



Département de la physique appliquée

Master en sciences et techniques

Génie Biomédical : Instrumentation et maintenance

Réalisation d'un ECG avec NodeMCU

Encadré par

➤ M .HAJJAMI

Par :
Meryem OTMAN
Sara NASSIM

Avant tout, un grand merci à toute la famille qui me soutient de façon inconditionnelle non seulement le long de nos parcours scolaires, mais aussi dans nos quotidiens.

*Nous tenons à présenter très sincères remerciements à nos encadrant à Fst Monsieur **HAJJAMI** pour ses conseils, ses remarques, sa disponibilité et son soutien tout au long de la période d'étude ; ainsi pour sa disponibilité à nous faire partager ses connaissances, son expérience et son savoir- faire, et n'ayant ménagé aucun effort pour initier à la vie professionnelle, par ses conseils précieux et ses recommandations qu'il nous prodigué durant toute sa période*

Finalement, nous remercions tout le personnel de Fst, ainsi que toute personne qui, de près ou de loin, a contribué à l'accomplissement de ce modeste travail.

Introduction

Le domaine médical a bénéficié ces dernières décennies du progrès réalisé dans les technologies de l'électronique. Actuellement, Le matériel, les systèmes et les appareils médicaux utilisent des composants électroniques de plus en plus spécifiques rendant ces instruments plus efficaces et répondant aux normes des utilisateurs. L'affichage simultané/en temps réel des grandeurs physiologiques vitales a permis de connaître et de surveiller l'état du patient. Dans ce contexte, on peut citer l'activité électrique du cœur (ECG). Dans ce travail, nous avons discuté et réalisé un prototype de mise en forme de l'ECG 3 dérivations. Le résultat final de notre conception a abouti à une chaîne d'étages électroniques nécessaires pour extraire cette grandeur physiologique et une carte d'acquisition autour d'un microcontrôleur du Node MCU pour gérer l'affichage de cette grandeur physiologique sur une page web.

Pour faciliter l'étude et la réalisation on a décomposé le problème en trois grandes parties :

- La première partie consiste à analyser le problème ;
- La deuxième partie est dédiée à la résolution du problème par la mise en place du modèle de la solution ;
- La troisième partie est la réalisation et l'affichage de signal ECG dans une page web avec la gestion de plusieurs patients ;

Sommaire

Table de figures.....	5
Chapitre1:analyse du problème	7
I. Introduction	7
1. Fonctionnement du Coeur humain	7
2. L'électrocardiogramme, sa structure et son rôle	8
3. Le Principe d'ECG	10
chapitre2:Etude et réalisation pratique d'une carte d'acquisition du signal ECG	12
I. Introduction	12
1. Conditionnement d'ECG	12
1-1 La detection d'ECG	13
1-2La mise en forme du signal.....	13
1-3 Le circuit d'alimentation.....	16
2. Simulation du circuit et affichage du signal ECG.....	17
2-1 Simulation du circuit de conditionnement à l'aide du Porteus.....	17
2-2 Affichage du signal ECG à l'aide de l'oscilloscope	20
Chapitre3:Le traitement et l'affichage local du signal ECG	21
I. Introduction	21
1. Traitement du signal ECG par le microcontrôleur arduino	21
1-1 Arduino	21
1-2 Logiciel/langage IDE ARDUINO	22
1-3Affichae du signal à l'aide du traceur série.....	24
Chapitre4:la transmission sans fils du signal ECG vers la page web	26
I. Introduction	26
1. les matériel /logiciels /langages utilisé.....	26
1-1 le NODE MCU	26
1-2XAMPP	28
1-3HTML et CSS	32
2. prototypage et simulation	34
2-1 le Principe de fonctionnement.....	34
2-2les étapes de programmation	35
Conclusion.....	50
Annexe	51
Bibliographie	54

Table de figure

Figure 1:La structure interne du cœur

Figure 2: L'électrocardiogramme en état normal

Figure 3 :les Etapes de conditionnement de l'ECG

Figure 4 :les électrodes

Figure 5:AD620

Figure 6: schémas de AD620 sous ISIS

Figure 7 :Filtre pass haut actif

Figure 8 :Filtre pass bas actif

Figure 9: Le circuit d'alimentation

Figure 10 : Le circuit de mise en forme

Figure 11 : Simulation du circuit de conditionnement à l'aide du Porteus

Figure 13 : la carte arduino

Figure 14 : le logiciel de arduino

Figure 15 : Figure montrant la programmation sous Arduino

Figure 16: Le node MCU

Figure 17: Installation du package de carte ESP8266

Figure 18 : XAMPP

Figure 19: APACHE

Figure 20: PHP

Figure 21: MYSQL

Figure 22 : premier étape de l'installation de XAMP

Figure 23 : deuxième étape de l'installation de XAMP

Figure 24 : troisième étape de l'installation de XAMP

Figure 25 : affichage XAMP

Figure 26 : CSS

Figure 27 : communication client/serveur avec Xamp

Figure 28: Le fonctionnement global du projet

Figure 29: Filephp :addUser.phpcomplet

Figure 30: Filephp :add_UserData.php

Figure 31: Le code Arduino

Figure 32 : Fichier Php1 : listC page Web

Figure 33:Fichier Php2 : listC page Web

Figure 34 : Codephp du graph : printer.php

Chapitre I : Analyse du problème

I. Introduction :

Dans les pays pauvres, les décès sont surtout dus à des causes infectieuses mais, dans les pays riches, ils sont liés aux maladies cardio-vasculaires, au diabète et au cancer. Les maladies cardio-vasculaires restent la cause principale de décès dans le monde, selon le rapport publié par l'Organisation mondiale de la santé (OMS). Plus précisément, ce sont les maladies coronariennes et les accidents vasculaires cérébraux qui, en 2011, ont le plus tué: 13,2 millions de personnes soit 24 % des 54,6 millions de décès enregistrés dans l'année. La comparaison établie par l'OMS, des taux de mortalité entre 2000 et 2011, indique que les six principales causes de décès restent inchangées alors que les causes les moins fréquentes, en bas de classement, ont subi quelques modifications. Les personnes souffrant de maladies cardiovasculaires ou exposées à un risque élevé de maladies cardiovasculaires (du fait de la présence d'un ou plusieurs facteurs de risque comme l'hypertension, le diabète ou une maladie déjà installée) nécessitent une détection précoce et une prise en charge comprenant soutien psychologique et médicaments, selon les besoins.

1. Fonctionnement du cœur humain :

La santé des artères et le bon fonctionnement du cœur, peuvent être altérés par de nombreux facteurs : mode de vie, âge, diabète, cholestérol, antécédents familiaux ... Pour se protéger des maladies cardio-vasculaires il faut comprendre le fonctionnement du cœur. A la base de la vie, il y a les cellules. Des milliards de cellules qui respirent, consomment et produisent des déchets. Ensemble, elles forment nos tissus et nos organes : cœur, poumons, reins, foie, cerveau, muscles, etc. Et toutes ces "unités de vie", ont besoin de l'oxygène apporté par le sang pour vivre. Les muscles qui travaillent en sont de gros consommateurs. Le cœur, un muscle composé de cellules spécifiques (les cardiomyocytes) consomme, à lui seul, 10 % de tout l'oxygène fourni à l'organisme et ne pèse qu'environ 300 grammes chez un adulte. Les cellules du cœur présentent 2 particularités :

- Elles sont capables de se contracter comme toutes les cellules musculaires,
- Elles peuvent aussi conduire de l'électricité, ce qui déclenche la contraction.

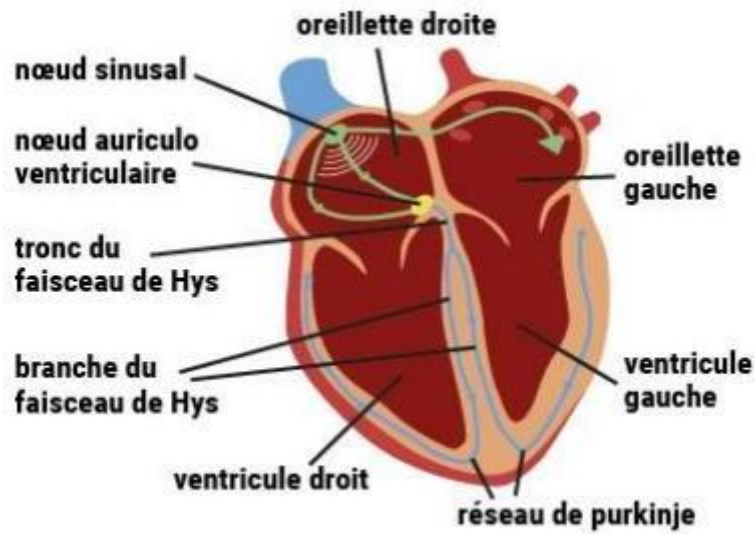


Figure 1: La structure interne du cœur

Cette source est constituée d'un amas de cellules capables de fabriquer un courant électrique de quelques millivolts. Partant du nœud sinusal, le courant se propage en tache d'huile dans le muscle cardiaque. Il circule dans les 2 oreillettes jusqu'à leur base, provoquant leur contraction. L'examen permettant de mesurer l'activité électrique du cœur est l'électrocardiogramme, ou ECG.

2. L'électrocardiogramme, sa structure et son rôle :

Un électrocardiogramme (ECG) correspond à l'enregistrement de l'activité électrique du cœur, nécessaire à ses contractions. Réalisé au cabinet du médecin généraliste ou cardiologue, il enregistre pendant quelques minutes les impulsions électriques du cœur par le biais d'électrodes placées au niveau des poignets, des chevilles et de la poitrine. Il permet de détecter certaines anomalies cardiaques (troubles du rythme ou de la conduction électrique, défauts d'oxygénation du cœur). C'est un point d'appel pour aller plus loin en cas de résultats anormaux. L'électrocardiogramme fournit immédiatement des renseignements sur la santé du cœur. Il est utilisé pour différentes raisons :

- En prévention (bilan de santé, avant une opération chirurgicale lourde) ;
- En cas de douleur thoracique ;
- dans le cas de certaines maladies cardiovasculaires (infarctus du myocarde, hypertension artérielle...).

- L'ECG permet aussi de diagnostiquer des accélérations anormales du rythme cardiaque (tachycardies) ou des ralentissements anormaux (bradycardies).

Ainsi Il devrait être systématiquement fait à tous les sportifs de haut niveau et à tous les plus de 50 ans, surtout s'ils sont sédentaires et se (re)mettent au sport, ou s'ils ont des facteurs de risque cardiovasculaires (tabagisme, hypertension artérielle...)

En termes généraux, regardons ce que représente un ECG et comment nous pouvons le sentir. L'ECG est séparé en deux intervalles de base, l'intervalle PR et l'intervalle QT, décrits cidessous.

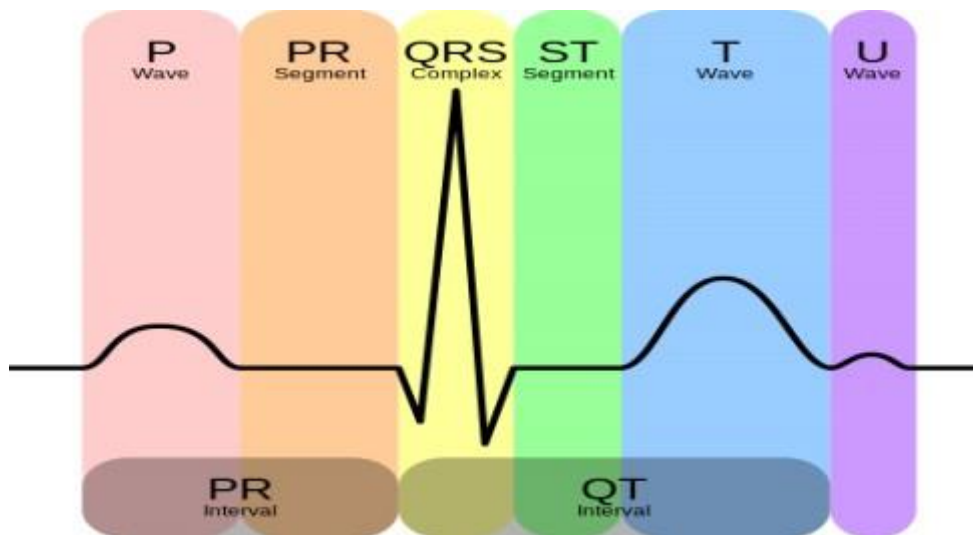


Figure 2: L'électrocardiogramme en état normal

Intervalle PR : L'intervalle PR est l'onde initiale générée par une impulsion électrique qui se déplace de l'oreillette droite vers la gauche. L'oreillette droite est la première chambre à voir une impulsion électrique. Cette impulsion électrique provoque la "dépolérisation" des chambres. Cela l'oblige à contracter et drainer le sang désoxygéné de la veine cave supérieure et inférieure dans le ventricule droit. Lorsque l'impulsion électrique traverse le sommet du cœur, elle provoque la contraction de l'oreillette gauche. L'oreillette gauche est responsable de la réception du sang nouvellement oxygéné des poumons dans le ventricule gauche via les veines pulmonaires gauche et droite.

