



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

THEME

Conception et réalisation d'un système de télémédecine

Réalisé par : - SANDEL Abdelhak
- DJELLAL Redha

Soutenu le 19/06/2018 Devant le jury composé de:

Mr. LAANNANI Abderrahim	Président	Université Abbes Laghrour-Khenchela
Mr. Sahour Hakim	Encadreur	Université Abbes Laghrour-Khenchela
Mr. SAIGAA Mohamed	Examineur	Université Abbes Laghrour-Khenchela

Promotion 2017/2018

Dédicace

Je dédie ce modeste travail :

A mes très chers parents

A mes frères et sœur

A mon binôme Redha

Et toute personne que je connais et qui me sont chers et tous ceux qui m'aiment.

SANDEL ABDELHAK

Dédicace

Je dédie ce travail à...

Ma mère et mon père qui m'ont donné naissance et l'amour ;
Ce qui je ne pourrais jamais exprimer leurs sacrifices, leur
soutien moral et physique le long de mes années d'études.

Mon très chère sœur.

Mes très chers frères.

Toute ma famille.

Tous mes amis.

Tous ceux que me sont chers

DJELLAL Redha

Remerciements

Remerciements à Dieu - le tout puissant-qui nous a aidé à réaliser ce modeste travail.

Nous tenons à remercier notre promoteur Sahour ABDELHAKIM d'avoir accepter de nous encadrer et de nous suivre durant toute cette période.

Nos remerciements vont aussi au président du jury et aux membres du jury examinateurs qui nous ont fait l'honneur de participer au jury de ce travail.

Et enfin nous remercions l'ensemble, enseignants et collègues de notre promotion, qui nous ont aidés à réaliser ce modeste travail.

Résumé

La surveillance des patients, dans leurs vie quotidienne en dehors des hôpitaux, présentant une maladie chronique ou une pathologie aiguë et ce dans le respect d'une prise en charge efficiente et de qualité sont devenus un phénomène répandu. La raison derrière ce projet est de concevoir un système de télésurveillance du patient à tout moment en utilisant la connectivité Internet. La fonction globale du système proposé est de mesurer certains paramètres physiologiques du corps humain comme par exemple la température, le rythme cardiaque, le pouls, en utilisant des capteurs et de les envoyer par la plate-forme IOT via le un réseau WIFI au Cloud. Toutes les données seront reçues puis stockées sur le cloud, où ils peuvent être visualisés en temps réel en envoyant les données directement sur un PC, un ordinateur portable ou un Smartphone. Ces informations peuvent être utilisées pour surveiller en temps réel l'état d'un patient ou pour obtenir des données sensibles afin d'être ensuite analysé pour un diagnostic médical. Ce qui permet au professionnel de la santé de répondre plus rapidement aux besoins d'un patient. Au-delà de sa contribution à l'amélioration de la qualité de l'offre de soins.

Mots-clés: télésurveillance médicale, télémedecine, capteur physiologique, capteur de rythme cardiaque, capteur de température, capteur de fréquence cardiaque, module Wifi ESP8266, Cloud IOT, microcontrôleur AVR Atmega328, Arduino, page web, système d'information, XAMP, serveur locale. PHP, Java Script, HTML, CSS,

ABSTRACT

Monitoring and Recording of various medical parameters of patient outside hospitals has become widespread phenomenon. The Reason behind this project is to design a system for monitoring the patient's body at any time using internet connectivity. The function of this system is to measuring some biological parameter of the patient's body like Temperature, Heartbeat, pulse rate sensor, by using sensors and the sensors will sense the body temperature, heartbeat and pulse rate sensor of the patient and sends the values to IOT Cloud platform through WIFI-Module. All information about the patient health will be stored on the cloud, it enables the doctors to monitor patient's health, where the doctor can continuously monitor the patient's condition on his computer. The results showed that this project can effectively use Wi-Fi technology to monitor patient health status. And the power consumption of Wi-Fi module (ESP8266) can be reduced as much as possible. Thus, the designed system provides low complexity, low power consumptions and highly portable for healthcare monitoring of patients.

