _ Faible 200 kΩ
- _ VOH Sortie haute tension (IOH = -5 mA) 2,4 V
- _ VOL Output Low Voltage (IOL = 8 mA) 0.6 V
- _ Interface série
- _ L'EasyVR est un module "esclave" qui se transmet via une interface série asynchrone (communément appelée interface UART), avec les caractéristiques suivantes:
 - ✓ Baud Rate: 9600 (par défaut), 19200, 38700, 57600, 115200
 - ✓ Cadre: 8 bits de réception des données, pas de parité, 1 bit d'arrêt

2.8.5 Caractéristiques de Microphone

Le microphone fourni avec le module EasyVR est un microphone omnidirectionnel à condensateur à électret (Horn EM9745P-382):

- ✓ Sensibilité -38dB (0dB = 1V / Pa @ 1KHz)
- ✓ Impédance de charge 2.2K
- ✓ Tension de fonctionnement 3V
- ✓ Réponse de fréquence presque plate dans la gamme 100Hz - 20kHz

Si vous utilisez un microphone avec des spécifications différentes, la précision de la reconnaissance peut être compromise. Aucun autre type de microphone n'est supporté par EasyVR.

Remarque: Les commandes vocales doivent être données à partir d'environ 60 cm du microphone, mais vous pouvez essayer à plus grande distance en parlant plus fort.

Veuillez noter que le positionnement acoustique incorrect du microphone réduira la précision de la reconnaissance. De nombreux arrangements mécaniques sont possibles pour l'élément du microphone, et certains fonctionneront mieux que d'autres. Lors du montage du microphone dans l'appareil final, gardez à l'esprit les consignes suivantes:

1. Montage encastré : L'élément du microphone doit être positionné le plus près possible de la surface de montage et doit être complètement assis dans le boîtier en plastique. Il ne doit pas y avoir d'espace aérien entre l'élément du microphone et le boîtier. L'existence d'un tel espace aérien peut conduire à une résonance acoustique,

2. Aucune obstruction, grand trou : La zone située devant l'élément du microphone doit être exempte d'obstructions pour éviter toute interférence avec la reconnaissance. Le diamètre du trou dans le boîtier devant le microphone doit être d'au moins 5 mm. Toute surface plastique nécessaire devant le microphone doit être aussi mince que possible, ne dépassant pas 0,7 mm, si possible :

- ✓ zone dégagée
- ✓ interne
- ✓ diaphragme

3. Isolation : Le microphone doit être isolé acoustiquement du boîtier si possible. Cela peut être réalisé en entourant l'élément de microphone avec un matériau spongieux comme le caoutchouc ou la mousse. Le microphone fourni possède ce type de mousse isolante. L'objectif est d'éviter que les bruits auditifs produits par la manipulation ou le dérangement du périphérique ne soient «ramassés» par le microphone. De tels bruits étrangers peuvent réduire la précision de la reconnaissance :

- ✓ BON et MAUVAIS
- ✓ Absorbant fixé
- ✓ Matériel directement

4. Distance : Si le microphone est déplacé de 15 cm à 30 cm de la bouche du haut-parleur, la puissance du signal diminue d'un facteur de quatre. La différence entre une voix forte et douce peut aussi être plus qu'un facteur de quatre. Bien que le préamplificateur interne de l'EasyVR compense une large gamme dynamique de puissance du signal d'entrée, si sa portée est dépassée, l'application de l'utilisateur peut fournir des commentaires au locuteur sur le volume de la voix (voir les CODES D'ERREUR de l'annexe).

Sortie audio

L'interface de sortie audio EasyVR est capable de diriger directement un haut-parleur de 8Ω . Il pourrait également être connecté à un amplificateur audio externe pour conduire des haut-parleurs à impédance inférieure.

Remarque

La connexion de haut-parleurs avec une impédance inférieure directement au module peut endommager de manière permanente la sortie audio EasyVR ou tout le module.

Il est possible de connecter des charges d'impédance plus élevées telles que les écouteurs, pourvu que vous réduisez la puissance de sortie en fonction de la qualité des enceintes, par exemple en utilisant une résistance en série. La valeur de résistance exacte dépend de la puissance des écouteurs et du volume de sortie désiré (habituellement de l'ordre de $10\text{ k}\Omega$).

La connexion de haut-parleurs casques directement à la sortie audio EasyVR peut endommager votre audition.

E / S à usage général : étant donné que l'interface de communication EasyVR prend deux broches du contrôleur hôte, quelques broches d'E / S de rechange sont fournies, qui peuvent être contrôlées avec le protocole de communication, pour récupérer ces broches pour les tâches de base, telles que l'éclairage d'une LED.

Les trois broches d'E / S IO1-IO3 sont connectées directement au microcontrôleur intégré sur le module EasyVR, de sorte qu'elles sont référencées à l'alimentation interne régulée 3.0V.

Si vous avez besoin d'interface avec des circuits utilisant une alimentation différente, il existe un certain nombre de solutions que vous pouvez adopter. Certains d'entre eux sont décrits ci-dessous (ici IO*n* indique l'une des trois broches E / S de l'EasyVR).

- ✓ Utilisez une broche en tant que sortie. : toutes les broches d'E / S sont des entrées avec un pull-up interne faible après la mise sous tension. Vous devez configurer explicitement une broche avant de pouvoir l'utiliser comme sortie (voir l'exemple de code UTILISER LES Boutons E / S GÉNÉRALES).

- ✓ Broche E / S entraînant directement une broche E / S connectée à une broche E / S haute, en commutant une charge sur une haute tension. Circuit à 5 V d'impédance à LED à courant faible (comme la ligne à l'aide d'un relais de 12 V)
- ✓ Broche d'entrée MCU : Les valeurs des composants exacts dans ces circuits peuvent varier. Vous devez calculer les valeurs requises pour votre application et le choix des composants. Par exemple, la valeur de résistance pour le circuit LED peut être calculée approximativement comme suit:

Dans le cas où VLED est la tension avant à DEL, comme indiqué sur la fiche technique LED, au courant de conduite IOH (voir la section CARACTÉRISTIQUES ÉLECTRIQUES).

Supposons qu'une DEL typique à faible courant ait un $V_F = 1.8V$ à 5mA, la valeur de la résistance est:

Maintenant, restez sur le côté sûr et choisissez une résistance légèrement plus grande, comme 150 Ohm.

Si vous souhaitez conduire des LED de courant plus élevées, vous avez besoin d'un circuit comme le second, où vous placez la LED entre la résistance de sortie et le collecteur du transistor NPN.

Utilisez une broche comme entrée :

- ✓ Toutes les broches d'E / S sont des entrées avec un pull-up interne faible après la mise sous tension ou la réinitialisation. Vous pouvez également configurer la broche pour avoir un fort pull-up ou aucun pull-up du tout.
- ✓ Broche E / S connectée à une broche d'E / S d'interrupteur connectée Source 5V :
- ✓ (Ou capteur de commutation) (comme une broche de sortie MCU)
- ✓ Broche E / S avec entrée isolée (pour circuits de sécurité)
- ✓ Tous ces circuits supposent que la broche EasyVR a été configurée avec un pull-up interne (la valeur des composants passifs peut être ajustée pour tenir compte d'un pull-up faible ou fort).
- ✓ La désactivation de l'extraction interne pourrait être utilisée pour mettre la broche en état de haute impédance, par exemple pour simuler un port de sortie tri-state ou open-drain.
- ✓ Encore une fois, vous devez vous référer à la fiche technique du fabricant lors de l'interfaçage de tous les composants externes et pour calculer les valeurs des résistances requises ou d'autres composants passifs.

- ✓ Le module EasyVR comprend un chargeur de démarrage qui permet de mettre à jour le microprogramme et de télécharger de nouvelles tables de son à la mémoire embarquée.
- ✓ Le mode de démarrage est activé en maintenant le signal / XM à un niveau logique élevé à la mise sous tension ou à la réinitialisation. Cela peut être facilement effectué avec un cavalier prenant le signal sur une résistance d'extraction appropriée.
- ✓ Pour télécharger une mise à jour du microprogramme ou une table de son à EasyVR, allumez le module avec le cavalier fermé. Pour un fonctionnement normal, laissez simplement le cavalier ouvert. Ne modifiez pas la position du cavalier pendant que le module est déjà sous tension. Il est sécurisé de changer / niveau XM pendant que le module est réinitialisé (/ RST faible).