Intervalle QT : L'intervalle QT est l'endroit où les choses deviennent vraiment intéressantes. Le QRS est un processus complexe qui génère la signature "bip" dans les

moniteurs cardiaques. Pendant QRS les deux ventricules commencent à pomper. Le ventricule droit commence à pomper le sang désoxygéné dans les poumons à travers les artères pulmonaires gauche et droite. Ils sont encore appelés artères parce que les artères transportent le sang du cœur. Ventricule gauche commence également à pomper le sang fraîchement oxygéné à travers l'aorte et dans le reste du corps. Après la contraction initiale vient le segment ST. Le segment ST est assez calme électriquement car c'est le moment où les ventricules attendent d'être "repolarisés". Enfin, l'onde T devient présente pour «re-plastifier» activement ou détendre les ventricules. Cette phase de relaxation remet à zéro les ventricules à remplir à nouveau par les oreillettes.

3. Le principe de l'ECG :

La télésurveillance des patients semble être une solution intéressante dans la mesure où elle permettrait à un seul infirmier de surveiller plusieurs malades à la fois. Le projet vise à :

- ❖ Suivre l'état du patient à distance
- ❖ Permettre au patient de visualiser l'évolution cardiaque (battement du cœur) graphiquement
- ❖ Avoir une base de données interprétable
- ❖ L'utilisation d'un matériel d'un prix acceptable.

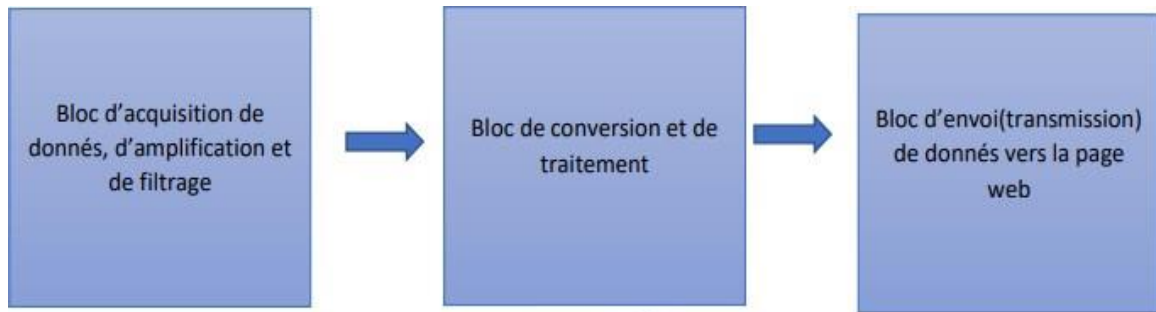
En effet le système est constitué de trois blocs : le bloc d'acquisition, le bloc de traitement et d'affichage et le bloc de transmission sans fil.

Le bloc d'acquisition : Il est constitué des capteurs électrocardiogramme (électrodes ECG) et d'un circuit de mesure du signal, Les capteurs électrocardiogramme permettent de transformer les activités du muscle cardiaque à des signaux électriques (tension électrique d'amplitude 10 mV crête à crête), Le circuit d'acquisition qui permet le conditionnement du signal issu des capteurs électrocardiogramme. C'est-à-dire il effectue l'amplification et le filtrage du signal issu des capteurs.

Le bloc d'affichage et traitement : Il permet la sauvegarde, le traitement et l'affichage en local du signal ECG.

Le bloc de transmission : il permet de transmettre les données du signal ECG vers un serveur (Programmer l'ESP8266 pour qu'il se comporte comme un serveur web). C'est-à-

dire un serveur où l'on peut se connecter et voir en temps réel l'évolution des signaux ECG du patient.



Chapitre II : Etude et réalisation pratique d'une carte d'acquisition du signal ECG

I. Introduction :

Toute transformation biologique doit, pour être exploitée, se présenter sous la forme de signaux compréhensibles, enregistrables et mesurable en valeurs normalisées. Ce chapitre sera consacré à l'étude de la réalisation d'un électrocardiogramme, à cet effet il sera représenté une description générale des différents étages électroniques de détection, mise en forme et affichage permettant la mesure du signal électrocardiogramme.

1. Conditionnement d'ECG :

Le signal ECG recueilli par les électrodes, nécessite un circuit électronique de mise en forme. Dans ce qui suit, on s'intéresse à la réalisation pratique d'un électrocardiographe et qui regroupe les différents étages illustrés dans la figure ci-dessous :

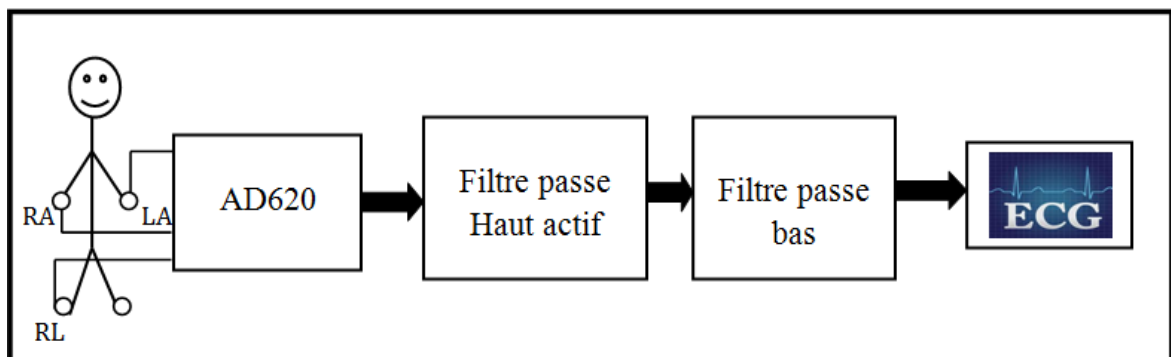


Figure 3 :les Etapes de conditionnement de l'ECG

1 - 1 . La détection d'ECG :

- **Capteur biomédical :**

Il s'agit d'un lot d'électrodes biomédicales, des électrodes jetables qui peuvent être utilisées pour mesurer les niveaux ECG et EMG. Ces petits tampons sont parfaits pour la surveillance à court terme (c'est pourquoi on a intérêt de les utiliser dans la réalisation de ce projet). Ils doivent être utilisés une fois et sont très pratiques grâce au gel intégré sans latex. Chaque coussinet adhère très bien à la peau et le connecteur à pression peut être poussé ou retiré du fil d'électrode sans problème.



Figure 4 :les électrodes

1-2 .La mise en forme du signal :

Le signal ECG est un signal à faible amplitude. Il est contaminé par des bruits de différentes natures. Ceci implique qu'un circuit de mise en forme est incontournable. Un étage d'amplification et d'autre de filtrage sont alors indispensables.

a. L'amplification du signal :

Comme la plupart des signaux biomédicaux, le signal ECG est détecté au moyen de trois électrodes placées sur le corps. Le signal ECG est donc obtenu en tant que différence de potentiel entre deux points, ce bio potentiel est de quelques mV d'amplitude donc, il est nécessaire d'utiliser un amplificateur différentiel pour amplifier ce signal.

En fait cette amplification ne doit concerner que le signal utile. Il faut donc faire une amplification « sélective » qui élimine ou atténue fortement tout signal ne contenant pas d'information pour ne garder que le signal utile. Afin d'accomplir ces performances, on

fait appel à l'amplificateur d'instrumentation qui adapte le signal utile à la chaîne d'acquisition de manière la plus précise.

Dans notre réalisation le choix est porté sur une version intégrée de l'amplificateur d'instrumentation ; il s'agit du circuit intégré le AD620 d'AnalogDevices. L'AD620 est à faible coût avec une haute précision qui nécessite une seule résistance externe pour régler des gains de 1 à 1000. L'AD620, avec sa grande précision et sa bonne linéarité, sa faible tension de décalage de 50 mV max et de dérive de température 0,6 mV / ° C max, est idéal pour une utilisation dans la détection de grandeurs de très faible niveau. Par ailleurs, le faible niveau de bruit, à faible courant de polarisation d'entrée, et de faible puissance, le rendent bien adapté pour des applications médicales telles que l'ECG.

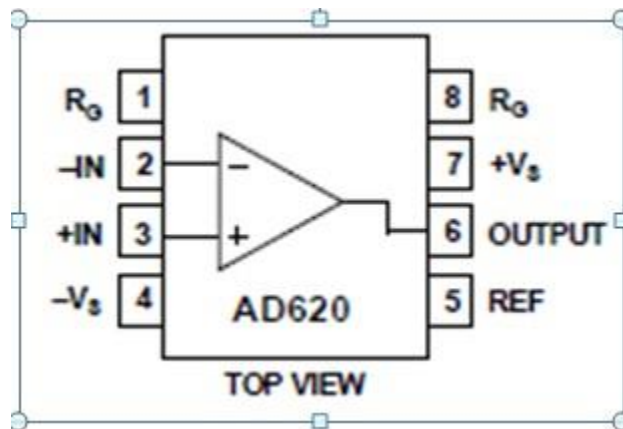


Figure 5 AD620

Amplificateurs d'instrumentation, comme l'AD620, offrent une haute TCMR, qui est une mesure de la variation de la tension de sortie lorsque les deux entrées sont changées en des quantités égales. Pour des performances optimales TCMR, la borne de référence doit être liée à une Point de faible impédance, et les différences dans la capacité et La résistance doit être maintenue à un minimum entre les deux entrées. Le gain assuré par le circuit intégré dans le circuit de 1 jusqu'à 1000.

En effet, le gain de l'AD620 est fixé par la résistance externe R1 selon l'équation suivante :

$$G = \frac{49.4K\Omega}{R1} + 1$$

Dans notre réalisation R1 est fixée à 560Ω afin d'obtenir un gain égale à 89.214

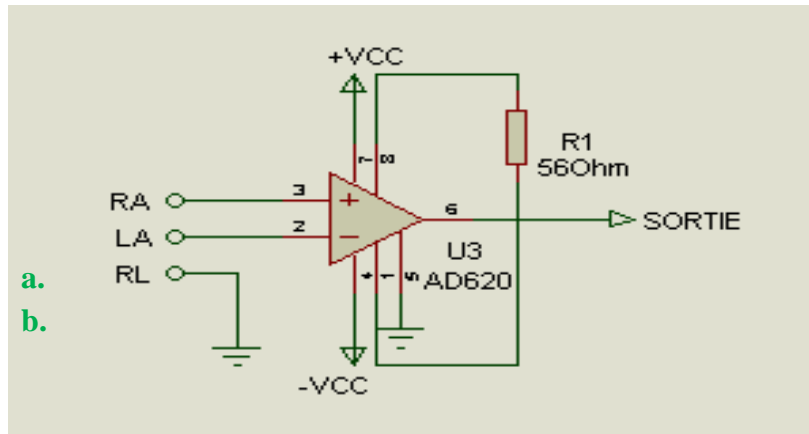


Figure 6: schémas d'AD620 associé à la résistance externe

b. Filtrage du signal :

Le signal ECG, ainsi amplifié, peut être noyé dans différentes sources de bruits. Parmi les plus importants, on peut citer :

- ❖ La dérive de la ligne de base (fréquence < 5Hz),
- ❖ Les artefacts dus aux mouvements (fréquence 1-10Hz),
- ❖ Le secteur et ses harmoniques (fondamental 50 ou 60Hz),

Dans notre réalisation, nous avons utilisé deux types de filtrage.