Keywords: medical telemonitoring , Heart beat sensor, Temperature sensor, Pulse Rate sensor, ESP8266 Wifi module ,IOT cloud , AVR Atmega328 Microcontroller, Arduino, web page, XAMP, local server .php,java script,html ,css,

ملخص

وقد أصبحت مراقبة المرضى ، في حياتهم اليومية خارج المستشفيات ، المصابين بمرض مزمن أو بمرض حاد ، وذلك بتقديم الرعاية الجيدة و الفعالة ، ظاهره واسعه الانتشار. والسبب وراء هذا المشروع هو تصميم نظام مراقبه المريض عن بعد في اي وقت باستخدام الاتصال بالإنترنت. وتتمثل الوظيفة العامة للنظام المقترح في قياس بعض المتغيرات الفسيولوجية للجسم البشري ، مثل درجة الحرارة ، ومعدل ضربات القلب ، والنبض ، بإستخدام أجهزه الاستشعار المتصلة بالمتحكم الدقيق الخاص بمنصة انترنت الاشياء وإرسالها عبر انترنت عبر شبكه وأي فاي و سيتم تلقي جميع البيانات وتخزينها تخزينا سحابيا ، حيث يمكن مشاهدتها في الوقت الحقيقي عن طريق إرسال البيانات مباشرة إلى جهاز كمبيوتر محمول ، أو الهاتف الذكي. ويمكن استخدام هذه المعلومات لرصد حالة المريض في الوقت اللحضي أو للحصول علي البيانات الحساسة لكي يتم تحليلها بعد ذلك للتشخيص الطبي. وهذا يسمح لمختصي الصحة للاستجابة بسرعة أكبر لاحتياجات المريض. إلى جانب مساهمتها في تحسين نوعية الرعاية المقدمة.

الكلمة المفتاحية

المراقبة الطبية عن بعد -منصة إنترنت الأشياء - التخزين السحابي-الواي فاي -صفحة أجهزة الاستشعار معدل -الويب-بطاقة الأردوينو- أجهزة الاستشعار الفسيولوجية جهاز استشعار - أجهزة استشعار درجة الحرارة -ضربات القلب لغات البرمجة -
- الخادم المحلي - شبكة محلية - معدل ضربات القلب - إرسال البيانات - شبكة لاسلكية

Dédicace 1	
Dédicace 2	
Remerciement	
Table des matières	
Liste des figures	
Liste des tableaux	
Glossaire	
Introduction générale	3
<u>Chapitre 1 : Généralité sur le système Arduino</u>	5
I.1 Introduction :	6
I.2 Généralité sur les microcontrôleurs :	6
I.3 Rôle d'un système à microcontrôleur [1]:	7
I.4 Généralités sur Les système arduino :	7
I.5 Les différents types d'arduino [3] :	8
a. La carte arduino uno :	8
b. La carte arduino lenardo :	8
c. La carte Arduino Mega :	8
d. La carte Arduino Mega Adk :	9
e. La carte Arduino Due :	9
f. La carte Arduino nano :	9
g. La carte Arduino mini Pro :	10
h. La carte Arduino yun :	10
I.6 cartes d'extension « shield » :	11
I.6.1 Les modules d'affichage :	11
I.6.2 Les modules de prototypage :	11
I.6.3 Les modules de commandes des moteurs :	12
I.6.4 Les modules de Communication sans fil :	12
a. Le module bluetooth :	12
b. Le module GSM :	13

c.	Le module Wifi (ESP 8266):.....	13
I.7	L'environnement de développement Arduino :.....	14
I.7.1.	Présentation :	14
I.7.2	Structure générale du programme (IDE Arduino) :.....	16
I.7.3	Le langage de programmation :.....	16
I.7.3.1	Déclaration des constantes et variables :.....	17
I.7.3.2	Configuration des entrée et sorties :.....	17
I.7.3.3	Programmation des interactions et comportements :.....	18
I.7.3.4	Références du langage (jeu d'instruction) :.....	18
I.8	Conclusion :.....	19
<u>Chapitre 2 : Généralité sur les capteurs et les capteurs biomédicaux</u>		20
II.1	Introduction :.....	21
II.2	Généralités sur les capteurs :.....	21
II.2.1	Chaine de mesure :	22
II.2.2	Types de grandeur physique :.....	22
II.2.3	Modes de fonctionnement des capteurs :.....	23
a.	Capteurs actifs :.....	23
b.	Capteurs passifs :.....	24
c.	Capteurs intelligents :.....	24
II.2.4	Caractéristiques métrologiques du capteur :.....	25
a	Etendue de mesure :	25
b	Sensibilité :.....	26
c	Finesse :.....	26
d	Linéarité :	26
II.2.5	Caractéristiques statiques d'un capteur :.....	27
a	Fidélité :.....	27
b	Justesse :.....	27
c	Précision :.....	27