2.8.6 Le contact entre le module EASY VR et le microcontrôleur

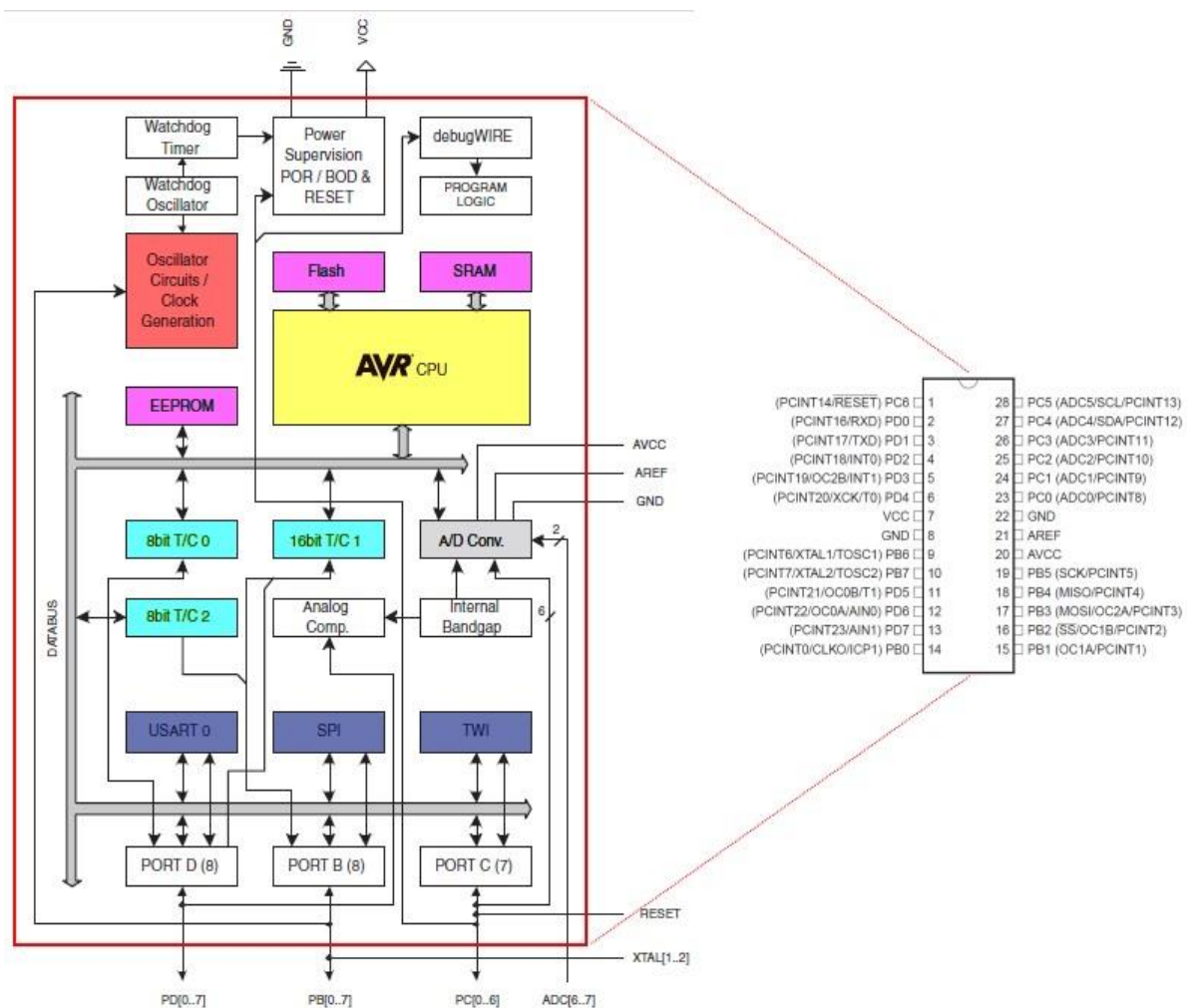


Figure 2.14 : EASY VR avec le microcontrôleur

2.9 Circuit de sélection du mode de démarrage

La valeur de la résistance d'extraction à utiliser dépend de la tension d'alimentation VCC. Pour la tension de la broche / XM lorsque le cavalier est fermé (court), la relation suivante se maintient (notez que vous avez un circuit diviseur de tension):

Maintenant, si vous voulez / XM être à 3V (logique haute) et résoudre pour R, vous obtenez:

Cela fait 100Ω pour 3.3V et environ 680Ω pour les alimentations 5V. D'autres types de circuits sont possibles, ce n'est qu'un exemple et l'un des plus simples à réaliser.

La communication avec le module EasyVR utilise une interface UART standard compatible avec les niveaux logiques TTL / CMOS de 3,3-5 V, en fonction de la tension d'alimentation VCC.

Une connexion typique à un hôte basé sur un MCU:

La configuration initiale à l'allumage est de 9600 bauds, 8 bits de données, pas de parité, 1 bit d'arrêt. Le débit en bauds peut être changé plus tard pour fonctionner dans la plage de 9600 à 115200 bauds.

Le protocole de communication utilise uniquement des caractères ASCII imprimables, qui peuvent être divisés en deux groupes principaux:

- ✓ Caractères de commande et d'état, respectivement sur les lignes TX et RX, choisis parmi les minuscules.
- ✓ Les arguments de commande ou les détails d'état, à nouveau sur les lignes TX et RX, couvrant la portée des majuscules.

Chaque commande envoyée sur la ligne TX, avec zéro ou plus d'octets d'argument supplémentaires, reçoit une réponse sur la ligne RX sous la forme d'un octet de statut suivi de zéro ou plus d'arguments.

Il y a un délai minimum avant chaque byte envoyé du module EasyVR à la ligne RX, qui est initialement réglé sur 20 ms et peut être sélectionné plus tard dans les plages 0 à 9 ms, 10 à 90 ms, 100 ms - 1 s. Cela explique les systèmes hôtes plus lents ou plus rapides et convient donc également à la communication en série basée sur le logiciel (bit-banging).

Étant donné que l'interface série EasyVR est également basée sur le logiciel, un délai très court pourrait être nécessaire avant de transmettre un caractère au module, surtout si l'hôte est très rapide, pour permettre à EasyVR de revenir à l'écoute d'un nouveau personnage.

La communication est guidée par l'hôte et chaque octet de la réponse à une commande doit être reconnu par l'hôte pour recevoir des données d'état supplémentaires, en utilisant le

caractère de l'espace. La réponse est annulée si un autre caractère est reçu et qu'il n'est pas nécessaire de lire tous les octets d'une réponse s'il n'est pas nécessaire.

Les combinaisons de commandes ou d'arguments non valides sont signalés par un octet d'état spécifique, que l'hôte doit être prêt à recevoir si la communication échoue. De même, un délai d'attente raisonnable devrait être utilisé pour récupérer des pannes inattendues.

Si l'hôte n'émet pas tous les arguments requis d'une commande, la commande est ignorée par le module, sans autre notification, et l'hôte peut commencer à envoyer une autre commande.

Le module passe automatiquement au mode veille économique le plus bas après la mise sous tension. Pour lancer une communication, envoyez un caractère pour réveiller le module.

Les messages de commande ou d'état envoyés sur la liaison série peuvent avoir un ou plusieurs arguments numériques dans la plage de -1 à 31, qui sont encodés en utilisant principalement des caractères dans la plage de lettres majuscules. Ce sont des constantes utiles pour traiter facilement les arguments:

Conclusion

Dans ce chapitre, nous avons présenté les différentes parties de notre stéthoscope numérique que nous avons développé et qui permettent la détection et mise en forme du signal phono cardiogramme PCG.

L'étude et le test de chaque partie du circuit permet de mieux choisir les composants nécessaire pour une meilleure détection du signal PCG.

Un intérêt particulier était porté au capteur ainsi que les circuits de mise en forme comme tenu que c'est la source de détection de l'information recherchée. Dans le chapitre suivant, la réalisation pratique de notre conception sera présentée, ainsi que la partie acquisition et traitement numérique du signal acquis.

Chapitre 3

Réalisation pratique du stéthoscope Numérique

Introduction

Dans ce chapitre nous intéressons à la description et la réalisation de notre stéthoscope numérique que nous avons réalisé

Ce chapitre sera consacré à la description des différents logiciels que nous avons utilisés au cours de notre travail.

Nous commençons tout d'abord par l'éditeur Arduino qui est une application de programmation, sur le quelle le programme est écrire, puis en passeras à la réalisation de notre système.

3.1 Environnement de programmation Arduino

La carte Arduino présente le noyau de la partie numérique .pour qu'on puisse la programmer, on doit disposer du logiciel compatible avec cette carte .ce logiciel est Arduino EDI (espace de développement Intégré), qui porte le nom même de la carte Arduino.

3.2 Description de logiciel ARDUINO

Le logiciel Arduino a pour fonction principales :

- ✓ Le pouvoir d'écrire et compiler le programme pour la carte Arduino.
- ✓ De se connecter avec la carte Arduino pour y transférer les programmes .es
- ✓ De communiquer avec la carte Arduino.

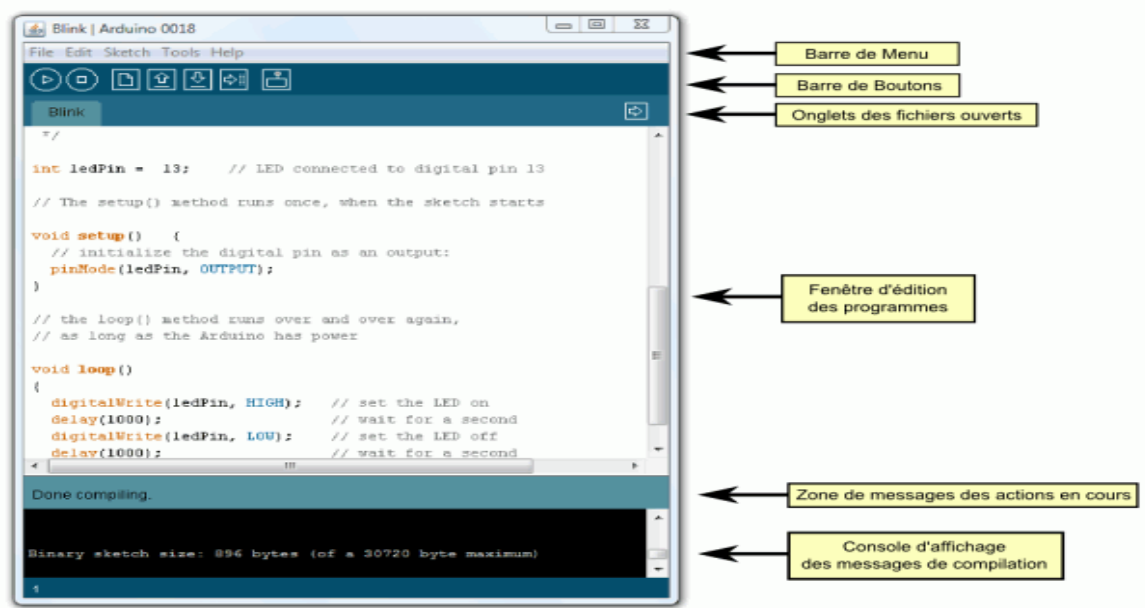


Figure 3.1 : Les différentes parties de la fenêtre principale de logiciel Arduino

Il comporte

- ✓ **Une BARRE DE MENUS** : comme pour tout logiciel une interface graphique (GUI).
- ✓ **Une BARRE DE BOUTONS** : cette barre nous donne un accès direct aux fonctions essentielles du logiciel, elle importe une simplicité d'utilisation.
- ✓ **Un EDITEUR (à coloration syntaxique)** : c'est dans cette espace ou on va écrire notre programme, il dispose aussi des onglets de navigation.
- ✓ **Une ZONE DE MESSAGES** : permet d'afficher et d'indiquer l'état des actions en cours (ex : vérification, téléversement).
- ✓ **Une CONSOLE TEXTE** : elle permet d'afficher les messages concernant le résultat de la compilation du programme (il nous indique s'il y a des erreurs).
- ✓ **Un MONITEUR SERIE** : ce moniteur est utilisé pour la visualisation des données (message, valeurs, caractères) transmises vers l'ordinateur via le câble USB connecté à la carte Arduino et d'envoyer vers la carte Arduino. Cette fonctionnalité facilite la mise au point des programmes, l'affichage de l'état des variables d'un programme de calculs ou mesures ainsi que le résultat de la conversions analogique –numérique : un élément essentiel pour améliorer, tester et corriger les programmes.