Filtre pass haut actif :

Le filtre passe haut actif est constitué d'un filtre passe haut passif en cascade avec un amplificateur et le circuit de rétroaction (UA709 voir annexe), ces deux derniers sont connectés dans une configuration non inverseur. Pour ce type de filtre, nous avons pris $C1=C2=C$. ce qui implique que la fonction de transfert de ce filtre est donnée par

$$H(j\omega) \rightarrow 0 = \frac{K \left(\frac{j\omega}{\omega_c} \right)^2}{1 + \frac{j\omega}{\omega_c} + \left(\frac{j\omega}{\omega_c} \right)^2}$$

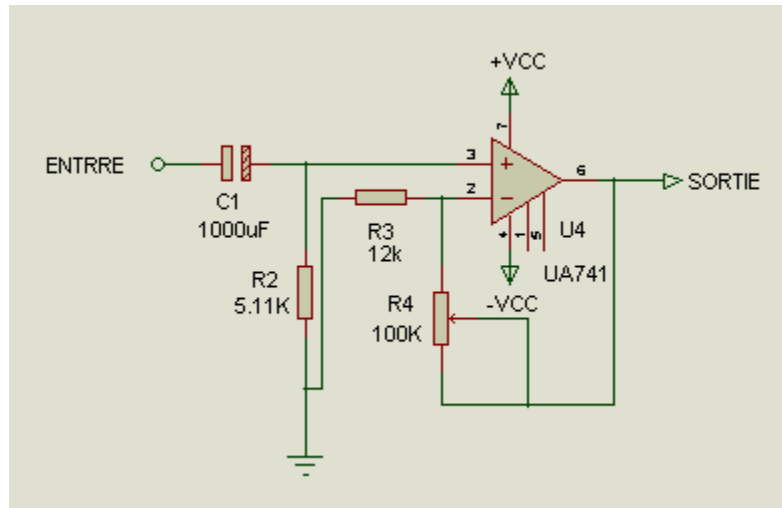


Figure 7 : Filtre pass haut actif

Le filtre passe haut est constitué par la capacité C1 et la résistance R2. Sa fréquence de coupure est égale à : $F_c = \frac{1}{2\pi C1 R2} = 0.03 \text{ HZ}$

L'amplificateur est de type non inverseur ou le gain est fixé par les deux résistances R3 et R4.

✚ Filtre pass bas passif :

Le filtre passe bas passif est constitué par la résistance R5 et la capacité C2. Les caractéristiques principales de ce filtre sont :

- La fréquence de coupure est calculée selon l'équation : $F_c = \frac{1}{2\pi C2 R5}$
- Avec une résistance $R_5 = 10 \text{ K}\Omega$ et une capacité $C_2 = 1 \mu\text{F}$, on obtient une fréquence de coupure : $F_c = 15.91 \text{ HZ}$

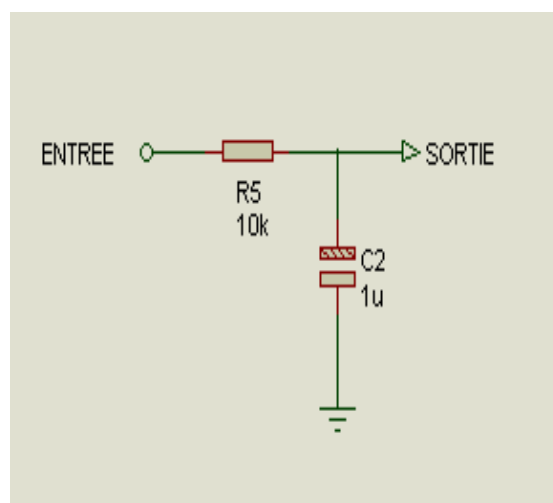


Figure 8 : Filtre pass bas passif

1-3 .Le circuit d'alimentation :

La plus part des étages électroniques indiqués au paravent contiennent des circuits intégrés, ces circuits intégrés nécessitent une alimentation de +5V et -5V. Notre système va être en contact direct avec le patient à travers les électrodes. Il est important donc, d'éviter les chocs électriques par l'utilisation d'une alimentation indépendante du secteur.

Dans ce cas, la meilleure solution c'est d'utiliser deux piles de 9V, cette tension va être branchée afin d'obtenir une tension de 9V. Ensuite en inversant cette tension à -9V comme indique la figure suivante :



Figure 9: Le circuit d'alimentation

La figure représente le circuit de mise en forme que nous avons réalisé et qui est dédié à l'amplification et au filtrage du signal ECG.

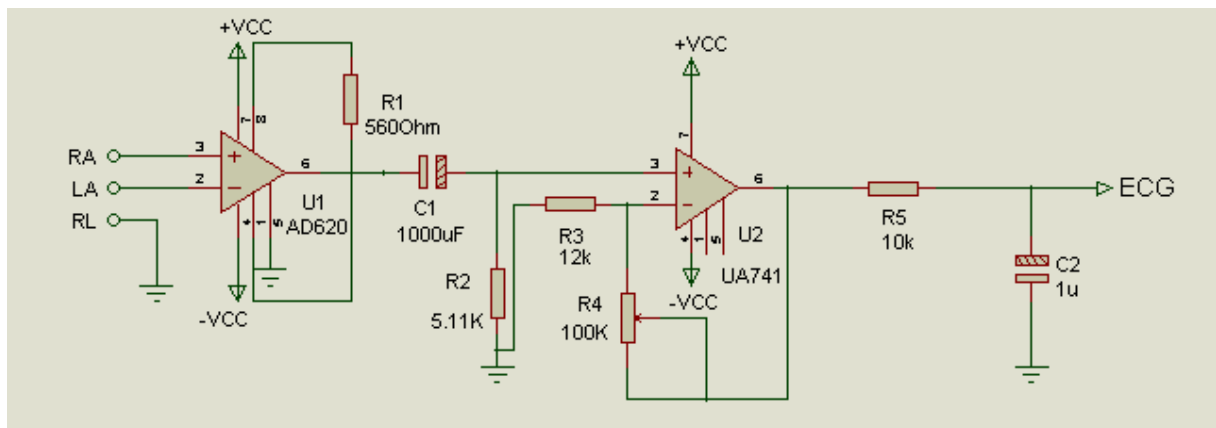


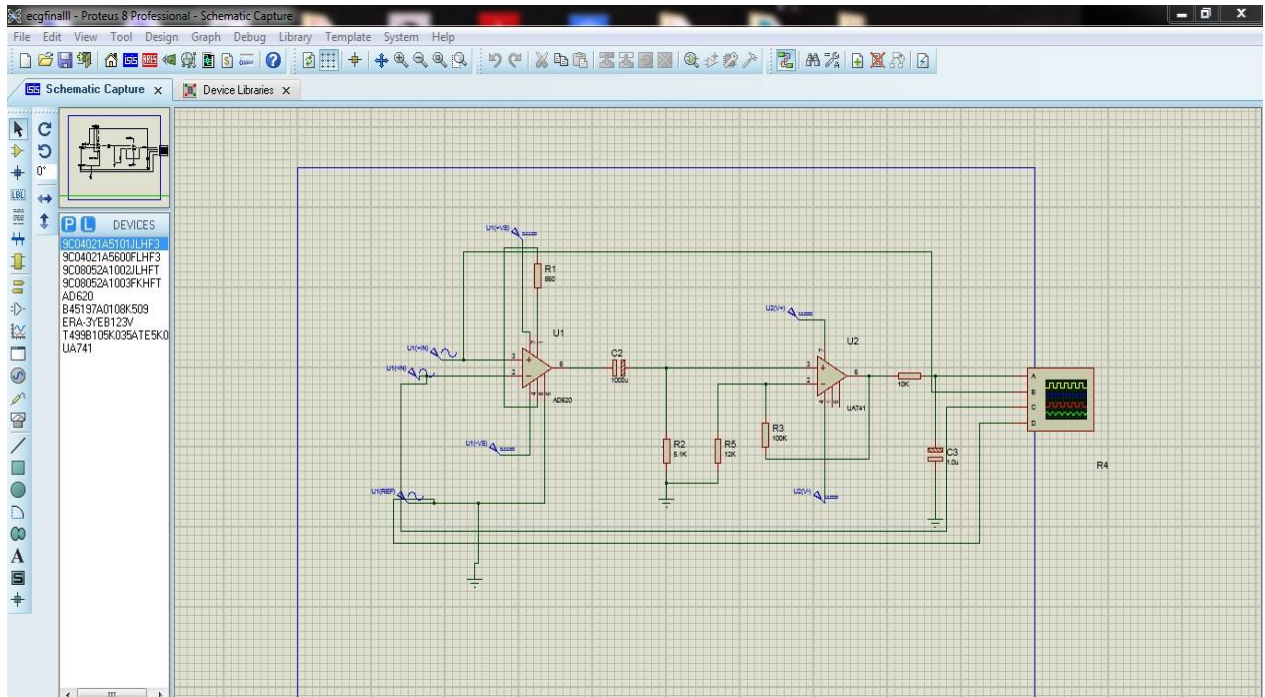
Figure 10: Le circuit de mise en forme

2. Simulation du circuit et affichage du signal ECG:

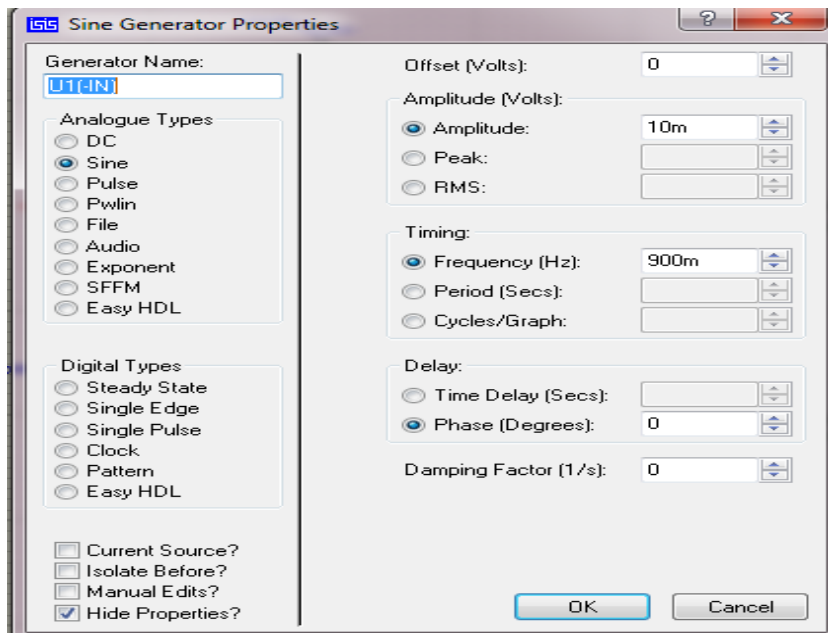
2.1 . Simulation du circuit de conditionnement à l'aide du Proteus :

Proteus est une suite logicielle destinée à l'électronique. Développé par la société LabcenterElectronics, les logiciels inclus dans Proteus permettent la

CAO(conception assistée par ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES, PROSPICE et VSM.