d	Rapidité :	28
II.2.6	Choix D'un Capteur :	28
II.3	Les capteurs biomédicaux :	28
II.3.1	Introduction:	28
II.3.2	Classement des capteurs biomédicaux:	29
II.3.3	Technologie de mesure biomédicale :	29
II.3.4	les différents capteurs médicaux:	29
a.	Le capteur d'électrocardiogramme ECG :	30
b.	Le capteur de la saturation pulsée en oxygène (SpO2) :	32
c.	Capteur de température :	33
d.	Capteur de pression artérielle :	34
e	Capteur de position et de chutes :	35
f	Capteur de glucometre :	35
g	Capteur de débit d'air :	36
II.4	Conclusion.....	37
<u>Chapitre 3 : Méthodologie de la conception</u>		38
III.1	Introduction :	39
III.2	présentation du système :	39
III.3	Architecture globale du système :	40
III.4	Partie émetteur (au niveau du Patient) :	41
III.4.1	Description du matériel utilises :	42
III.4.1.1	La carte arduino uno :	42
III.4.1.1.1	Pourquoi arduino uno :	42
III.4.1.1.2	Technologie de la carte arduino uno :	43
III.4.1.1.3	Microcontrôleur ATmega328 :	44
III.4.1.1.4	Sources de l'alimentation :	45
III.4.1.1.5	Entrées & sorties :	46
III.4.1.1.6	Les ports de communications :	47
III.4.1.2	Le module Wifi ESP 8266 :	48

TABLE DES MATIERES

III.4.1.2.1	Principales caractéristiques d'ESP-01 :.....	48
III.4.1.2.2	Branchement (ESP-01) :.....	49
III.4.1.2.3	Alimentation du module ESP8266-01 :.....	49
III.4.1.3	Le Capteur de température LM 35:.....	49
III.4.1.3.1	Avantage du capteur LM 35:.....	50
III.4.1.3.2	Caractéristiques du capteur LM 35:.....	50
III.4.1.3.3	Branchement du capteur LM35 :.....	50
III.4.1.4	Le capteur du signal cardiaque AD8232 :.....	50
III.4.1.4.1	Les électrodes.....	52
III.4.1.5	Le Capteur de pouls et la saturation en oxygène dans le sang (SPO2):.....	52
III.4.2	Réalisation pratique du système :.....	54
III.4.2.1	Montage et programmation du module WIFI.....	54
III.4.2.1.1	Configuration d'ESP 8266 avec l'environnement Arduino.....	55
III.4.2.2	Montage et programmation du capteur de température Lm 35:.....	58
III.4.2.3	Montage et programmation PPG pour Arduino :.....	60
III.4.2.4	Montage et programmation ECG pour Arduino :.....	61
III.4.3	Montage globale :.....	62
III.4.3.1	Les Tests avec Commande AT.....	63
III.4.3.1.1	Connexion à un réseau.....	64
III.4.3.1.2	Envoie d'une donnée :.....	64
III.4.3.1.3	Recevoir des données :.....	65
III.4.3.2	Organigramme de la carte de commande.....	65
III.5	Partie récepteur :.....	67
III.5.1	Définition d'un serveur :.....	67
III.5.2	Serveur http :.....	67
III.5.3	L'environnement de développement XAMPP :.....	67
III.5.3.1	Présentation de server local XAMP [14] :.....	67
III.5.4	La base de données :.....	69
III.5.5	La programmation de la page de la réception :.....	70

TABLE DES MATIERES

III.5.6	Le serveur de bases de données MySQL :.....	70
III.5.7	La programmation de la page de l’affichage :.....	71
III.5.8	Organigramme de la partie serveur :	72
III.6	Présentation de l'interfaçage :.....	73
III.6.1	Interface principale :.....	73
III.6.2	Interface de la température :.....	75
III.6.3	Interface ECG :.....	76
III.6.4	Interface La saturation pulsée en oxygène (SpO2):.....	77
III.6.5	Éléments de graphique	78
III.7	Conclusion.....	80