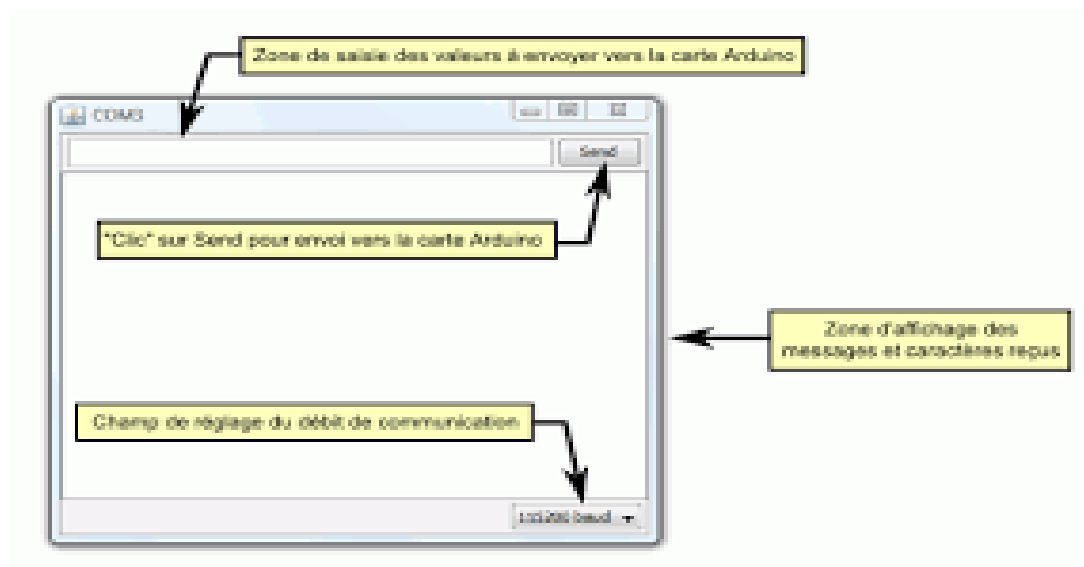


Figure 3.2 : Fenêtre associée au moniteur série

3.3 DESCRIPTION DE LA BARRE DES BOUTONS

La barre des boutons nous permet de vérifier la syntaxe, le transfert des programme, la création ouverture et la sauvegarde d'un écrit. elle permet aussi d'ouvrir le moniteur série. elle dispose de sept bouton ou icones :

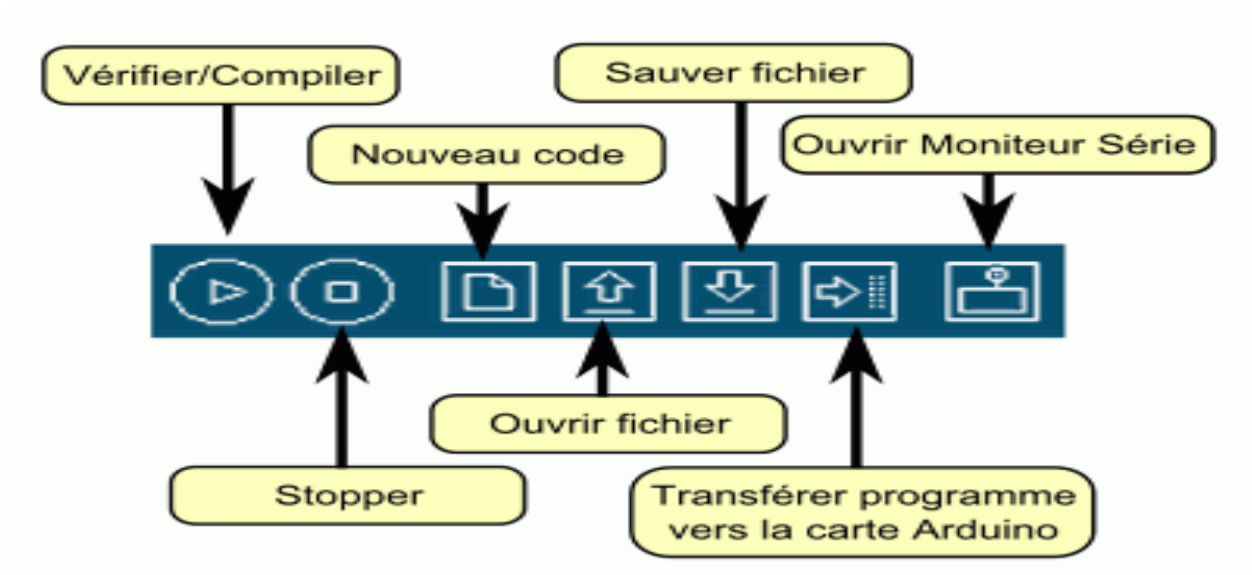


Figure 3.3 : Structure de la barre des boutons

- ✓ **VERIFIER/COMPILER** : ce bouton nous permet de vérifier s'il y a des erreurs dans le programme en sélectionnant la ligne où se trouve l'erreur.
- ✓ **STOP** : ce bouton oblige le moniteur série ou les autres boutons actifs à s'arrêter.
- ✓ **NOUVEAU** : en appuyant sur cette icône, une fenêtre d'édition vide s'ouvre pour la saisie d'un nouveau programme.
- ✓ **TRANSFERER OU TELEVERSER VERS LA CARTE** : ce bouton permet de compiler votre code et de le transférer vers la carte Arduino.
- ✓ **OUVRIR** : il suffit de cliquer sur ce bouton pour accéder à la liste de tous les programmes qui se trouvent dans le livre de programmes. cliquer sur l'un des programmes l'ouvre dans la fenêtre courante.
- ✓ **SAUEGARDER** : avant de quitter le programme saisi, il faut l'enregistrer, et ça peut se faire en cliquant sur cette icône.
- ✓ **MONITEUR SERIE** : pour ouvrir la fenêtre du moniteur série il suffit d'appuyer sur ce bouton.

3.4 DESCRIPTION DE LA BARRE DES MENUS

Le logiciel Arduino est équipé d'une barre dit des menus qui contient un ensemble de menus (d'où le Barre des menus) permettant de faire la gestion du programme :

Cette barre contient les icônes suivantes :

- ✓ **FILE (fichier)** : ce menu contient les différentes options de création, d'ouverture d'un exemple parmi l'exemple qui accompagnent le logiciel Arduino.
- ✓ **EDITE(EDITER)** : ce menu contient les options de copier/coller, sélection, et les options de rechercher.
- ✓ **SKETCH (PROGRAMME Ou séquence)** : ce menu contient les différentes fonctions de la barre de boutons, ainsi que les options d'ajout de bibliothèques ou de fichiers.
- ✓ **TOOLS(OUTILS)** : c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilise ainsi que la fonction de chargement du boot loader dans l'ATmega.
- ✓ **HELP(AIDE)** : ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino.

3.5 Les étapes de téléversement d'un programme vers la carte ARDUINO

Une fois le programme est saisi (première phase), une deuxième phase consiste à vérifier s'il y a des erreurs dans le programme, et pour la faire il suffit de cliquer sur le bouton de vérification se trouvant à l'extrémité gauche de la barre des boutons et attendre un petit moment.

Après cette vérification s'il y a des erreurs, un message en orange s'affiche au-dessus de l'onglet d'édition des programmes indiquant qu'elle se trouve (pour la correction des erreurs, il suffit de consulter l'aide disponible dans la barre des menus).

Une fois la correction des erreurs terminée, on passe à la phase de la sélection du type de la carte à programmer et le port série sur lequel est branchée cette carte, ce petit réglage peut se faire à l'aide du menu Tools (outils) une fois terminé il reste qu'à cliquer sur le bouton de téléversement qui se trouve dans la barre des boutons juste avant le bouton du moniteur série.

Si tout marche bien les deux LEDs Rx et Tx de la carte s'allument, sinon un message d'erreur s'affiche dans la console des messages de compilation.

À ce niveau la carte est programmée et prête pour l'exécution du programme, il suffit juste de la mettre sous tension et lui brancher les composants ou les éléments nécessaires pour le bon déroulement du processus à exécuter.

3.6 Syntaxe du langage Arduino

3.6.1 Commandes de structure du programme

Structure générale

`void setup()` (configuration-préparation)

`void loop()` (exécution).

Variables

`char` (variable 'caractère')

`int` (variable 'nombre entier')

`long` (variable 'nombre entier de très grande taille')

`string` (variable 'chaîne de caractères')

`array` (tableau de variables)

Niveaux logiques des connecteurs numériques

1.1. `HIGH` (état 1)

`LOW` (état 0)

`INPUT` (configuré en entrée)

`OUTPUT` (configuré en sortie)

Contrôle et conditions

1.1 `if` (si...)

1. `if...else` (si...alors...)

2. `for` (pour...)

3. `switch case` (dans le cas où...)

4. `while` (pendant que ...)

Opérations de comparaison

Contrôle et conditions

1.1 `if` (si...)

1. `if...else` (si...alors...)

2. `for` (pour...)

3. `switch case` (dans le cas où...)

4. `while` (pendant que ...)

- 2.1. `!=` (différent de)
- 2.2. `<` (inférieur à)
- 2.3. `>` (supérieur à)
- 2.4. `<=` (inférieur ou égal à)
- 2.5. `>=` (supérieur ou égal à)

Opérations booléennes

- 3.1. `&&` (et)
- 3.2. `||` (ou)
- 3.3. `!` (et pas)

Autres commandes

- `//` (commentaire simple ligne)
- `/* */` (commentaire multi-lignes)
- `#define` (donner une valeur à un nom)

Fonctions

Entrées-sorties numériques

- (a) `pinMode(broche, état)` (configuration des broches)
- `digitalWrite(broche, état)` (écrire un état sur une broche num.)
- `digitalRead(broche)` (lire un état sur une broche num.)
- `unsigned long pulseIn(broche, état)` (lire une impulsion sur une broche num.)