Pour la simulation on a utilisé la configuration suivante des différents signaux d'entrées



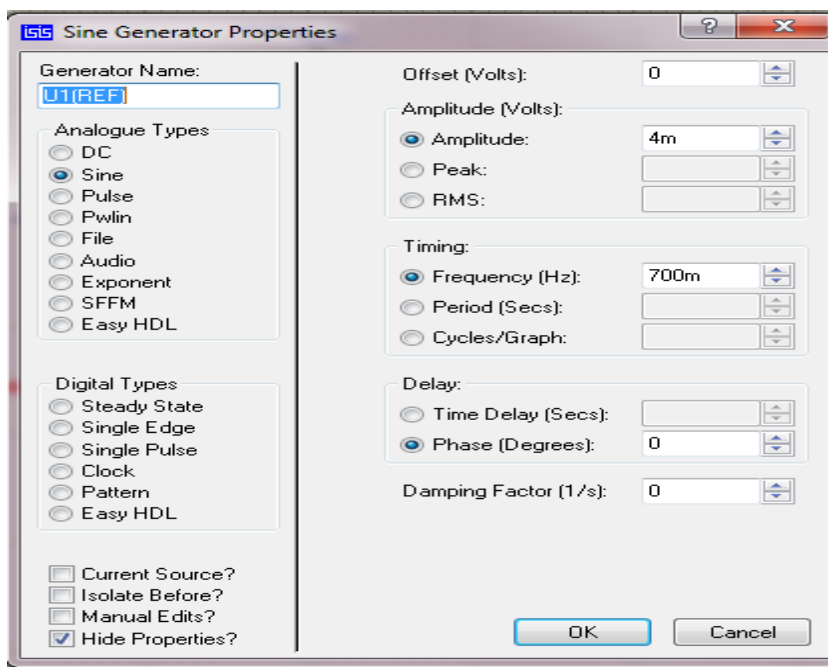
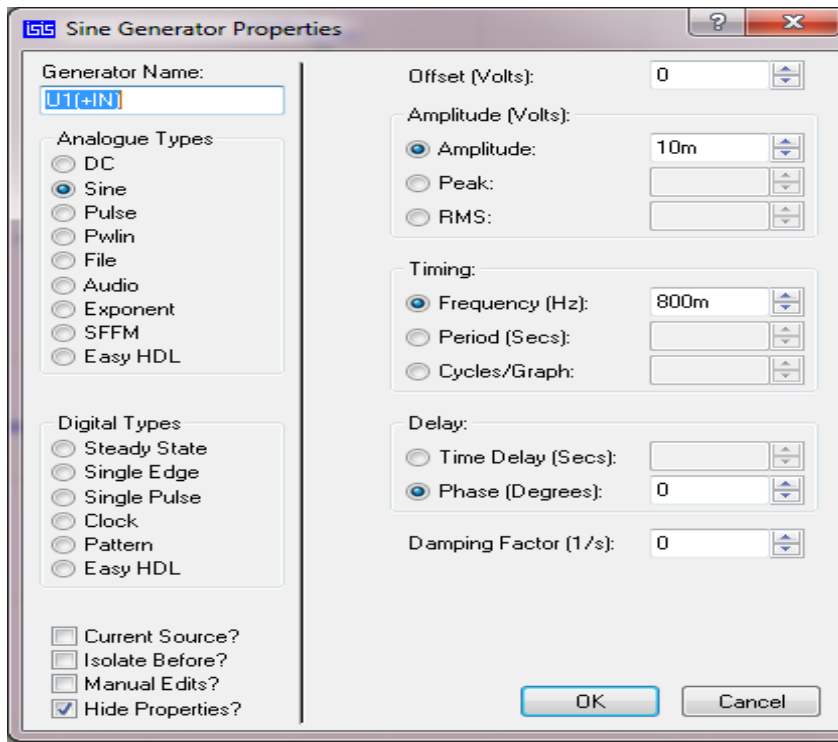


Figure 11 : la configuration des différents signaux d'entrées

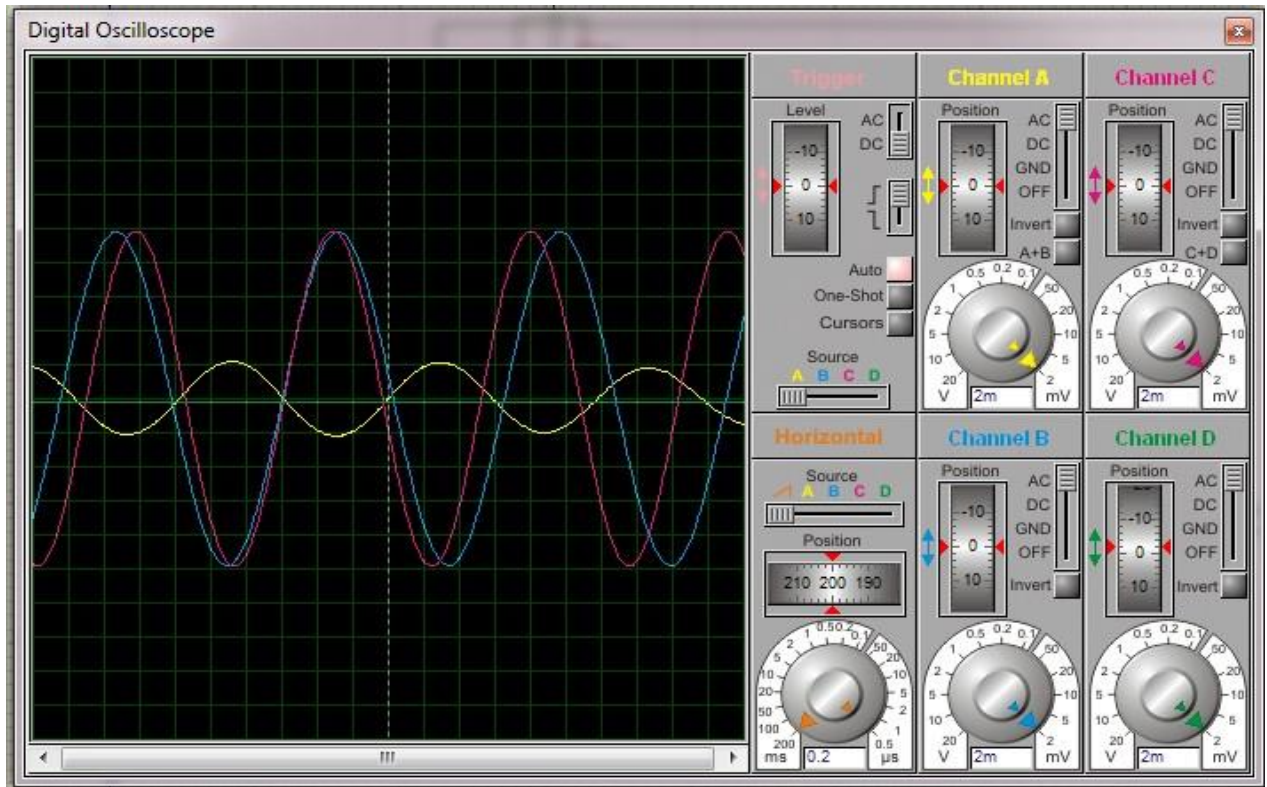


Figure 12 : Simulation du circuit de conditionnement à l'aide du Proteus

Channel A : La sortie de l'ECG

Channel B : Correspond au signal de la main droite

Channel C : Correspond au signal de la main gauche

Channel D : Correspond au signal du pied droit

2.2 . Affichage du signal ECG à l'aide de l'oscilloscope :

- La carte du circuit électronique : voir le DVD
- Affichage du signal par l'oscilloscope : Voir le DVD

Chapitre 3 : le traitement et l'affichage local du signal ECG

I. Introduction :

1. Traitement du signal ECG par le microcontrôleur Arduino :

1.1 . ARDUINO :

Est une carte microcontrôleur basée sur l'ATmega328P .Il possède 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un quartz à 16 MHz, une connexion USB, une prise d'alimentation et un bouton de réinitialisation. Il contient tout le nécessaire pour supporter le microcontrôleur ; connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur AC-DC ou une batterie. "Uno" signifie un en italien et a été choisi pour marquer la sortie de Arduino Software (IDE) 1.0. Le conseil d'Uno et la version 1.0 d'Arduino Software (IDE) étaient les versions de référence d'Arduino, maintenant évolué vers des versions plus récentes. La carte Uno est la première d'une série de cartes Arduino USB et le modèle de référence de la plate-forme Arduino

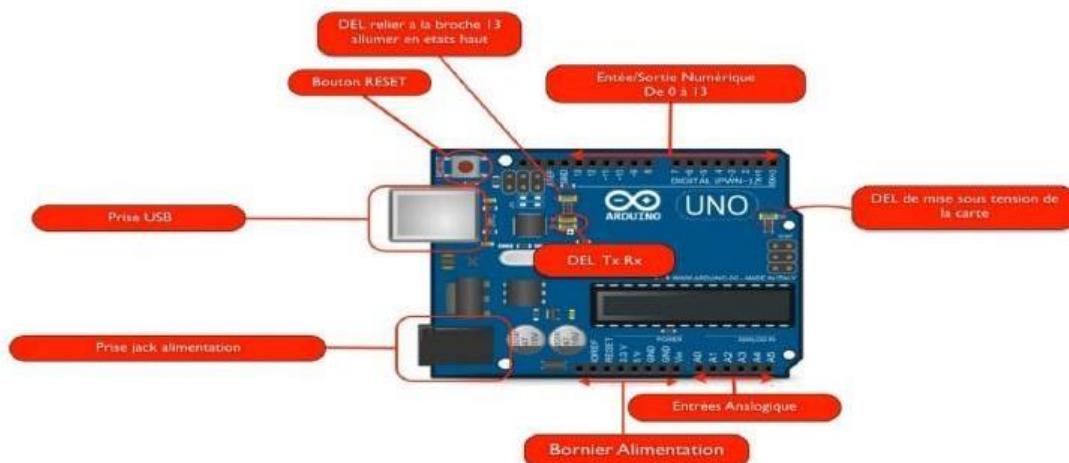


Figure 13 : La carte Arduino

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
Digital I/O Pins	54 (dont 15 fournissent sortie PWM)
Broches d'entrée analogiques	16
DC Courant par I/O Pin	40 mA
Courant DC pour 3,3 Pin	50 mA
Mémoire Flash	256 Ko (ATmega2560) dont 8 Kb utilisé par Boot Loader
SRAM	8 Kb(ATmega2560)
EEPROM	4 Kb (ATmega2560)
Fréquence d'horloge	16 MHz

Tableau 1: information sur la carte Arduino

1.2. Logiciel/langage IDE ARDUINO :



Figure 14: le logiciel dearduino

Unlogiciel , ou IDE (IntegratedDevelopmentEnvironment) qui fonctionne sur notre ordinateur, utilisé pour écrire et télécharger du code informatique sur la carte physique. La plate-forme Arduino est devenue très populaire auprès des personnes qui débutent avec l'électronique, et pour une bonne raison. Contrairement à la plupart des cartes de circuits programmables précédentes, l'Arduino n'a pas besoin d'un matériel séparé (appelé un programmeur) pour charger le nouveau code sur la carte on peut simplement utiliser un câble USB. En outre, l'IDE Arduino utilise une version simplifiée de C ++, ce qui facilite l'apprentissage de la programmation.

L'environnement de développement intégré Arduino - ou Arduino Software (IDE) - contient un éditeur de texte pour écrire du code, une zone de message, une console de texte, une barre d'outils avec des boutons pour les fonctions communes et une série de menus. Il se connecte au matériel Arduinopour télécharger des programmes et communiquer avec eux.



Figure 15 : Figure montrant la programmation par Arduino

Les programmes écrits en utilisant Arduino Software (IDE) sont appelés croquis. Ces esquisses sont écrites dans l'éditeur de texte et sont enregistrées avec l'extension de fichier .ino.

L'éditeur dispose de fonctionnalités pour couper / coller et pour rechercher / remplacer du texte. La zone de message donne des commentaires lors de la sauvegarde et de l'exportation et affiche également des erreurs. La console affiche la sortie de texte par le logiciel Arduino (IDE), y compris les messages d'erreur complets et d'autres informations. Le coin inférieur droit de la fenêtre affiche la carte et le port série configurée. Les boutons de la barre d'outils vous permettent de vérifier et de télécharger des programmes, de créer, d'ouvrir et de sauvegarder des croquis et d'ouvrir le moniteur série.