Liste des figures

Figure 1.1 : Les microcontrôleurs	6
Figure 1.2 : système à microcontrôleur	7
Figure 1.3 : La carte Arduino UNO	8
Figure 1.4 : La carte Arduino Lenardo	8
Figure 1.5 : La carte Arduino Mega ADK	9
Figure 1.6 : La carte Arduino Due	9
Figure 1.7 : La carte Arduino Nano	10
Figure 1.8 : La carte Arduino Mini Pro	10
Figure 1.9 : La carte Arduino Yun	10
Figure 1.10 : shield 7 segment (i2c)	11
Figure 1.11 : shield afficheur graphique	11
Figure 1.12: module shield "LCD 2 x 16 characters	11
Figure 1.13 : proto shield pour arduino	12
Figure 1.14 : proto-screwshield pour arduino	12
Figure 1.15: shield mega proto pcb r3	12
Figure 1.16 : Dual Dc Motor Shield	12
Figure 1.17 :Arduino Motor Shield R3	12
Figure 1.18: Adafruit Motor/Stepper/Servo Shield Pour Arduino	12
Figure 1.19 : Type de modules Bluetooth	13
Figure 1.20 : module GSM d'Arduino	13
Figure 1.21 : Les modules ESP8266	14
Figure 1.22 : Interface de la plateforme Arduino	15
Figure 1.23 : Barre de boutons Arduino	15
Figure 1.24 : HyperTerminal de l'Arduino (Moniteur Série)	16
Figure 1.25 : Structure générale du programme (IDE Arduino)	16
Figure 1.26 : Structure d'un programme Arduino. [10]	17
Figure 1.27 : Déclaration des constantes et variables	17
Figure 1.28 : La fonction de configuration	17
Figure 1.29 : Programmation des instructions et des comportements.	18
Figure 2.1: Définition d'un capteur	21
Figure 2.2 : Schéma bloc d'un capteur	21
Figure 2.3 : constitution d'une chaîne de mesure classique	22
Figure 2.4 : Capteur intelligent	25
Figure 2.5 : Courbe d'étalonnage d'un capteur	26
Figure 2.6 : Exemple de linéarisation de caractéristique	27
Figure 2.7 : (a) et (b) les électrodes de l'ECG à l'usage unique et (c) le jeu de fils d'électrodes	30

Liste des figures

Figure 2.8 : Les électrodes d'ECG utilisés.....	31
Figure 2.9 : les différentes formes du capteur du PPG,.....	31
Figure 2.10 : capteur de rythme cardiaque (pulse sensor).....	32
Figure 2.11 : principe de fonctionnement d'un capteur	32
Figure 2.12 : Le capteur de température.....	33
Figure 2.13 : Capteur de pression artérielle Particularités:	33
Figure 2.14 : Capteur de position et les chutes.....	34
Figure 2.15 : les positions de corps	35
Figure 2.16 : Capteur de glucomètre.....	35
Figure 2.17 : Capteur de débit d'air	36
Figure 2.18 : Placez le capteur comme indiqué dans l'image	36
Figure 3.1 : schéma bloc du système à réaliser.	39
Figure 3.2 : Architecture globale du système.....	40
Figure 3.3 : Schéma bloc de la Partie émetteur (au niveau du patient).	41
Figure 3.4 : Arduino Uno.	42
Figure 3.5: Microcontrôleur ATmega328.....	44
Figure 3.6: Sources de l'Alimentation de la carte Arduino UNO.	46
Figure 3.7 : Constitution de la carte Arduino UNO.	48
Figure 3.8: le module wifi ESP8266.	48
Figure 3.10 : Brochage du lm 35.....	50
Figure 3.12 : les électrodes, ECG.....	52
Figure 3.13 : Le Capteur (spo2).	53
Figure 3.14 : la position d'un spo2.....	53
Figure 3.15: vue de prototypage du Montage d'esp8266 avec L'Arduino.....	54
Figure 3.18: Vue prototypage du montage LM35 avec l'Arduino UNO.....	58
Figure 3.20 : Vue prototypage du montage de PPG avec Arduino Uno	60
Figure 3.21 : le code PPG.....	60
Figure 3.22 : Branchement ECG avec Arduino Uno.....	61
Figure 3.24 : montage prototypage.....	62
Figure 3.25 : Schéma électronique du montage	62
Figure 3.26 : Photo de montage fini	63
Figure 3.27 : la connexion de l'esp8266 avec le modem.	64
Figure 3.28 : la transmission des données dans le protocole TCP.	64
Figure 3.29 : la transmission des données.	65
Figure 3.30 : la réception des valeurs de température et ECG dans la base de données.	65
Figure 3.32 : choix de la langue d'installation.....	68
Figure 3.33 : Panneau de contrôle de XAMPP.....	69