Entrées analogiques

- 1. `int analogRead(broche)` (lire la valeur d'une broche ana.)
- 2. `analogWrite(broche, valeur)` (PWM : écrire une valeur analogique sur les broches 9, 10 ou 11)

Gestion du temps

- `unsigned long millis()` (temps de fonctionnement du programme)
- `delay(ms)` (attente, en millisecondes)

Nombres aléatoires

1. `randomSeed(seed)` (aléatoire 'pilote')
2. `long random(max)` (aléatoire à partir de telle valeur)
3. `long random(min, max)` (aléatoire entre deux valeurs)

Communications série entre Arduino et autres machines ou ordinateur

1. `Serial.begin(speed)` (configuration de la vitesse de communication Série)
2. `Serial.available()` (donne combien de caractères disponibles dans la zone tampon Série)
3. `Serial.read()` (lit les données Série)
4. `Serial.print(data)` (envoi des données Série)
`Serial.println(data)` (envoi des données Série suivies de caractères spécifiques).
- 2.6. `!=` (différent de)
- 2.7. `<` (inférieur à)
- 2.8. `>` (supérieur à)
- 2.9. `<=` (inférieur ou égal à)
- 2.10. `>=` (supérieur ou égal à)

Opérations booléennes

- 3.4. `&&` (et)
- 3.5. `||` (ou)
- 3.6. `!` (et pas)

Autres commandes

`//` (commentaire simple ligne)
`/* */` (commentaire multi-lignes)
`#define` (donner une valeur à un nom)

Fonctions

Entrées-sorties numériques

- (b) `pinMode(broche, état)` (configuration des broches)
`digitalWrite(broche, état)` (écrire un état sur une broche num.)
`digitalRead(broche)` (lire un état sur une broche num.)
`unsigned long pulseIn(broche, état)` (lire une impulsion sur une broche num.)

int **analogRead**(broche) (lire la valeur d'une broche ana.)

3. **analogWrite**(broche, valeur) (PWM : écrire une valeur analogique sur les broches 9, 10 ou 11)

Gestion du temps

unsigned long **millis**() (temps de fonctionnement du programme)

delay(ms) (attente, en millisecondes)

Nombres aléatoires

4. **randomSeed**(seed) (aléatoire 'pilote')
5. long **random**(max) (aléatoire à partir de telle valeur)
6. long **random**(min, max) (aléatoire entre deux valeurs)

Communications série entre Arduino et autres machines ou ordinateur

5. **Serial.begin**(speed) (configuration de la vitesse de communication Série)
6. **Serial.available**() (donne combien de caractères disponibles dans la zone tampon Série)
7. **Serial.read**() (lit les données Série)
8. **Serial.print**(data) (envoi des données Série)
Serial.println(data) (envoi des données Série suivies de caractères spécifiques.)

3.7 Le circuit réalisé

Le circuit électronique du stéthoscope numérique que nous avons réalisé est développé sur la base du schéma bloc présenté dans la figure 2 .1 dans le précédant chapitre.

Ainsi l'assemblage des différents circuits précédemment décrits dans le chapitre 2 dans un même boîtier permet d'avoir le circuit finale de notre stéthoscope numérique.

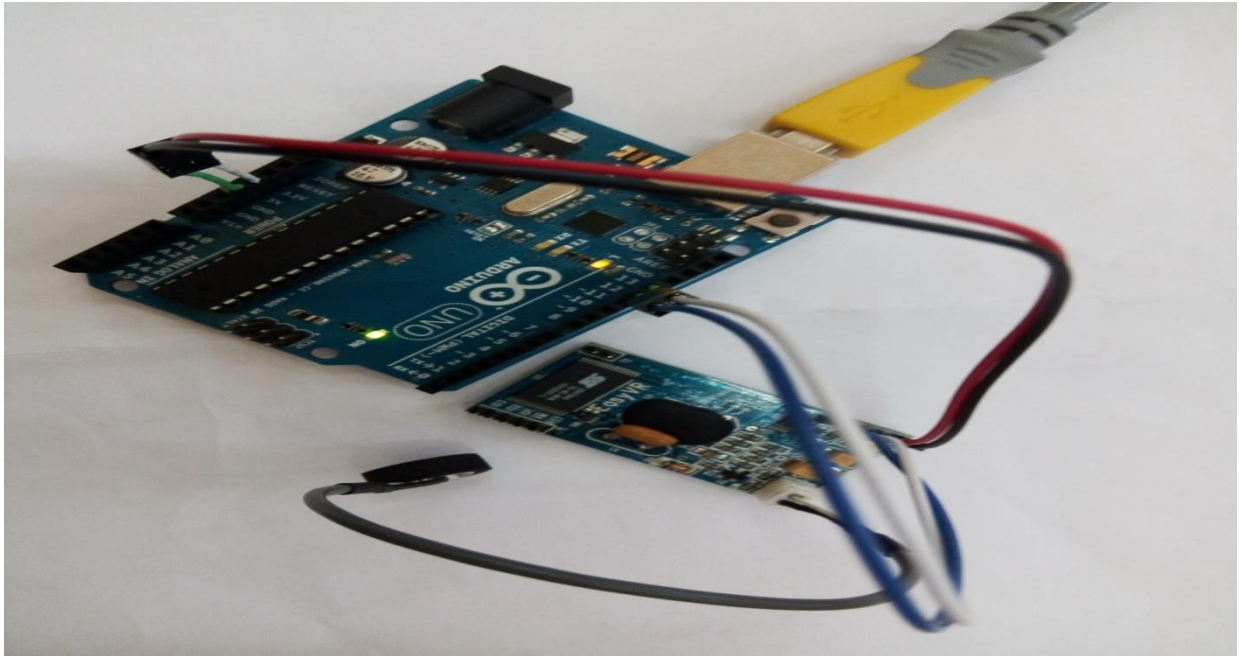


Figure 3.4 : Branchement de module EASY VR avec la carte Arduino Uno

3.8 Circuit réalisée avec stéthoscope



Figure : 3.5 : Circuit réalisée avec Stéthoscope

3.9 Test de validation de notre réalisation

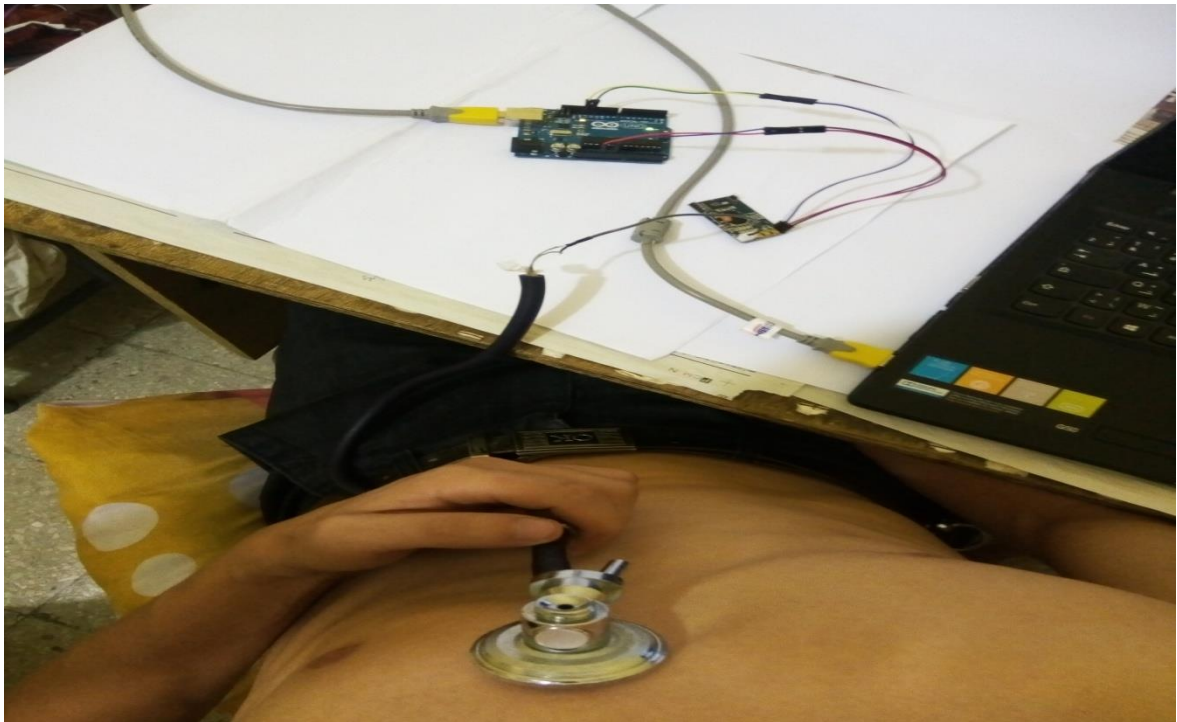


Figure 3.6: Test de validation de notre système

3.10 EASY VR Cammander

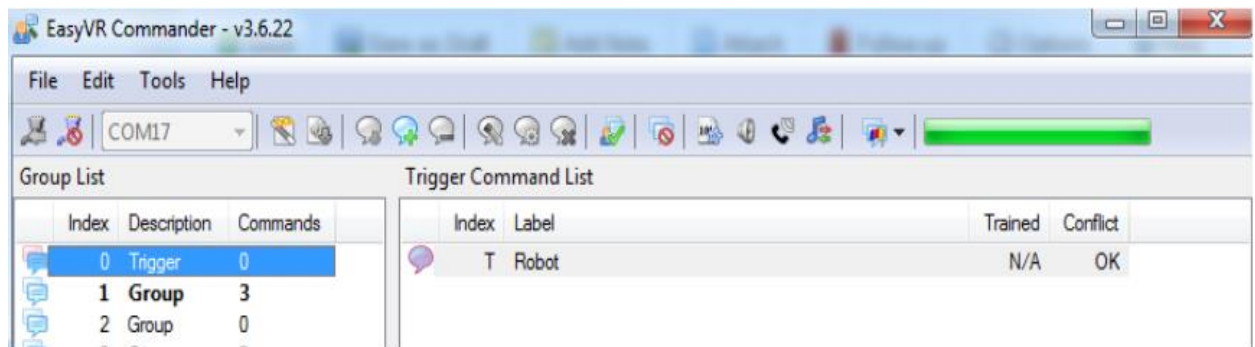
Cette note d'application doit indiquer comment fonctionne le flux de travail commun avec EasyVR Shields and Modules. Ce document ne remplace pas le manuel, il passe par le haut-parleur standard Processus de création de commande pour l'intégration d'une action que le contrôleur effectue si une commande parlée est reconnue. C'est l'exemple le plus simple, l'action réalisable se limite aux cartes contrôleur Arduino Et votre imagination. Que ce soit que vous envisagez de faire, commencez à changer les leds en premier. Dans ce cas, nous changeons 3 Sorties HIGH. S'amuser, Votre équipe de support Veear

Comment obtenir un soutien Manuels / Notes d'application / Code de démo Un mode d'emploi qui comprend toutes les informations nécessaires pour démarrer est fourni dans la section de téléchargement. Tous les logiciels nécessaires sont installés avec EasyVR Commander