1.3. Affichage du signal à l'aide du traceur série :

a. Programme de l'affichage :


```
Fichier Édition Croquis Outils Aide
[Icons]
Graph $
http://www.arduino.cc/en/Tutorial/Graph
*/

void setup() {
  // initialize the serial communication:
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  // send the value of analog input 0:
  Serial.println(analogRead(A0));
  // wait a bit for the analog-to-digital converter to stabilize after the last
  // reading:
  delay(200);
}
```

b. Affichage du signal ECG dans le traceur série:

Chapitre 4 : La transmission sans fils du signal ECG vers la page web

I. Introduction :

Les principes de base d'un projet avancé consistent à pouvoir **stocker les données** que nous avons lues à partir des capteurs connectés. De cette façon, nous pouvons surveiller les données en temps réel, mais on peut également obtenir des informations historiques. Il nous permet également de capturer des données à partir de plusieurs périphériques d'entrée de données et de les afficher quand et comme nous le souhaitons. Même si cela pourrait également être fait avec une page Web dédiée en ajoutant un peu plus de code à l'Arduino, il est plus facile de le stocker dans une base de données et de créer une page Web (ou une interface utilisateur) qui lit les données de la base de données.

Dans cette partie, on va détailler comment on a transféré des données sur un réseau vers la base de données et les afficher via la page Web.

1. Les Matériels /logiciels /Langages utilisés :

1.1. Le NODE MCU :

Module basé sur un ESP8266 cadencé à 80 MHz et exécutant le firmware open source NodeMCU. Cette carte se programme via l'IDE Arduino et est compatible avec les scripts LUA. Ce microcontrôleur dispose d'une interface WiFi idéale pour les objets connectés. Des connecteurs latéraux mâles et femelles permettent d'enficher le module sur une plaque de montage rapide. L'interface sans fil Wifi permet la création de point d'accès sans fil, l'hébergement d'un serveur, la connexion à internet et le partage des données par exemple. Le module se programme directement à partir de l'IDE Arduino (installation d'une extension nécessaire) et nécessite un cordon microUSB (non inclus). Son implantation le rend compatible avec les plaques de connexions rapides.

Remarque: l'utilisation de cette carte est réservée à un public averti.

Caractéristiques:

- Alimentation:
 - 5 Vcc via micro
 - USB - 3,3 Vcc viabroches
- Vin
- Microcontrôleur: ESP8266
- Microprocesseur: Tensilica LX106
- Fréquence: 80 MHz
- Mémoire RAM: 64 kB
- Mémoire Flash: 96 kB
- 10 E/S digitales compatibles PWM
- Interfaces: I2C, SPI, UART
- Interface Wifi 802.11 b/g/n 2,4 GHz
- Antenne intégrée
- Température de service: -40 à 125 °C
- Dimensions: 58 x 31 x 12 mm



Référence fabricant: NodeMCU ESP8266

❖ **Programmation de l'ESP8266 avec ide arduino :**

Le NodeMCU est préprogrammé avec interpréteur Lua pour rendre leur programmation simple en va le configurer pour qu'il fonctionne avec l'IDE d'Arduino.

NodeMCU avec l'IDE Arduino, il va écrire directement dans le firmware. la principale différence est la distribution des broches dans la carte nodemcu.

On a Utilisé le câble USB pour connecter NodeMCU à l'ordinateur, à la mise sous tension on doit avoir un clignotement de la LED bleu.

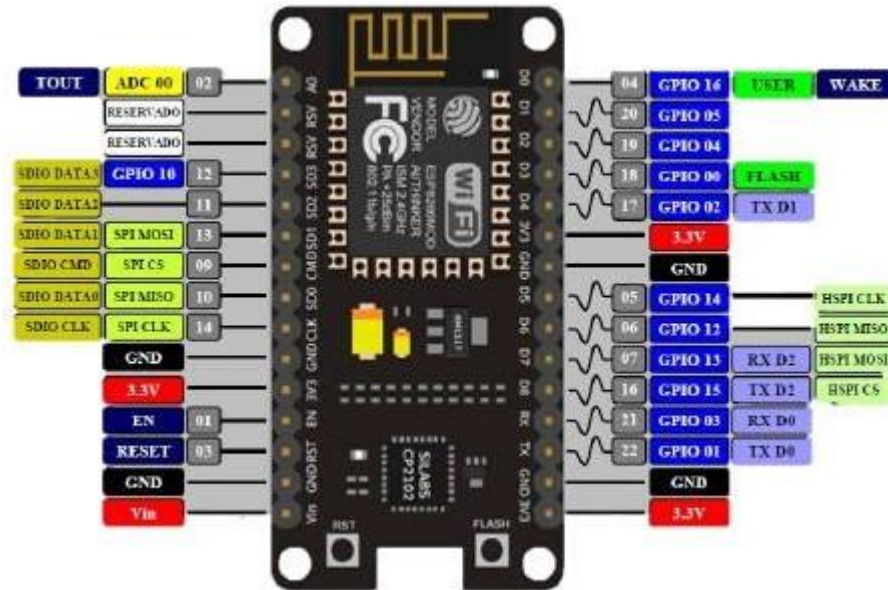


Figure 16:Lenode MCU

- Dans notre projet on aura besoin de l'entrée analogique A₀ pour connecter le le signal ECG avec le NODEMCU et le GND du Node MCU est reliée à la masse du circuit .

Installation du package de carte ESP8266

http://arduino.esp8266.com/stable/package_esp8266com_index.json

dansArduino IDE-> File->Perferences-> Settings->le champ URL de gestionnaire de forum supplémentaire.

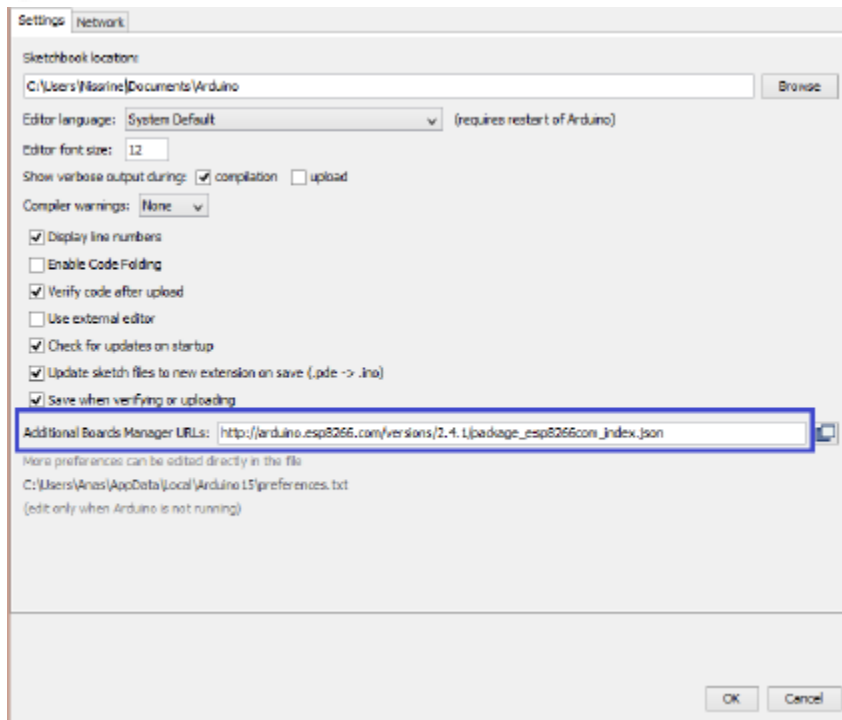


Figure 17: Installation ESP8266

1.2. XAMPP :

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

XAMPP permettant facilement de créer une interface web interagissant avec une base de données SQL.



Figure 18 : XAMPP

XAMPP se compose des trois choses principales qu'on doit savoir au démarrage du développement web. Elles sont :

- ✚ **Serveur Web Apache** : est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web Il permet de transférer un certain nombre de types de fichiers (HTML,XML,...) I Il répond à des requêtes défini dans un langage qui lui est propre (GET, PUT, CONNECT,...) : GET /index.html pour avoir la page index.html racine du site auquel on est connectée. Le navigateur interagit avec Apache.



Figure 19 : Apache

- ✚ **PHP** : Hypertext*Preprocessor*, plus connu sous son sigle PHP,est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.



Figure 20 : PHP

✚ **MySQL** : est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.

MySQL est un serveur de bases de données relationnelles SQL, il est écrit en C et C++ et il est multiplateforme.



Figure 21 : MYSQL

Pour installer xamp on Sélectionne le site de téléchargement. Dans votre navigateur internet : <https://www.apachefriends.org/index.html>.



Figure 22 : premier étape de l'installation de XAMP

Puis on Lance l'installation de XAMPP. Double clic sur l'icône sur bureau pour que l'installation commence.

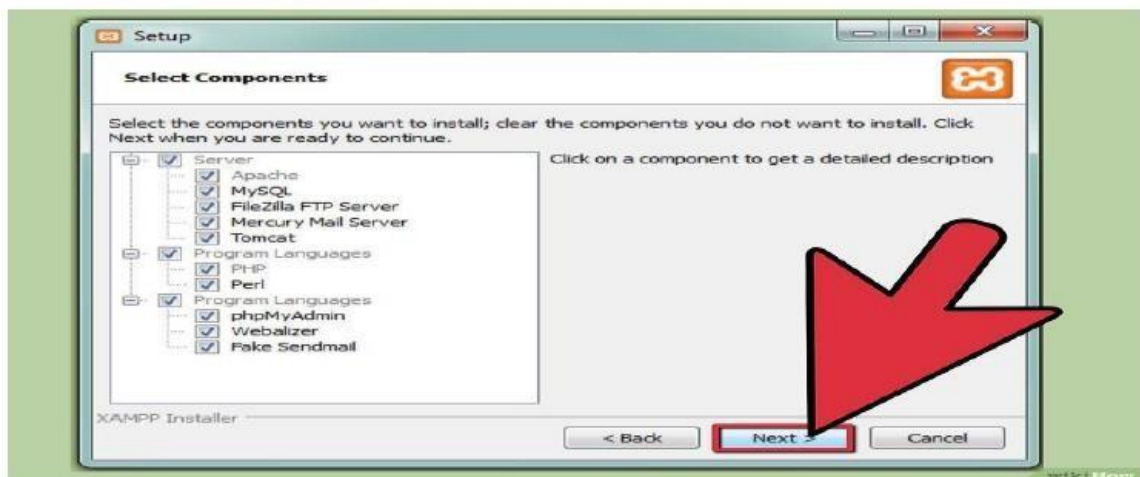


Figure 23 : Deuxième étape de l'installation de XAMP

On ouvre XAMPP Control Panel. Double cliquez sur l'icône XAMPP Control Panel pour ouvrir la fenêtre de contrôle de XAMP :