Liste des figures

Figure 3.34 : Une base de données permet de stocker la totalité des informations relatives à une activité.....	69
Figure 3.35 : Code PHP de page de réception.....	70
Figure 3.36 : Code PHP de la page de réception.....	71
Figure 3.37 : La programmation de la page de l’affichage :	71
Figure 3.38:Diagramme du programme de la partie serveur.....	72
Figure 3.40 : Les données de température.....	74
Figure 3.41: Les données D’ECG	74
Figure 3.42 : Pagination de signal ECG	74
Figure 3.43 : Les données de pouls	75
Figure 3.45 : La courbe de température.....	76
Figure 3.46: Interface ECG	76
Figure 3.47 : Le signal d’ECG	77
Figure 3.48: Interface La saturation pulsée en oxygène (SpO2)	77
Figure 3.49 : Signal de La saturation pulsée en oxygène (SpO2)	78
Figure 3.50: le signal ECG	78
Figure 3.52 : Positionner de température.	79
Figure 3.53 : Enregistrer le graph de l'ECG forme JPEG ou PNG.....	79
Figure 3.54 : imprimer l'ECG sur papier.....	80

LISTE DES TABLEAUX

Tableau 2.1 : Capteurs actifs : principes physiques de base.....	23
Tableau 2.2 : Capteurs passifs : principes physiques et matériaux.....	24
Tableau 3.1 : Caractéristiques de la Carte Arduino UNO.	43
Tableau 3.2 : connecter le capteur ECG avec Arduino.....	51
Tableau 3.3 : les couleurs de câble et position	52
Tableau 3.4 : connexion du capteur PPG avec Arduino.....	53

GSM: Global System for Mobile Communications

GPRS: General Packet Radio Service

WIFI: wireless fidelity

IDE: integrated development environment

ECG : électrocardiogramme

EMG : électromyographie

EEG : électroencéphalographie

SPo2 : The pulsed oxygen saturation

IOT : internet of things

µc : Microcontrôleur

TCP/IP : Transmission Control Protocol/Internet Protocol.

INTRODUCTION
GÉNÉRALE

Introduction générale :

La technologie a toujours été là pour simplifier et rendre la vie humaine beaucoup plus facile. Elle a affecté les différents domaines de la vie ; le domaine médical est l'un d'eux. Il bénéficie de la technologie de différentes façons ; maintenant cela se traduit dans la télémédecine qui constitue une nouvelle forme de pratique médicale dont le développement est aujourd'hui en pleine expansion. La télémédecine est présentée comme une solution prometteuse. Il est plus facile de diagnostiquer les maladies à l'aide de certains appareils numériques. Le secteur des soins de santé traverse un énorme changement, avec des capacités numériques changeant la façon dont les médecins interagissent avec leurs patients. De nos jours, les patients ont les outils pour visualiser eux-mêmes leurs signes vitaux clés et aider les médecins à avoir un accès immédiat aux données des patients. Dans une forme de dispositifs portables par exemple des montres Apple [1].

Cependant, nous notons que ces dispositifs sont assez chers et il est difficile de trouver ces dispositifs utilisés par les pauvres, qui sont confrontés à la plus grande partie de la charge des maladies. En outre, la plupart des dispositifs médicaux existants ont encore besoin de quelques investigations une fois qu'il vient au concept de l'Internet des Objets.