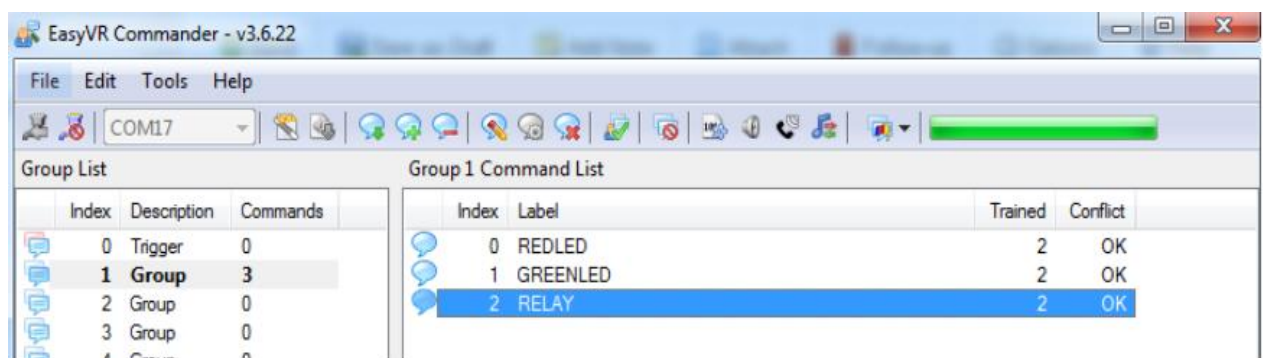
Comment obtenir un soutien Manuels / Notes d'application / Code de démo Un mode d'emploi qui comprend toutes les informations nécessaires pour démarrer est fourni dans la section de téléchargement. Tous les logiciels nécessaires sont installés avec EasyVR Commander

Des bibliothèques et des exemples supplémentaires pour Arduino sont également disponibles. Veuillez vérifier la section de téléchargement pour plus de détails. FAQ Si vous rencontrez des problèmes qui ne peuvent être résolus avec le manuel, nous vous recommandons de vérifier le service d'assistance pour Des solutions et des problèmes qui ont été signalés auparavant. Un moyen pratique de rechercher des solutions est d'utiliser le Fonction de recherche O page principale Helpdesk cliquez ici. O Aide spécifique EasyVR cliquez ici. O Aide spécifique SmartVR Cliquez ici. Contact N'hésitez pas à nous contacter pour toute question, requête ou suggestion. Si votre question concerne les techniques Support ou dépannage pour l'un de nos produits, nous vous prions de vérifier d'abord le mode d'emploi, les FAQ et Helpdesk pour une solution possible. Si vous ne trouvez pas une solution existante dans ces ressources, contactez-nous En utilisant le formulaire de contact. Plus vous fournissez des détails, nous vous offrons un meilleur support. Pour nous contacter, veuillez suivre ce lien vers notre formulaire de contact (vous serez redirigé vers le service d'assistance de TIGAL)

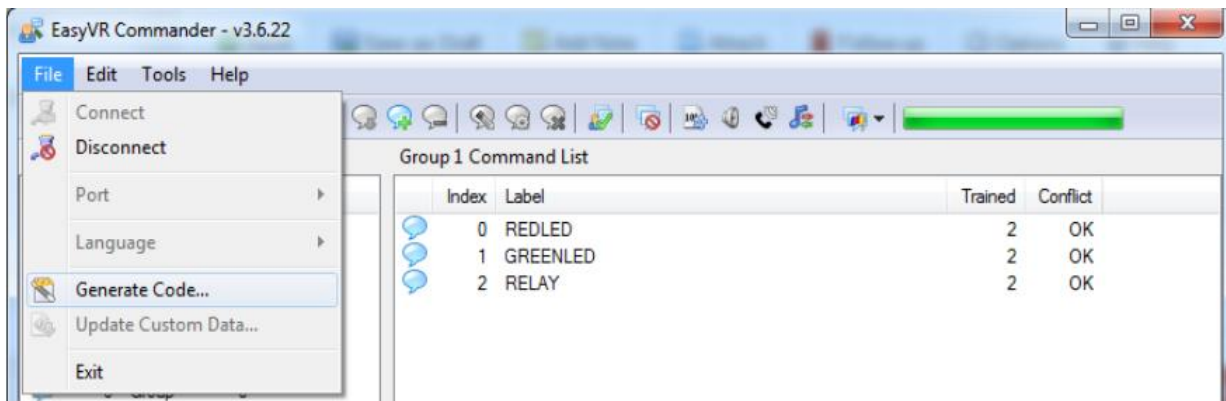
3.10.1 Vérification EASY VR Commander



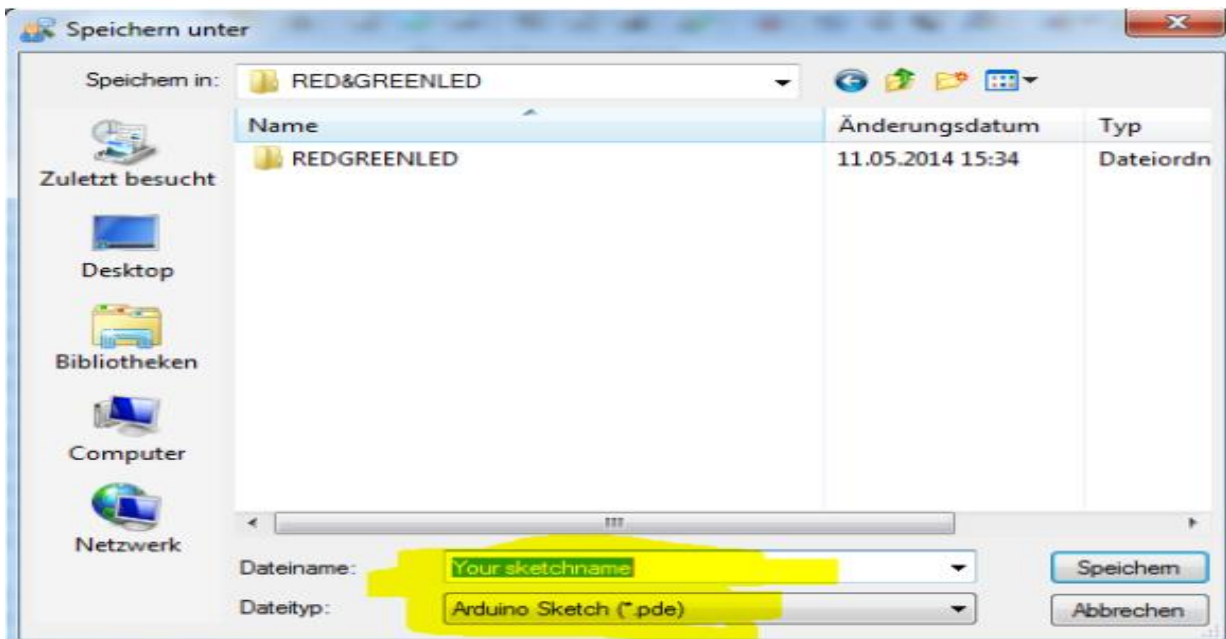
3.10.2 Création de trois commandes dans un groupe



3.10.3 Le code de la commande pour arduino



3.10.4 Enregistrement de code comme arduino



3.11. Résultats et interprétation

Ainsi le signal obtenu et visualisé sur un oscilloscope digital à mémoire est illustré sur la figure suivante :



Figure 3.7 : L'allure du signal obtenu visualisé sur oscilloscope

Un zoom sur ce même signal avec un changement du calibre nous permet une meilleure visualisation de ce signal comme est illustré dans la figure 3.7 suivante :

La qualité du signal PCG issu sur le circuit final réalisé visualisé sur un oscilloscope digital est illustrée dans la figure 3.8



Figure 3.8 : signal PCG détecté par notre stéthoscope Numérique

Sur ces figure, nous remarquons très clairement les deux composants essentiels du signal phono cardiogramme PCG, à savoir les bruits B1 et B2 déjà vu dans le chapitre précédent.

Conclusion

Dans ce chapitre, nous avons présenté la partie analogique et digitale de notre stéthoscope

Numérique développé permettant la mise en forme et l'acquisition du signal Phono cardiogramme PCG.

Un mode d'acquisition de données est établi entre notre stéthoscope digitale et l'ordinateur afin d'assurer un transfert simple et efficace du signal.

Nous avons aussi présenté les résultats de mesure et d'acquisition de signaux PCG sur notre circuit. Les mesures ont été effectuées sur plusieurs sujets.

Plusieurs programme ont été développé dans l'objectif de proposer aux médecins des outils simples et efficaces pour l'enregistrement, le traitement et l'analyse du signal PCG.

Conclusion générale

Dans notre projet de fin d'études, nous avons réalisé un stéthoscope numérique qui permet de l'acquisition des bruits cardiaques sous forme d'un signal phono cardiogramme PCG et aussi de mesurer le rythme cardiaque.

Le circuit général de notre stéthoscope réalisé consiste en deux parties élémentaires. Une première partie analogique et une autre numérique.

La partie analogique comporte un microphone, des circuits analogique de mise en forme ; en particulier un capture Easy vr branché à la sortie du microphone afin d'obtenir un signal exploitable par la suite.

La partie numérique a été consacrée à la numérisation du signal à l'aide d'une carte ARDUINO et la transmission des données vers un micro-ordinateur, ou la visualisation des données a été faite en utilisant l'exploitation de la carte son de l'ordinateur comme une carte dédié à l'acquisition de notre signal PCG vu qu'il est représenté des caractéristiques signaux sont des trames.

Ce travail a été très bénéfique pour nous car il nous a permis d'approfondir nos connaissances sur les captures, les électrodes l'amplificateur, le filtrage et bien d'autres aspects des fonctions électronique en instrumentation biomédicale. Un traitement numérique ultérieur est alors possible en vue d'extraire des paramètres en relation avec le diagnostic médical associé à l'examen clinique effectué par le médecin.

Plusieurs mesures ont été effectuées sur différents sujets. Ces mesures se présentent sous forme d'un dossier médical qui contient en plus du signal PCG, des informations personnelles sur le patient. Ces mesures sont enregistrées sous forme d'une base de données. Un autre programme a été aussi développé et qui est dédié au calcul du rythme cardiaque à partir du signal PCG en exploitant l'énergie du signal.

Les résultats que nous avons obtenus sont prometteurs à la fois dans la partie analogique que pour la partie numérique. Dans l'avenir, nous envisageons de mettre en place des méthodes pour améliorer le rapport signal-sur-bruit en utilisant des filtres numériques et de traitement avancé du signal, afin d'obtenir une segmentation automatique et d'extraire plus de paramètres d'intérêt clinique pour les cardiologues.