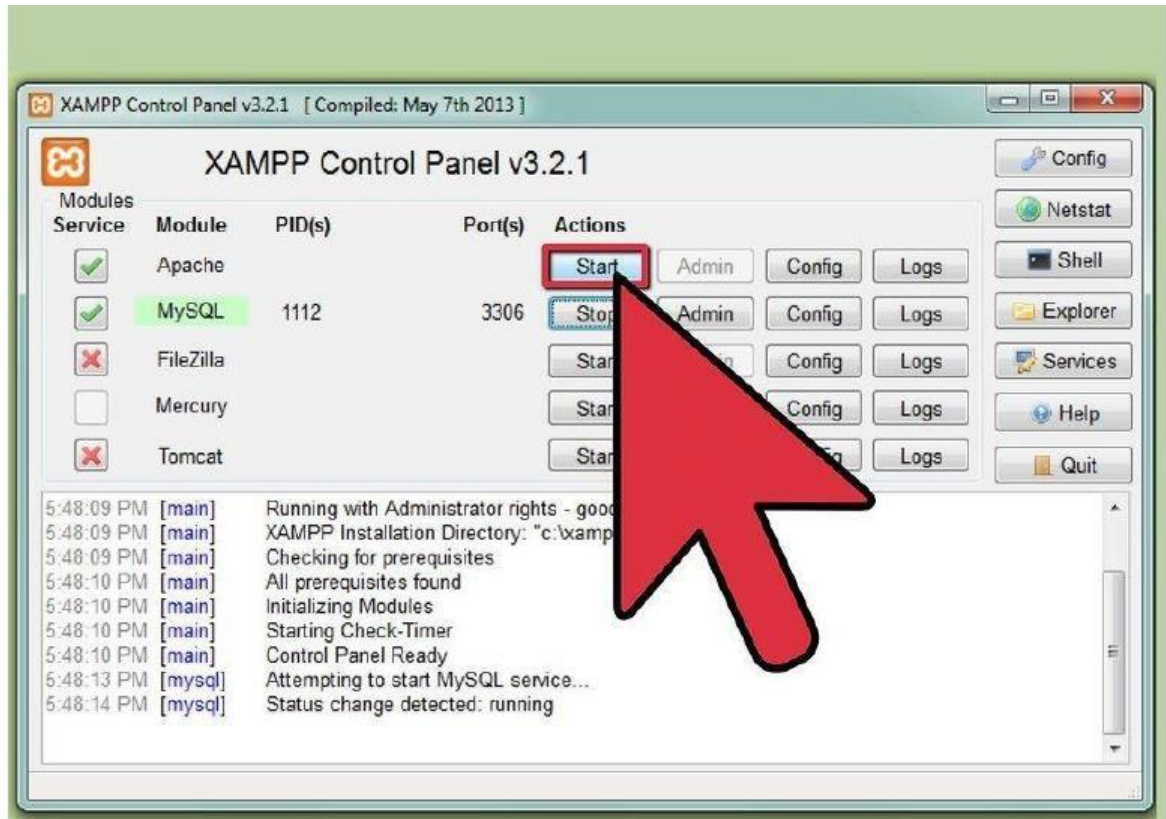


Figure 24 : troisième étape de l'installation de XAMP



Figure 25 : Affichage XAMP

1.3. HTML et CSS :

L'**HyperText MarkupLanguage**, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de

l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.

Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS).

Les éléments de base constituent la colonne vertébrale de tout document HTML. Ils se trouvent sur toutes les pages web après la déclaration du doctype qui définit la version de (X)HTML utilisée par la page. Les éléments décrivant le contenu d'une page web se trouvent entre la balise ouvrante <html> et la balise fermante </html>. Ces éléments peuvent être des headers, des paragraphes, des liens, ou des images ...

- **CSS :** Le terme CSS est l'acronyme anglais de *Cascading Style Sheets* qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelées les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. Bien que l'HTML puisse être mis en forme à l'aide de balises prévus à cet effet, de nos jours il est plus judicieux d'utiliser le CSS. L'avantage de l'utilisation d'un fichier CSS pour la mise en forme d'un site réside dans la possibilité de modifier tous les titres du site en une seule fois en modifiant une seule partie du fichier CSS. Sans ce fichier CSS, il serait nécessaire de modifier chaque titre de chaque page du site (difficilement envisageable pour les énormes sites de plusieurs milliers de pages).



Figure 26 : CSS

En résumé :

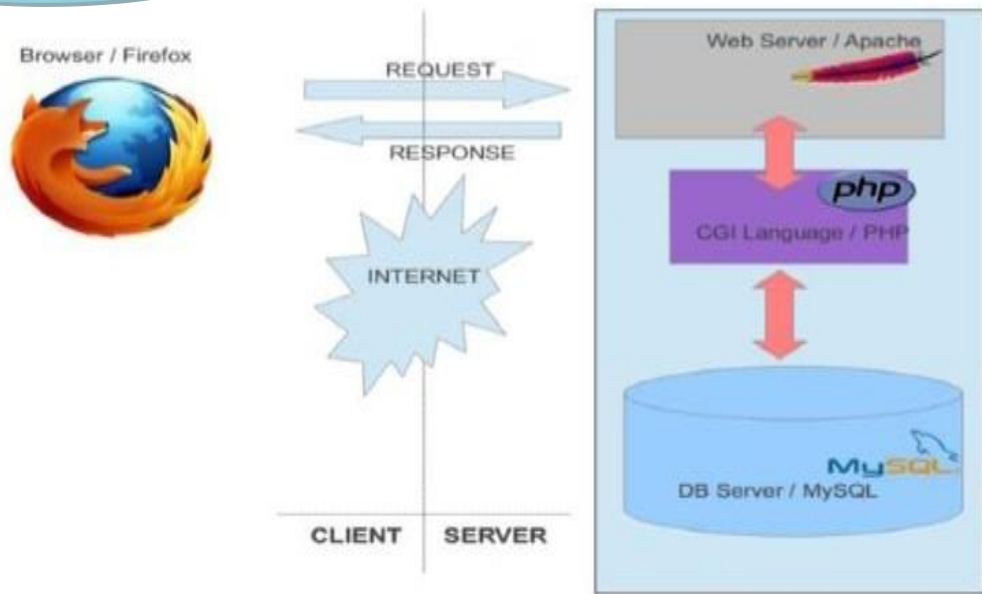


Figure 27 : communication client/serveur avec Xamp

2. Prototypage et simulation :

2.1. Le Principe de fonctionnement :

Dans la figure ci-dessous on modélise le fonctionnement global du projet :

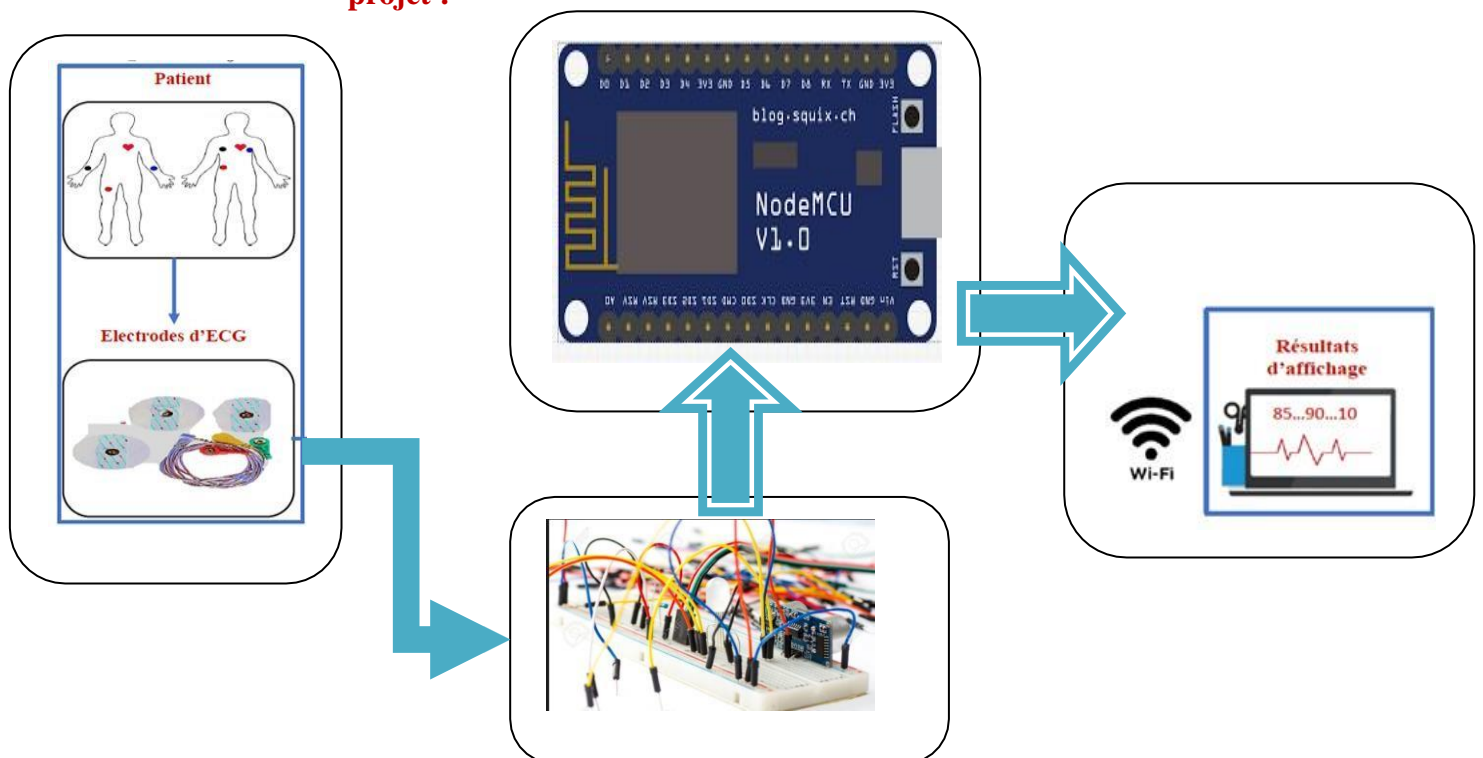


Figure 28: Le fonctionnement global du projet

2.2. Les étapes de programmation

Étape 1 :

- Connectez le **NODE MCU** à notre réseau local :

ESP8266WiFi.h est nécessaire pour faire toutes les fonctionnalités liées au WiFi telles que la connexion, AP, etc.

ESP8266HTTPClient.h gère tous les protocoles HTTP

On définit SSID et le mot de passe du routeur WiFi, là où l'ESP se connecte

```
const char* ssid = "";
```

```
const char* pwd = "";
```

Pour faire un serveur Web on va se connecter au point d'accès WiFi par la fonction :

```
WiFi.begin(ssid,pwd);
```

Pour obtenir l'adresse IP, c.-à-d. Attribuée à ESP8266 par le routeur WiFi, on utilise cette commande :

```
Serial.print("IP address: ");  
Serial.println(WiFi.localIP()); //IP address assigned to your ESP
```

```
void setup() {
  Serial.begin(115200);
  //WiFi.mode(WIFI_OFF);          //Prevents reconnection issue (taking too long to connect)
  delay(1000);
  WiFi.mode(WIFI_STA);          //This line hides the viewing of ESP as wifi hotspot

  WiFi.begin(ssid, password);    //Connect to your WiFi router
  Serial.println("");

  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  //If connection successful show IP address in serial monitor
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP()); //IP address assigned to your ESP
}
```

■ Identification du chaque patient

L'identification du chaque patient se fait par la validation de son nom de son référence et son statut pour cela on a utilisé la fonction get input qui permet de retourner un string dès que le utilisateur clique surentree. **Le code utilisé est ci-dessous :**

```
String get_input(String what_to_get){
  /* this function return String from the stdin of the serial and ask dynamicly for the desired input */
  Serial.println("Enter "+what_to_get+" : ");
  while(true){
    String temp ;
    while(Serial.available()) {
      temp = Serial.readString(); // read the incoming data as string
      if (temp.length() > 0){
        Serial.print("You Entred "+what_to_get+" : ");
        //Trimming last character \n
        temp.trim() ;
        Serial.println(temp);
        return temp;
      }
    }
  }
}
```

Et puis on fait l'appel de cette fonction pour entrer le nom la référence et le statut du notre patient.

Étape 2 :

■ Préparer la base de données :

On a un service de base de données du serveur XAMPP qui est déjà installé, on se connecte simplement à notre interface phpMyAdmin pour créer une nouvelle base de données.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	ID	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modi
2	time	timestamp		on update CURRENT_TIMESTAMP	Non	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP	Modi
3	nom	text	latin1_swedish_ci		Non	Aucun(e)			Modi
4	reference	text	latin1_swedish_ci		Non	Aucun(e)			Modi
5	status	text	latin1_swedish_ci		Non	Aucun(e)			Modi
6	graph_name	text	latin1_swedish_ci		Non	Aucun(e)			Modi

On a créé la Data base Test dans la quelle on a créé le tableau patients ou on va stocker les données (Nom ; référence ; statu)

Cette Db contient 6 colonnes :

- **ID :**
- **Time :**
- **Nom complet du patient :**
- **Référence :**
- **Status :**
- **graph_name :** qui permet de référencer une autre table qui est appelé graph_name se régénère automatiquement pour tout nouveau patient et qui contient les valeurs des capteurs qui vont être envoyées par la NODEMCU vers la DB.