Le concept de l'IOT (internet of things) implique l'utilisation d'appareils électroniques qui captent ou surveillent les données et sont connectés à un nuage privé ou public, ce qui leur permet de déclencher automatiquement certains événements. Les données médicales telles que la pression artérielle et la fréquence cardiaque sont collectées par des capteurs sur les périphériques ; ces données sont transmises aux fournisseurs de soins de santé ou à des tiers via des dispositifs de télécommunication sans fil. Les données sont évaluées pour les problèmes potentiels par un professionnel de la santé et les fournisseurs de soins de santé sont immédiatement alertés si un problème est détecté [2]. Par conséquent, l'intervention en temps opportun assure des résultats positifs pour les patients. Deux groupes de personnes, qui ont des niveaux élevés de soins médicaux, sont les personnes âgées et les malades chroniques. L'Organisation mondiale de la santé (2016) indique que les personnes âgées, qui souffrent fréquemment de maladies chroniques, nécessitent une prestation de soins très efficace et efficace [1]. Selon le national Broadband plan (2015) de la Fédéral Communications Commission (FCC), l'utilisation de la surveillance des patients à distance sous la technologie de l'ITO permettra d'économiser l'industrie des soins de santé \$700 milliards de plus de 15 à 20 ans [3]. Ensuite, il est important de se concentrer sur la technologie de l'ITO pour bénéficier de la technologie la plus récente.

Bluetooth, ZigBee et Wi-Fi sont les technologies sans fil communes pour les systèmes de surveillance des patients à distance. Cependant, leur pertinence et leur convivialité pour cette tâche sont très variables. Par conséquent, il y a des considérations importantes tout en choisissant une technologie pour les dispositifs médicaux d'IOT; vous devez être conscient de ce que leurs caractéristiques pour des exigences spécifiques.

Dans ce contexte nous nous sommes intéressés à la conception d'un système électronique (système embarqué). Le système proposé aidera les patients dans des endroits éloignés (p. ex. soins de santé à domicile) et il ne s'agit pas seulement de surveiller un état de maladie chronique, mais d'aider à empêcher les patients de se rendre à cet État. En mettant en œuvre ce projet, il offrira un système peu coûteux pour la téléconsultation, la télé-expertise, la télésurveillance médicale, la téléassistance médicale, la réponse médicale ou régulation médicale.

Présentations Générale de projet:

Dans ce travail dans le cadre d'une mémoire de fin d'étude, un système de télésurveillance médicale mobile est présenté, qui est capable de surveiller continuellement quelque paramètres physiologiques, peut aussi être étendue à d'autre suivant les besoin.

Nous avons proposé une conception de surveillance et de contrôle continu de l'état d'un patient. Par l'acquisition de données à distance à l'aide de la communication sans fil basée sur le module Wi-Fi et un système d'information. Le travail futur du projet est très essentiel afin de rendre le système de conception plus avancé. Dans le système conçu l'amélioration serait de relier plus de capteurs à Internet qui mesure divers autres paramètres de santé et serait bénéfique pour le suivi des patients, c'est à dire la connexion de tous les objets à Internet pour un accès rapide et facile.

Présentation du mémoire :

Pour bien attendre nos objectifs nous avons organisé le travail comme suit :

Le premier chapitre sera consacré à donner des généralités sur le système Arduino. Puis en exposant la configuration de base des différentes cartes Arduino ainsi que l'environnement de programmation Arduino.

Dans le deuxième chapitre nous donnons en premier temps des généralités sur les capteurs ainsi leur mode de fonctionnement, dans un deuxième temps nous devons présenter les capteurs biomédicaux leur prospérité et leur classification.

Dans le troisième chapitre, on présentera la méthodologie de la conception et de la réalisation de notre application, nous commençons par donner une vie globale de la

conception, la présentation des différents éléments. Par la suite on présentera une explication détaillé de chaque partie et en terminera par les différentes étapes de test.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude: les différents résultats obtenus expérimentalement, et ensuite nous allons essayer de lister des perspectives de notre système. Autrement dit, nous allons présenter les améliorations qui peuvent y être apportées.

CHAPITRE 1 :

Généralité sur le système
Arduino

I.1 Introduction :

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation.