Annexes

TL081, TL081A, TL081B, TL082, TL082A, TL082B
 TL084, TL084A, TL084B
JFET INPUT OPERATIONAL AMPLIFIERS
 SLOS081G – FEBRUARY 1977 – REVISED SEPTEMBER 2004

- D Low Power Consumption
- D Wide Common-Mode and Differential Voltage Ranges
- D Low Input Bias and Offset Currents
- D Output Short-Circuit Protection
- D Low Total Harmonic Distortion . . . 0.003% Typ
- D High Input Impedance . . . JFET-Input Stage
- D Latch-Up-Free Operation
- D High Slew Rate . . . 13 V/μs Typ
- D Common-Mode Input Voltage Range Includes V_{CC}+

description/ordering information

The TL08x JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient. Offset adjustment and external compensation options are available within the TL08x family.

The C-suffix devices are characterized for operation from 0°C to 70°C. The I-suffix devices are characterized for operation from –40°C to 85°C. The Q-suffix devices are characterized for operation from –40°C to 125°C. The M-suffix devices are characterized for operation over the full military temperature range of –55°C to 125°C.

ORDERING INFORMATION

TJ	VIOMax AT 25°C	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	15 mV	PDIP (P)	Tube of 50	TL081CP	TL081CP
			Tube of 50	TL082CP	TL082CP
		PDIP (N)	Tube of 25	TL084CN	TL084CN
		SOIC (D)	Tube of 75	TL081CD	TL081C
			Reel of 2500	TL081CDR	
			Tube of 75	TL082CD	TL082C
			Reel of 2500	TL082CDR	
			Tube of 50	TL084CD	TL084C
			Reel of 2500	TL084CDR	
		SOP (PS)	Reel of 2000	TL081CPSR	T081
		SOP (NS)	Reel of 2000	TL082CPSR	T082
			Reel of 2000	TL084CNSR	TL084
		TSSOP (PW)	Tube of 150	TL082CPW	T082
			Reel of 2000	TL082CPWR	
			Tube of 90	TL084CPW	T084
			Reel of 2000	TL084CPWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



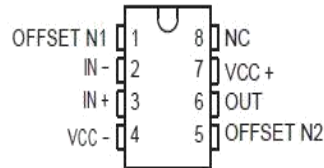
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated
 On products compliant to MIL PRF 38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

TL081, TL081A, TL081B, TL082, TL082A, TL082B
 TL084, TL084A, TL084B
 JFET INPUT OPERATIONAL AMPLIFIERS

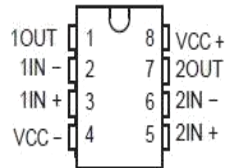
SLOS081G - FEBRUARY 1977 - REVISED SEPTEMBER 2004

TL081, TL081A, TL081B
 D, P, OR PS PACKAGE
 (TOP VIEW)

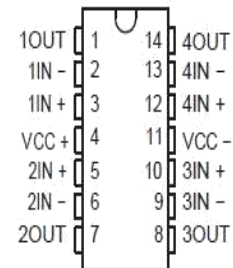


NC - No internal connection

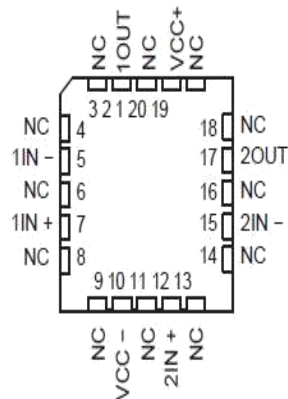
TL082, TL082A, TL082B
 D, JG, P, PS, OR PW PACKAGE
 (TOP VIEW)



TL084, TL084A, TL084B
 D, J, N, NS, OR PW PACKAGE
 (TOP VIEW)

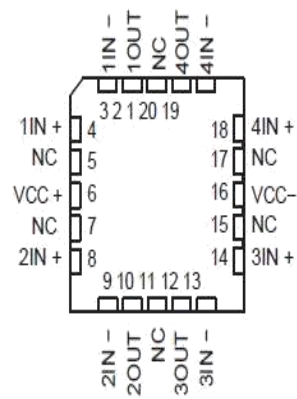


TL082M... FK PACKAGE
 (TOP VIEW)



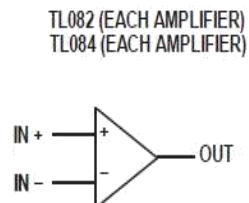
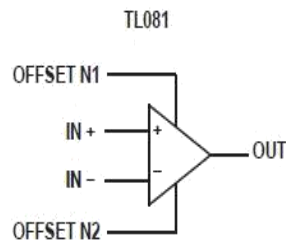
NC - No internal connection

TL084M... FK PACKAGE
 (TOP VIEW)



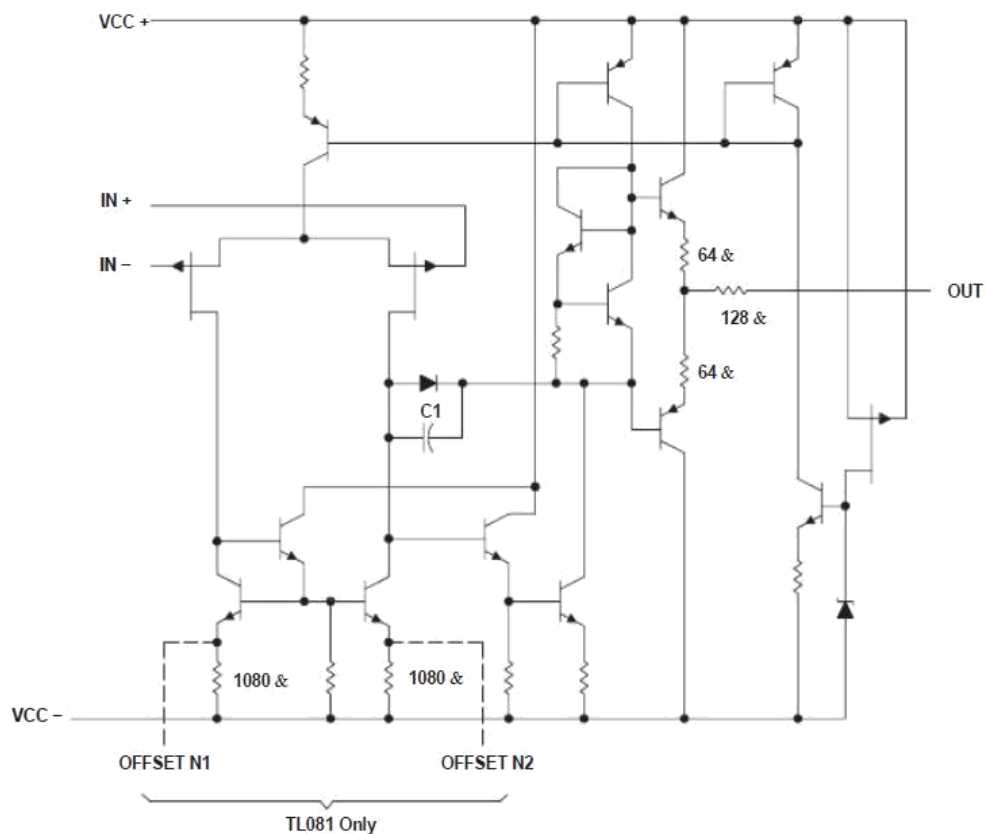
NC - No internal connection

symbols



TL081, TL081A, TL081B, TL082, TL082A, TL082B
 TL084, TL084A, TL084B
 JFET INPUT OPERATIONAL AMPLIFIERS
 SLOS081G – FEBRUARY 1977 – REVISED SEPTEMBER 2004

schematic (each amplifier)



Component values shown are nominal.

TL081, TL081A, TL081B, TL082, TL082A, TL082B
 TL084, TL084A, TL084B
 JFET INPUT OPERATIONAL AMPLIFIERS
 SLOS081G – FEBRUARY 1977 – REVISED SEPTEMBER 2004

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

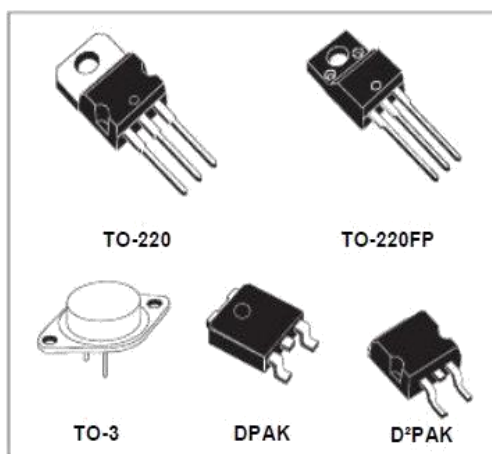
	TL08_C TL08_AC TL08_BC	TL08_I	TL084Q	TL08_M	UNIT
Supply voltage, V_{CC} (see Note 1)	18	18	18	18	V
Supply voltage V_{CC}^- (see Note 1)	-18	-18	-18	-18	V
Differential input voltage, V_{ID} (see Note 2)	± 30	± 30	± 30	± 30	V
Input voltage, V_I (see Notes 1 and 3)	± 15	± 15	± 15	± 15	V
Duration of output short circuit (see Note 4)	Unlimited	Unlimited	Unlimited	Unlimited	
Continuous total power dissipation	See Dissipation Rating Table				
Operating free-air temperature range, T_A	0 to 70	- 40 to 85	- 40 to 125	- 55 to 125	°C
Package thermal impedance, θ_{JA} (see Notes 5 and 6)	D package (8-pin)	97	97		°C/W
	D package (14-pin)	86	86		
	N package (14-pin)	76	76		
	NS package (14-pin)	80			
	P package (8-pin)	85	85		
	PS package (8-pin)	95	95		
	PW package (8-pin)	149			
	PW package (14-pin)	113	113		
Operating virtual junction temperature	150	150	150	150	°C
Case temperature for 60 seconds, T_C	FK package			260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds	J or JG package			300	°C
Storage temperature range, T_{stg}	- 65 to 150	- 65 to 150	- 65 to 150	- 65 to 150	°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. All voltage values, except differential voltages, are with respect to the midpoint between V_{CC}^+ and V_{CC}^- .
 2. Differential voltages are at IN^+ with respect to IN^- .
 3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
 4. The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.
 5. Maximum power dissipation is a function of $T_J(max)$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
 6. The package thermal impedance is calculated in accordance with JESD 51-7.

DISSIPATION RATING TABLE

PACKAGE	$T_A = 25^\circ\text{C}$ POWER RATING	DERATING FACTOR	DERATE ABOVE T_A	$T_A = 70^\circ\text{C}$ POWER RATING	$T_A = 85^\circ\text{C}$ POWER RATING	$T_A = 125^\circ\text{C}$ POWER RATING
D (14 pin)	680 mW	7.6 mW/°C	60°C	604 mW	490 mW	186 mW
FK	680 mW	11.0 mW/°C	88°C	680 mW	680 mW	273 mW
J	680 mW	11.0 mW/°C	88°C	680 mW	680 mW	273 mW
JG	680 mW	8.4 mW/°C	69°C	672 mW	546 mW	210 mW



Description

The L78xx series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3, D²PAK and DPAK packages and several fixed output voltages, making it useful in a wide range of applications.