Étape 3 : Recevoir et enregistrer les données :

- Créez des fichiers qui captureront les données envoyées par NODE MCU et écrivez-les dans la base de données :

Le moyen le plus simple d'obtenir des données de la NodeMcu dans notre base de données consiste à utiliser les méthodes de requête PHP et HTTP GET.

```

    }
}

void send_data(String PHPpage ,String Data){
    HTTPClient http;    //Declare object of class HTTPClient
    String link = "http://"+host+"/projet.php/"+PHPpage+"?"+Data ;//arduino write or pass argument in the link
    http.begin(link);
    int httpCode = http.GET();    //Send the request
    // String payload = http.getString();    //Get the response payload
    Serial.print("link : ");Serial.println(link);    //Print HTTP return code
    // Serial.print("Result : ");Serial.println(payload);    //Print HTTP return code
    Serial.print("httpCode : "); Serial.println(httpCode);    //Print request response payload
    http.end();    //Close connection
}

```

On fait un appel à cette fonction dès qu'on aura les données qui identifient chaque patient.

```

//AddUser.php?c=patient1&n=Mohamedalloui&ref=11&a=alive
String Data = "c="+Users_table+"&n="+Name+"&ref="+Ref+"&a="+Stat;
send_data("AddUser.php",Data);// file already created name AddUser

}

```

Néanmoins, cela n'écrit pas directement les données dans la base de données, nous devons donc créer un fichier PHP auquel nous enverrons des données et qui écrira ces données dans la base de données.

Dans le code PHP on a utilisé la fonction **recordExists** qui permet de vérifier si les données du patient déjà existent dans la base de données ou non. Il return false si le patient déjà existe sinon elle return true qui permet par la suite de continuer les commande qui vient

par la suite.

```
1 <?php
2 function recordExists($table, $name, $ref, $mysqli) {
3     $query = "SELECT * FROM `{$table}` WHERE nom = '". $name ."' and reference = '". $ref ."'";
4     $result = $mysqli->query($query);
5
6     if($result->num_rows > 0) {
7         return false; // The record(s) do exist
8     }
9     return true; // No record found
10 }
11
```

On se connecte à la base de données par la fonction suivante :

```
// on se connecte a la base de donnees |
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";
$table = "patients";
// si y a un problem au niveau de la connexion y affiche error
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Insertion des données dans la base de données

```
if(recordExists($table,$_GET['n'],$_GET['ref'],$conn)){
//on insere les valeurs dans le tableau on insere les valeur de la collone VALEUR les autre se remplit automatiquement
$sql = "INSERT INTO " . $_GET['c'] . " (nom,reference,status,graph_name) VALUES ('" . $_GET['n'] . "','" . $_GET['ref'] . "','" . $_GET['a'] . "','" . $_GET['graph'] . "','" . $_GET['n'] . "','" . $_GET['ref'] . "')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Génération de une autre table qui contient deux colonnes time et values


```
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$create_user_table = "CREATE TABLE graph_" . $_GET['n'] . "_" . $_GET['ref'] . " (
time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
value INT NOT NULL
)";
```

➤ Fichier PHP : addUser.php complet :

```
<?php
function recordExists($table, $name, $ref, $mysqli) {
    $query = "SELECT * FROM `{$table}` WHERE nom = '". $name."' and reference = '". $ref.'";
    $result = $mysqli->query($query);

    if($result->num_rows > 0) {
        return false; // The record(s) do exist
    }
    return true; // No record found
}

// on se connecte a la base de donnees
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";
$table = "patients";
// si y a un problem au niveau de la connexion y affiche error
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if(recordExists($table,$_GET['n'],$_GET['ref'],$conn)){
t
$sql = "INSERT INTO " . $_GET['c'] . " (nom,reference,status,graph_name) VALUES ('" . $_GET['n'] . "','" . $_GET['ref'] . "','" . $_GET['a'] . "','" . $_GET['graph'] . $_GET['n'] . "_" . $_GET['ref'] . "')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

```
$create_user_table = "CREATE TABLE graph_" . $_GET['n'] . "_" . $_GET['ref'] . " (
time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
value INT NOT NULL
)";

if ($conn->query($create_user_table) === TRUE) {
    echo "New tabel for user , created successfully";
} else {
    echo "Error: " . $create_user_table . "<br>" . $conn->error;
}

}
else
echo $_GET['n'] . " Record already exist !";
$conn->close();
?> |
```

Figure 29: Fichier PHP : addUser.php complet

Puis après qu'on ajoute les données de l'utilisateur nom, référence et son statu on doit remplir le tableau qui va être lui y associé.

```
//=====
//                               Main Program Loop
//=====
void loop() {

    int sensorValue = analogRead(A0);
    // Serial.print(sensorValue);
    Serial.println("Sending " + String(sensorValue) + " to : " + Name);
    // String v = String (random(0,1024));
    String Data = "graph=graph_" + Name + "_" + Ref + "&v=" + sensorValue;
    send_data("Add_UserData.php", Data);
    delay(500);

}
//=====*/
```

```
<?php
// on se connecte a la basede donnees
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// si y a un problem au niveau de la connexion y affiche error
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO ".$_GET["graph"]." (value) VALUES ('".$_GET['v']."')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

Figure 30: FichierPHP : add_UserData.php

Étape 4 :

▪ Le code ARDUINO :

Lorsque nous avons préparé la base de données et le fichier PHP qui écrit les données dans la base de données, nous pouvons maintenant téléverser le code ARDUINO qui envoie les données d'identification du patient et des capteurs.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>

/* Set these to your desired credentials. */
const char *ssid = "LA-PERLE"; //ENTER YOUR WIFI SETTINGS
const char *password = "20182019";

//Global Var zone
String Name ;
String Ref ;
String Stat ;
//Web/Server address to read/write from
String host = "192.168.1.36"; // database IP
String Users_table = "patients";

String get_input(String what_to_get){
    /* this function return String from the stdin of the serial and ask dynamicly for the desired input */
    Serial.println("Enter "+what_to_get+" : ");
    while(true){
        String temp ;
        while(Serial.available()) {
            temp = Serial.readString();// read the incoming data as string
            if (temp.length() > 0){
                Serial.print("You Entred "+what_to_get+" : ");
                //Trinming last character \n
                temp.trim() ;
                Serial.println(temp);
                return temp;
            }
        }
    }
}

void send_data(String PHPpage ,String Data){
    HTTPClient http; //Declare object of class HTTPClient
    String link = "http://"+host+"/projet.php/"+PHPpage+"?"+"Data";//arduino write or pass argument
    http.begin(link);
    int httpCode = http.GET(); //Send the request
    // String payload = http.getString(); //Get the response payload
    Serial.print("link : ");Serial.println(link); //Print HTTP return code
    // Serial.print("Result : ");Serial.println(payload); //Print HTTP return code
    Serial.print("httpCode : "); Serial.println(httpCode); //Print request response payload
    http.end(); //Close connection
}

void setup() {
    Serial.begin(115200);
    //WiFi.mode(WIFI_OFF); //Prevents reconnection issue (taking too long to connect)
    delay(1000);
    WiFi.mode(WIFI_STA); //This line hides the viewing of ESP as wifi hotspot

    WiFi.begin(ssid, password); //Connect to your WiFi router
    Serial.println("");
}
```

```

Serial.print("Connecting");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
//If connection successful show IP address in serial monitor
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); //IP address assigned to your ESP

//-----
Name = get_input("Name");
Ref = get_input("reference");
Stat = get_input("Status");
//-----

//AddUser.php?c=patient1&n=bob&ref=11&a=alive
String Data = "c="+Users_table+"&n="+Name+"&ref="+Ref+"&a="+Stat;
send_data("AddUser.php",Data);// file already created name AddUser

}

//=====
//                               Main Program Loop
//=====
void loop() {

    int sensorValue = analogRead(A0);
    // Serial.print(sensorValue);
    Serial.println("Sending "+String(sensorValue)+" to : "+Name);
    // String v = String (random(0,1024));
    String Data = "graph=graph_"+Name+"_"+Ref+"&v="+sensorValue;
    send_data("Add_UserData.php",Data);
    delay(500);

}

//=====*/

```

Figure 31: Le code Arduino

Nous devons pouvoir voir de nouvelles données dans notre base de données via PhpMyAdmin.

	ID	time	nom	reference	status	graph_name
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	2019-01-19 17:43:41	zah	re	sd	graph_zah_re
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	2019-01-19 17:47:29	zahra	HH11243	sickofthis	graph_zahra_HH11243

☐ Tout cocher Avec la sélection : ☐ Éditer ☐ Copier ☐ Supprimer ☐ Exporter

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes : Chercher dans cette table Trier sur l'index:

Opérations sur les résultats de la requête

☐ Imprimer ☐ Copier dans le presse-papiers ☐ Exporter ☐ Afficher le graphique ☐ Créer une vue

+ Options	
time	value
2019-01-19 17:44:14	241
2019-01-19 17:44:15	1024
2019-01-19 17:44:16	1024
2019-01-19 17:44:16	1024
2019-01-19 17:44:17	336
2019-01-19 17:44:17	439
2019-01-19 17:44:18	0
2019-01-19 17:44:19	585
2019-01-19 17:44:19	1024
2019-01-19 17:44:20	1024
2019-01-19 17:44:21	1024
2019-01-19 17:44:21	1024

Nous pouvons maintenant créer une application qui se connecte à la base de données et la lire, ou créer une page Web php qui les affiche.

Étape 5 :

■ Affichage des données :

Pour afficher les nouvelles données dans Notre base de données, il est plus facile de créer une page Web php qui lira les données et se rafraîchira toutes les quelques secondes.


```

<!DOCTYPE html>
<?php
$con=mysqli_connect("localhost","root","","test");
// Check connection
if (mysqli_connect_errno())
{
echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$result = mysqli_query($con,"SELECT * FROM patients");

?>
<html lang="en">
<head>
<meta charset="UTF-8" />
<title>List Users</title>
<!--Import Google Icon Font-->
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<!--Import materialize.css-->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.99.0/css/materialize.min.css">
</head>
<body>
<div class="container">

<div class="card-panel green darken">
<span class="white-text text-darken-2">
<h4 class="center-align">Users</h4>
</span>
</div>
<table class="striped highlight" id="posts_table">
<thead>
<tr>
<th>ID</th>
<th>Time</th>

```

FIGURE 32 :Fichier Php1 : listC page Web

```

<table class="striped highlight" id="posts_table">
<thead>
<tr>
<th>ID</th>
<th>Time</th>
<th>Nom</th>
<th>Reference</th>
<th>Status</th>
<th>Heart Graph</th>
</tr>
</thead>
<tbody>
<?php
while($row = mysqli_fetch_array($result))
{
echo "<tr>";
echo "<td>" . $row['ID'] . "</td>";
echo "<td>" . $row['time'] . "</td>";
echo "<td>" . $row['nom'] . "</td>";
echo "<td>" . $row['reference'] . "</td>";
echo "<td>" . $row['status'] . "</td>";
echo "<td>" . <a href='http://localhost/projet.php/printer.php?graph_name=".$row['graph_name']."'>See Graph</a>
</td>";
echo "</tr>";
}
?>
</tbody>
</table>

</div>
</body>
</html>

```

FIGURE 33:Fichier Php2 : listC page Web

Lorsqu'on ouvre la page new.php, elle nous affiche l'interface qui va être affichée à l'utilisateur :

Users

ID	Time	Nom	Reference	Status	Heart Graph
1	2019-01-19 17:43:41	zah	re	sd	See Graph
2	2019-01-19 17:47:29	zahra	HH11243	sickofthis	See Graph

On a ajouté une colonne qui nous permet de visualiser le graphe ECG du patient

Donc pour cela on a développé un autre code php qui nous permet de mettre les valeurs de l'ECG :