These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

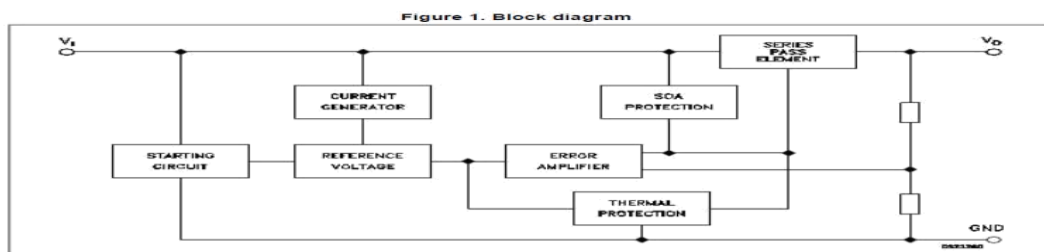
Features

- Output current up to 1.5 A
- Output voltages of 5; 6; 8; 8.5; 9; 12; 15; 18; 24 V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection
- 2 % output voltage tolerance (A version)
- Guaranteed in extended temperature range (A version)

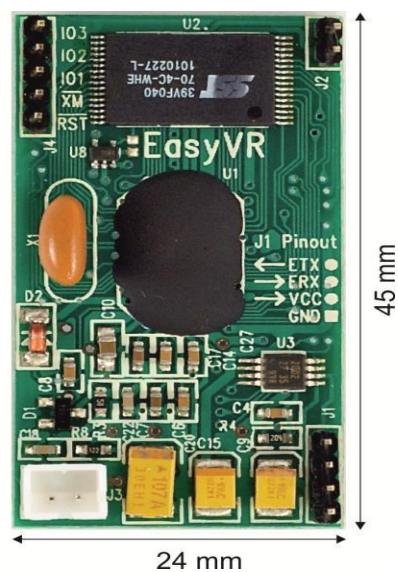
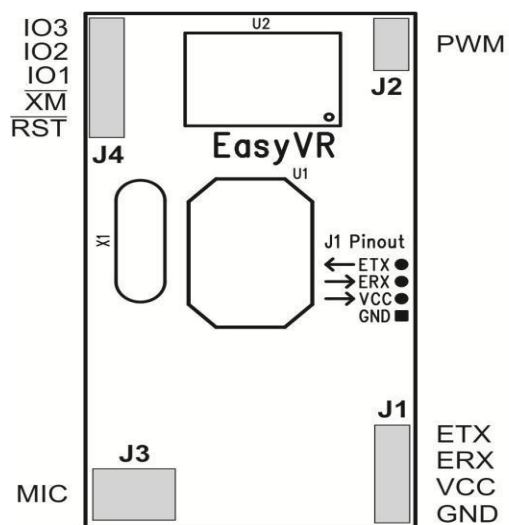
Table 1. Device summary

Part numbers			
L7805	L7806AC	L7809AB	L7815AB
L7805C	L7808C	L7809AC	L7815AC
L7805AB	L7808AB	L7812C	L7818C
L7805AC	L7808AC	L7812AB	L7824C
L7806C	L7885C	L7812AC	L7824AB
L7806AB	L7809C	L7815C	L7824AC

1 Diagram



Physical dimensions and pin assignment



2 Pin configuration

Figure 2. Pin connections (top view)

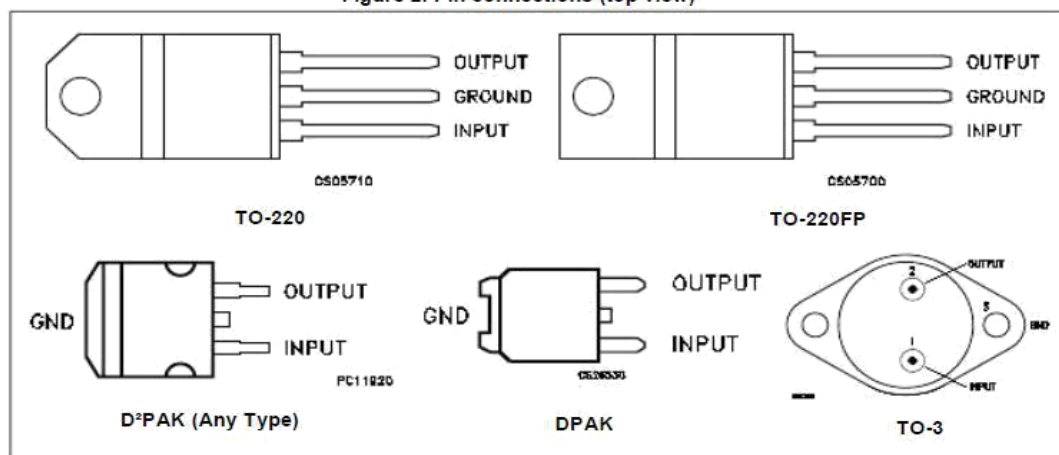
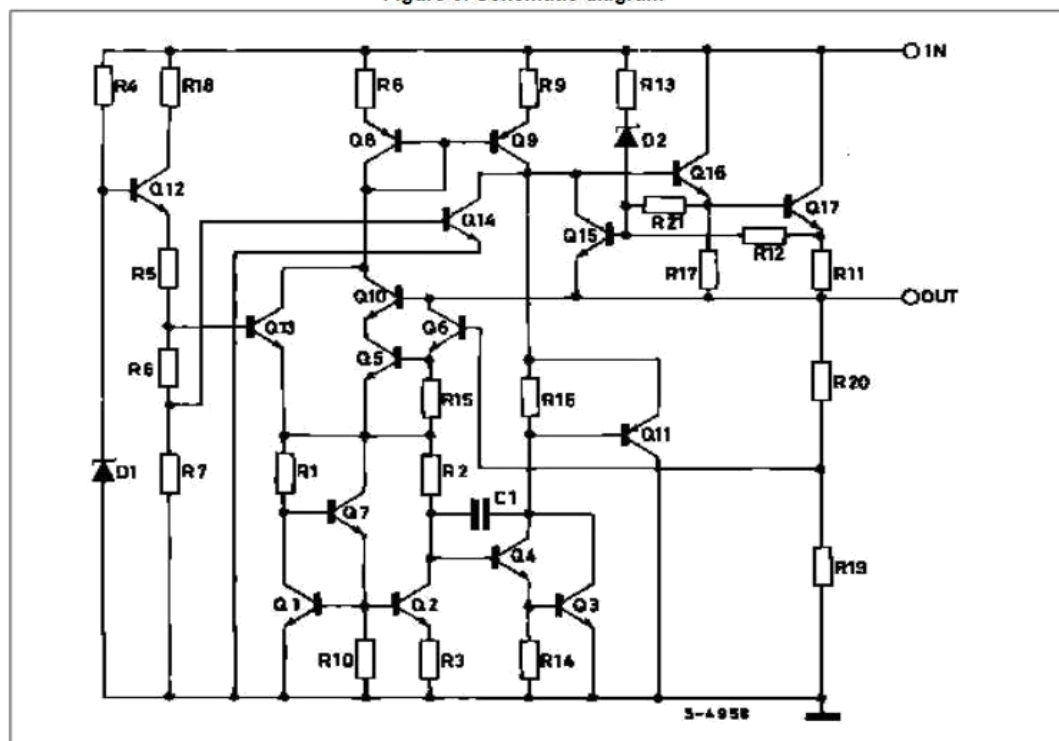


Figure 3. Schematic diagram



3 Maximum ratings

Table 2. Absolute maximum ratings

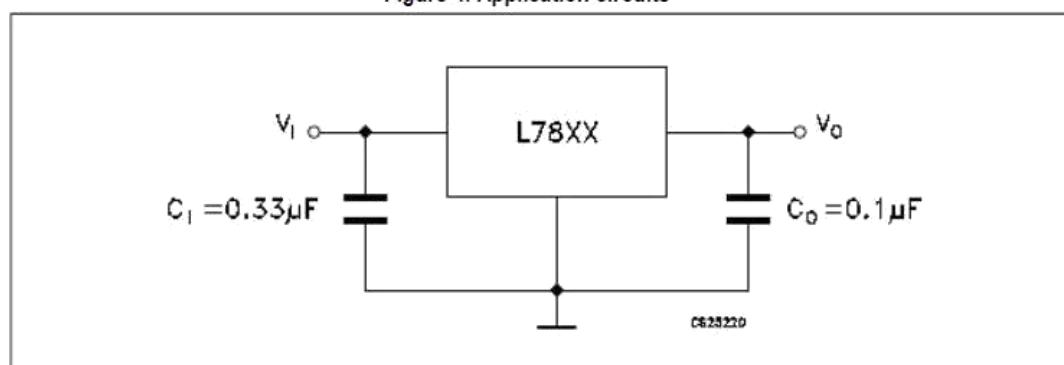
Symbol	Parameter		Value	Unit
V_I	DC input voltage	for $V_O = 5$ to 18 V	35	V
		for $V_O = 20, 24$ V	40	
I_O	Output current		Internally limited	
P_D	Power dissipation		Internally limited	
T_{STG}	Storage temperature range		-65 to 150	°C
T_{OP}	Operating junction temperature range	for L78xx	-55 to 150	°C
		for L78xxC, L78xxAC	0 to 125	
		for L78xxAB	-40 to 125	

Note: Absolute maximum ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 3. Thermal data

Symbol	Parameter	D ² PAK	DPAK	TO-220	TO-220FP	TO-3	Unit
R_{thJC}	Thermal resistance junction-case	3	8	5	5	4	°C/W
R_{thJA}	Thermal resistance junction-ambient	62.5	100	50	60	35	°C/W

Figure 4. Application circuits



Bibliographie

1. <http://www.ncbi.nlm.nih.gov/pubmed/20338798> [Accessed March 15, 2013].
2. Yan, Z. et al., 2010a. The moment segmentation analysis of heart sound pattern.
3. A.Waller :An introduction to human physiology.Longmans Green ;1893.
4. Bai'09] Y. Bai, C. Yeh. "Design and Implementation of a Remoted Embeedded DSP Stethoscope with a Method for Judging Heart Murmur," in International Instrumentation and Measurement
5. Computer methods and programs in biomedicine, 98(2), pp.140–50. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19854530> [Accessed March 6, 2013].
6. Cra'88+ Craig E. "Should Auscultation be Rehabilitated", New England Journal of
7. Deb'04+ S. M. DEBBAL, "Analyse du signal phonocardiogramme par application des transformees d'ondelette", Université A.B.Bekr Belkaid,Tlemcen, Thèse de doctorat, Juin 2004.
8. Dje'14+ Djebbari A. "Synthesis of time frequency analysis methods applied of phonocardiogram signal" Université de Tlemcen Doctorat thesis Septembre 2014
9. <http://linkinghub.elsevier.com/retrieve/pii/S0030402613000120> [Accessed May 14, 2013].
10. Jia'06+ Jiang Z, Choi S, "A cardiac sound characteristic waveform method for in-home heart disorder monitoring with electric stethoscope", Expert Systems with Applications 31, 2006, pp. 286-298.
11. M.Nachef :Conception et réalisation d'un système de télésurveillance médicale a base d'une carte ARDUINO Transmission bluetooth.UMMTO .2013
12. Oma'09+ M.Omari Tahar, « Etude de degré de sévérité pathologique des sténoses aortiques », thèse de magister, Université de Tlemcen, p 101, 2009.
13. Pel'04+ A. N. Pelech, "The physiology of cardiac auscultation," Pediatr Clin North Am, vol. 51, pp. 1515-1535, 2004
14. W.J.Tompkins : « biomedical digital signal processing ».prentice- hall ;1993.
15. Yan, Z. et al., 2010b. The moment segmentation analysis of heart sound pattern.
16. Yeh, J.-R. et al., 2010. Investigating fractal property and respiratory modulation of human heartbeat time series using empirical mode decomposition. Medical engineering & physics, 32(5), pp.490–6. Available at:

17. Yi, L., 2008. Portable Bluetooth Visual Electrical Stethoscope research. In
18. Zulkifli, N.S.A., Harun, F.K.C. & Azahar, N.S., XBee Wireless Sensor Networks for Heart Rate Monitoring in Sport Training. , (in C).
19. <http://www.sciencedirect.com/science/article/pii/S016926070900279X>.
20. Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on. pp. 634–636. Computer Methods and Programs in Biomedicine, 98(2), pp.140–150.
21. Medicine 1988, 318:1611-3
22. Technology Conference. May 2009

Résumé

L'utilisation des stéthoscopes Numérique est une nouvelle tendance dans le domaine de l'auscultation cardiaque. Les bruits cardiaques, engendrés par les fluctuations de la vitesse du sang et des vibrations de la structure musculaire, constituent un signal phonocardiographie (PCG) qui très important dans le diagnostic primaire des maladies cardiaques. Ce travail présente un stéthoscope numérique que nous avons développé. Le matériel typique du stéthoscope peut être divisé en parties analogique et numérique, respectivement. Le premier se compose d'un microphone, et d'un capture easy vr. Le second contient une carte d'acquisition du signal avec le logiciel de la carte arduino Uno, pour l'enregistrement et de traitement du signal PCG. Les résultats obtenus par ce système sont des signaux de bonne qualité visualisés sur oscilloscope puis numérisés et visualisé sur un PC grâce aux programmes PCGrecord et PCGanalysis que nous avons développée. Le signal PCG numérisé est enregistré comme fichier de données sous format WAV pour constitué une base de signaux PCG.

Abstract

The use of electronic stethoscopes is a new trend in the field of cardiac auscultation. Heart sounds, caused by fluctuations in blood velocity and vibration of muscle structure, are very important in the primary diagnosis of heart disease by using of the phonocardiographic signal (PCG). This paper presents an electronic stethoscope that we have developed. The typical hardware stethoscope can be divided into analog and digital parts, respectively. The first consists of a microphone, an amplifier stage and a filtering stage. The second contains a signal acquisition card with software control, recording and processing PCG signal. The results obtained by this system are of good quality signals displayed on oscilloscope and then digitalized and viewed on a PC using the "PCGrecord" and "PCGanalysis" programs. The digitized PCG signal is stored as a data file in WAV format to collect a PCG signals database.

Programme Easy vr

```
// initialize EasyVR

while (!easyvr.detect())

{

    pcSerial.println(F("EasyVR not detected!"));

    for (int i = 0; i < 10; ++i)

    {

        if (pcSerial.read() == EasyVR::BRIDGE_ESCAPE_CHAR)

            goto bridge;

        delay(100);

    }

}

pcSerial.print(F("EasyVR detected, version "));

pcSerial.print(easyvr.getID());

if (easyvr.getID() < EasyVR::EASYVR3)

    easyvr.setPinOutput(EasyVR::IO1, LOW); // Shield 2.0 LED off

if (easyvr.getID() < EasyVR::EASYVR)

    pcSerial.print(F(" = VRbot module"));

else if (easyvr.getID() < EasyVR::EASYVR2)

    pcSerial.print(F(" = EasyVR module"));

else if (easyvr.getID() < EasyVR::EASYVR3)

    pcSerial.print(F(" = EasyVR 2 module"));

else

    pcSerial.print(F(" = EasyVR 3 module"));

pcSerial.print(F(", FW Rev."));

pcSerial.println(easyvr.getID() & 7);
```

```

easyvr.setDelay(0); // speed-up replies

if (easyvr.getID() >= EasyVR::EASYVR3_1)
{
    if (!easyvr.checkMessages() && easyvr.getError() == EasyVR::ERR_CUSTOM_INVALID)
    {
        pcSerial.print(F("Message recovery needed, please wait..."));

        if (easyvr.fixMessages())

            pcSerial.println(F(" done!"));

        else

            pcSerial.println(F(" failed!"));

    }
}

```

```

pcSerial.print(F("Recorded messages:"));

if (easyvr.getID() >= EasyVR::EASYVR3_1)
{
    pcSerial.println();

    for (int8_t idx = 0; idx < 32; ++idx)
    {
        int8_t bits = -1; int32_t len = 0;

        if (easyvr.dumpMessage(idx, bits, len) && (bits == 0))

            continue; // skip empty

        pcSerial.print(idx);

        pcSerial.print(F(" = "));

        if (bits < 0)

            pcSerial.println(F(" has errors"));

        else

```

```

{
    pcSerial.print(bits);

    pcSerial.print(F(" bits, size "));

    pcSerial.println(len);
}
}
}

else

    pcSerial.println(F("n/a"));

    easyvr.setTimeout(5);

    lang = EasyVR::ENGLISH;

    easyvr.setLanguage(lang);

    // use fast recognition

    easyvr.setTrailingSilence(EasyVR::TRAILING_MIN);

    easyvr.setCommandLatency(EasyVR::MODE_FAST);

    int16_t count = 0;

    pcSerial.print(F("Sound table: "));

    if (easyvr.dumpSoundTable(name, count))
    {

        pcSerial.println(name);

        pcSerial.print(F("Sound entries: "));

        pcSerial.println(count);

    }

    else

        pcSerial.println(F("n/a"));

```

```
pcSerial.print(F("Custom Grammars: "));

grammars = easyvr.getGrammarsCount();

if (grammars > 4)
{
    pcSerial.println(grammars - 4);

    for (set = 4; set < grammars; ++set)
    {
        pcSerial.print(F("Grammar "));

        pcSerial.print(set);

        uint8_t flags, num;

        if (easyvr.dumpGrammar(set, flags, num))
        {
            pcSerial.print(F(" has "));

            pcSerial.print(num);

            if (flags & EasyVR::GF_TRIGGER)

                pcSerial.println(F(" trigger"));

            else

                pcSerial.println(F(" command(s)"));

        }

        else

        {

            pcSerial.println(F(" error"));

            continue;

        }

        for (int8_t idx = 0; idx < num; ++idx)
        {
```



```

pcSerial.print(idx);

    pcSerial.print(F(" = "));

    if (!easyvr.getNextWordLabel(name))

        break;

    pcSerial.println(name);

}

}

}

else

    pcSerial.println(F("n/a"));


if (easyvr.getGroupMask(mask))
{
    uint32_t msk = mask;

    for (group = 0; group <= EasyVR::PASSWORD; ++group, msk >>= 1)
    {
        if (!(msk & 1)) continue;

        if (group == EasyVR::TRIGGER)

            pcSerial.print(F("Trigger: "));

        else if (group == EasyVR::PASSWORD)

            pcSerial.print(F("Password: "));

        else

        {

            pcSerial.print(F("Group "));

            pcSerial.print(group);

            pcSerial.print(F(" has "));

        }
    }
}

```

```

count = easyvr.getCommandCount(group);

pcSerial.print(count);

if (group == 0)

    pcSerial.println(F(" trigger(s)"));

else

    pcSerial.println(F(" command(s)"));

for (int8_t idx = 0; idx < count; ++idx)
{
    if (easyvr.dumpCommand(group, idx, name, train))
    {
        pcSerial.print(idx);

        pcSerial.print(F(" = "));

        pcSerial.print(name);

        pcSerial.print(F(", Trained "));

        pcSerial.print(train, DEC);

        if (!easyvr.isConflict())

            pcSerial.println(F(" times, OK"));

        else

        {
            int8_t confl = easyvr.getWord();

            if (confl >= 0)

                pcSerial.print(F(" times, Similar to Word "));

            else

            {
                confl = easyvr.getCommand();

                pcSerial.print(F(" times, Similar to Command "));

            }
        }
    }
}

```

```
        pcSerial.println(confl);
    }
}
}
}
}
}
group = 0;
set = 0;
mask |= 1; // force to use trigger (mixed SI/SD)
useCommands = (mask != 0);
isSleeping = false;
pcSerial.println(F("---"));
}
const char* ws0[] =
{
    "ROBOT",
};
const char* ws1[] =
{
    "ACTION",
    "MOVE",
    "TURN",
    "RUN",
    "LOOK",
    "ATTACK",
    "STOP",
    "HELLO",
```

```
};

const char* ws2[] =
{
    "LEFT",
    "RIGHT",
    "UP",
    "DOWN",
    "FORWARD",
    "BACKWARD",
};

const char* ws3[] =
{
    "ZERO",
    "ONE",
    "TWO",
    "THREE",
    "FOUR",
    "FIVE",
    "SIX",
    "SEVEN",
    "EIGHT",
    "NINE",
    "TEN",
};

const char** ws[] = { ws0, ws1, ws2, ws3 };

int readNum()
{
```

```
int rx;

int num = 0;

delay(5);

while ((rx = pcSerial.read()) >= 0)

{

    delay(5);

    if (isdigit(rx))

        num = num * 10 + (rx - '0');

    else

        break;

}

return num;

}
```