```

1  <?php
2
3  $dataPoints = array();
4  //Best practice is to create a separate file for handling connection to database
5  try{
6      // Creating a new connection.
7      // Replace your-hostname, your-db, your-username, your-password according to your database
8      $link = new PDO( 'mysql:host=localhost;dbname=test;charset=utf8mb4', //
9                      'mysql:host=localhost;dbname=canvasjs_db;charset=utf8mb4',
10                      'root',
11                      '',
12                      array(
13                          PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
14                          PDO::ATTR_PERSISTENT => false
15                      )
16                      );
17
18      $handle = $link->prepare("SELECT value FROM ".$_GET["graph_name"]." ");
19      $handle->execute();
20      $result = $handle->fetchAll(PDO::FETCH_OBJ);
21      $i = 0;
22      foreach($result as $row){
23          array_push($dataPoints, array("y"=> $row->value));
24      }
25      $link = null;
26  }
27  catch(PDOException $ex){
28      print($ex->getMessage());
29  }
30  ?>
    
```



```

31 <!DOCTYPE HTML>
32 <html>
33 <head>
34 <meta http-equiv="refresh" content="2" />
35
36 <script>
37 window.onload = function () {
38
39     var chart = new CanvasJS.Chart("chartContainer", {
40         animationEnabled: true,
41         exportEnabled: true,
42         theme: "light2", // "light1", "light2", "dark1", "dark2"
43         title:{
44             text: "Heartbeat Graph "
45         },
46         data: [{
47             type: "spline", //change type to bar, line, area, pie, etc
48             dataPoints: <?php echo json_encode($dataPoints, JSON_NUMERIC_CHECK); ?>
49         }]
50     });
51     chart.render();
52
53 }
54 </script>
55 </head>
56 <body>
57 <div id="chartContainer" style="height: 370px; width: 100%;"></div>
58 <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
59 </body>
60 </html>
    
```

Figure 34 : CodePHP du graph : printer.php

Graph ECG : voir DVD

Conclusion

Dans ce travail, on s'est intéressé à l'étude, de la théorie à la pratique, d'une technique indispensable dans le domaine du diagnostic médical, l'électrocardiographie. L'électrocardiographie est considérée comme un examen de routine clinique dans n'importe quel diagnostic du système cardiovasculaire. C'est ainsi, qu'une étude théorique du signal électrocardiogramme a été menée. En fait, dans l'étude théorique que nous avons présentée, différents aspects de l'électrophysiologie et la manière avec laquelle, le signal traduisant l'activité cardiaque est généré physiologiquement ou encore détecté à la surface du corps ont été présentés.

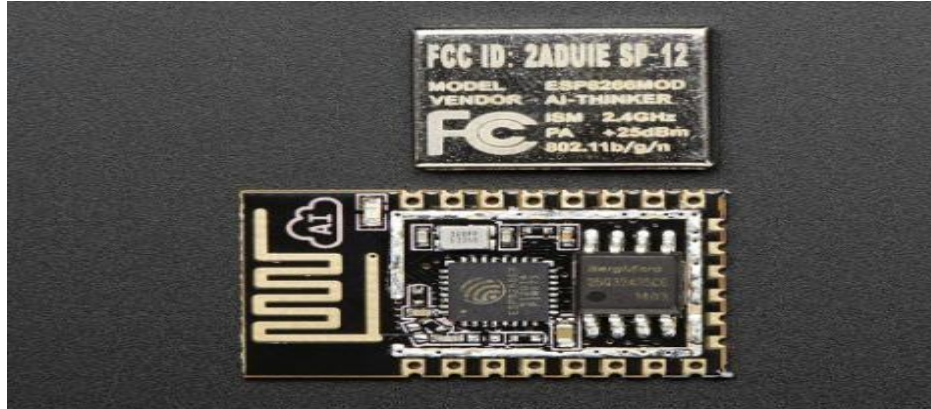
L'étape de la conception du dispositif : l'électrocardiographe, a été accompli, par une réalisation d'un électrocardiographe, a vu, une présentation et une étude pratique, où différents aspects et fonctions de l'électronique ont été illustrés.

Enfin, notre projet nous a permis, d'approfondir nos connaissances sur les capteurs, particulièrement, les électrodes, l'amplificateur d'instrumentation, le filtrage, l'amplification et bien d'autres aspects des fonctions électroniques entrant dans la conception de l'instrumentation médicale. Plus important encore, c'est d'être confronté aux problèmes pratiques de l'électronique et comment surmonter tous ces problèmes pour mener à bien une réalisation.

Après l'étape de conditionnement, on a observé le signal électrique du cœur localement et puis on a attaqué l'étape de la transmission sans fils qui consiste à stocker les données dans une base de données et les afficher dans une page Web. Les données transmises via un réseau sans fils sont enregistrées pour une éventuelle visualisation future.

ANNEXE :

1. La carte électronique nodemcu :

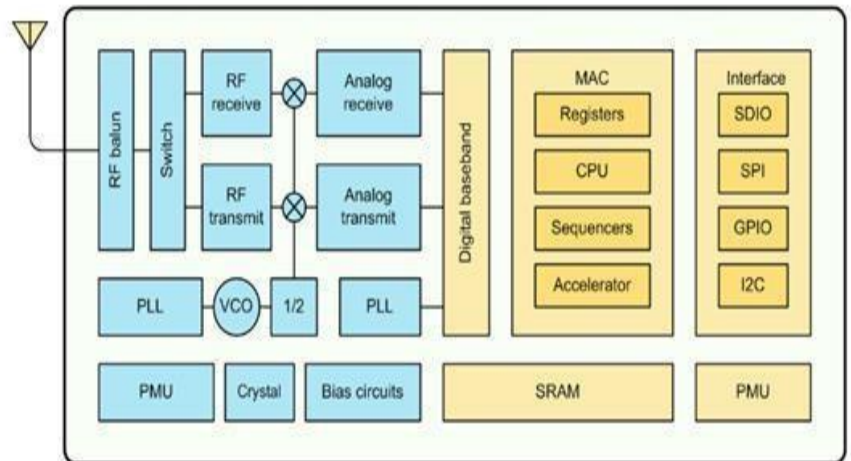


▪ Circuit ESP8266

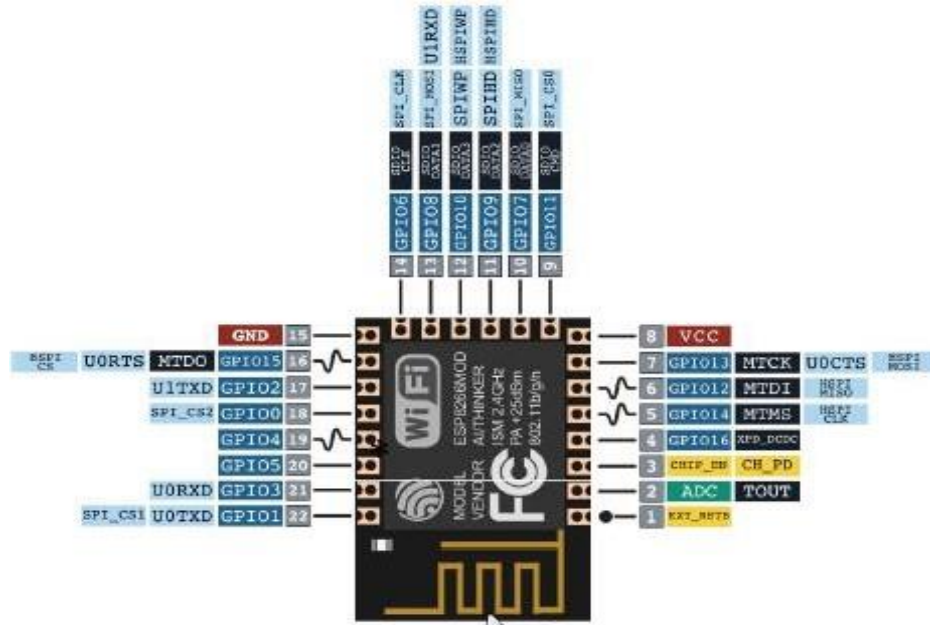
Espressif Systems (Shanghai) Pte.Ltd.

- 32 pin Ultra Low Power
- WIFI 2.4GHz b,g,n WPA/WPA2
- 80/160Mhz (pll)
- 32 bits RTOS inclus (pas visible)
- 80% de la puissance disponible (non wifi)

- Flash externe: 512b-16Mb



▪ Connexion d'ESP8266

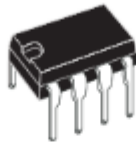


- Reset (High=run, Low=reset)
- ADC
- CH_PD, Enable (High=on, Low=off)
- GPIO16/D0/USER/WEAK
- GPIO14/D5/SPI-SCK
- GPIO12/D6/MISO
- GPIO13/D7/MOSI
- GPIO1/D10 TX0
- GPIO3/D9 RX0
- GPIO5 /D1/SCL(I2C)
- GPIO4 /D2/SDA(I2C)
- GPIO0/D3/FLASH
- GPIO2/D4
- GPIO15/D8/TX2/SPI-CS

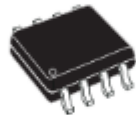
Si 6 pins en bas, elles sont connectées au SPI interne / en parallèle sur la mémoire

- Binary loader
- C code (linux)
- Commandes AT
- Langage LUA

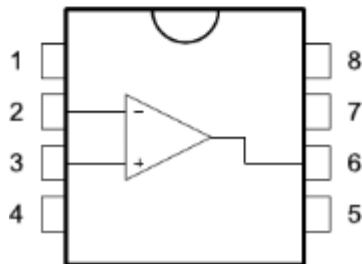
N
DIP8
 (plastic package)



D
S08
 (plastic micropackage)



Pin connections
(top view)



- 1 - Offset null 1
- 2 - Inverting input
- 3 - Non-inverting input
- 4 - V_{CC-}
- 5 - Offset null 2
- 6 - Output
- 7 - V_{CC+}
- 8 - N.C.

Features

- Large input voltage range
- No latch-up
- High gain
- Short-circuit protection
- No frequency compensation required
- Same pin configuration as the UA709

Applications

- Summing amplifiers
- Voltage followers
- Integrators
- Active filters
- Function generators

Description

The UA741 is a high performance monolithic operational amplifier constructed on a single silicon chip. It is intended for a wide range of analog applications.

The high gain and wide range of operating voltages provide superior performances in integrators, summing amplifiers and general feedback applications. The internal compensation network (6 dB/octave) ensures stability in closed-loop circuits.

Bibliographie

<https://canvasjs.com>

<https://www.apachefriends.org/fr/index.html>

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/913893-phpmyadmin>

[http://dlibrary.univ-](http://dlibrary.univ-boumerdes.dz:8080/bitstream/123456789/4229/1/AMRI%20MOHAMED%20CHAKIB.pdf?fbclid=IwAR2aDb0bxxliQl4Unl8TdPQpcnyQgMIAVUsmPfgVMMIGKVLka3s53FcCnrl)

[boumerdes.dz:8080/bitstream/123456789/4229/1/AMRI%20MOHAMED%20CHAKIB.pdf?fbclid=IwAR2aDb0bxxliQl4Unl8TdPQpcnyQgMIAVUsmPfgVMMIGKVLka3s53FcCnrl](http://dlibrary.univ-boumerdes.dz:8080/bitstream/123456789/4229/1/AMRI%20MOHAMED%20CHAKIB.pdf?fbclid=IwAR2aDb0bxxliQl4Unl8TdPQpcnyQgMIAVUsmPfgVMMIGKVLka3s53FcCnrl)

<http://www.elektronique.fr/logiciels/proteus.php?fbclid=IwAR0kjtNHLgrwiUqlirWz4QNXHicYJo4g8-8NIFa2aAVyRuCvulOUUbR-foo>

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/914508-ecrivez-des-donnees>

<https://docs.phpmyadmin.net/fr/latest/require.html#php>