On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

I.2 Généralité sur les microcontrôleurs:

Un microcontrôleur est un circuit intégré qui rassemble aux éléments essentiels d'un ordinateur : Processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités de périphériques et interfaces d'entrées/sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc. [1]:



Figure 1.1 : Les microcontrôleurs

I.3 Rôle d'un système à microcontrôleur [1]:

Un système à microprocesseur permet :

- D'acquérir des entrées logiques et analogiques représentant l'état du système technique,
- D'interpréter, la signification de ces entrées,
- De calculer, mémoriser, récupérer des variables logicielles intermédiaires,
- De gérer le temps,
- D'agir sur des sorties logiques et analogiques en fonction des entrées et des calculs réalisés de manière à modifier le fonctionnement du système technique (commande moteur, affichage d'informations,...)
- ✓ De communiquer par des liaisons séries avec d'autres systèmes techniques et/ou un ordinateur

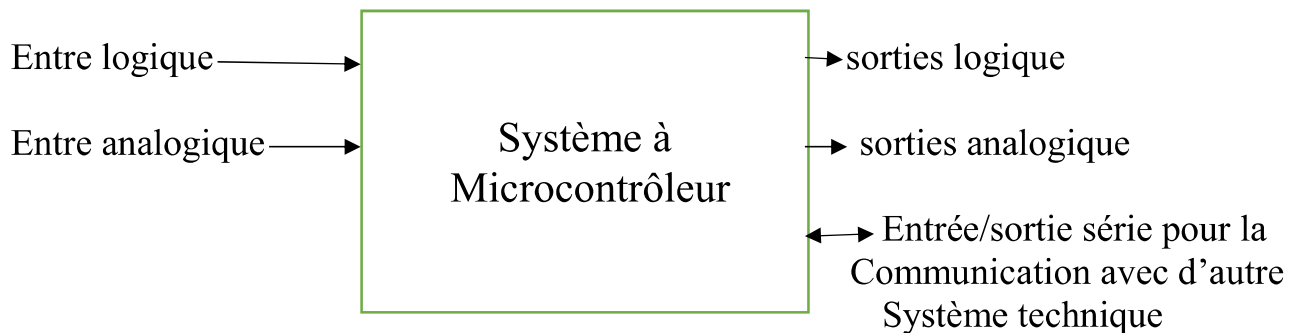


Figure 1.2 : système à microcontrôleur

I.4 Généralité sur le système arduino :

Le système Arduino est un outil pour fabriquer de petits ordinateurs qui peuvent capter et contrôler davantage de choses du monde matériel. C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées; le logiciel de développement open-source peut être téléchargé gratuitement. [2]

I.5 Les différents types de cartes arduino [3] :

a. La carte arduino Uno :

C'est la carte idéale pour découvrir l'environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c'est la carte la plus utilisée, il est très facile de se référer aux tutoriels très nombreux sur le net et ainsi de ne pas rester seul dans son exploration. Sa simplicité devient par contre un handicap lorsqu'il s'agit de multiplier les périphériques, de manipuler des algorithmes lourds ou d'interagir avec les OS Androïde pour lesquels d'autres cartes arduino sont plus adaptées.



Figure 1.3 : La carte Arduino UNO

b. La Carte arduino lenardo :

C'est la carte qui est prévue pour succéder à la carte Arduino Uno en présentant des caractéristiques équivalentes mais une ergonomie revue et une stabilité plus éprouvée. Sa diffusion moins importante limite le support utilisateur disponible sur le net



Figure 1.4 : La carte arduino lenardo

c. La carte arduino mega :

La carte Arduino Mega est la carte la plus diffusée après la carte Arduino Uno. Elle offre un nombre d'entrées/sorties beaucoup plus important (54 contre 14), un processeur plus puissant doté d'une mémoire plus vaste qui permet d'exploiter des algorithmes plus complexes.

d. La Carte arduino mega adk :

La carte Arduino mega ADK offre les mêmes caractéristiques techniques que la carte Arduino mega mais son port USB permet de la connecter avec un environnement Androïde ouvrant de nouvelles perspectives d'interaction avec le monde des smartphones et des capteurs dont ils sont dotés. Sa mise en œuvre nécessite par contre de solides connaissances en Java et la capacité à développer ses propres applications



Figure 1.5 :La carte Arduino Mega ADK

e. La Carte arduino due :

La carte Arduino Due est une évolution de la carte Arduino Mega et offre des performances réputées 3 fois supérieures. Elle permet de manipuler rapidement des algorithmes lourds particulièrement utiles dans le monde de la robotique par exemple.

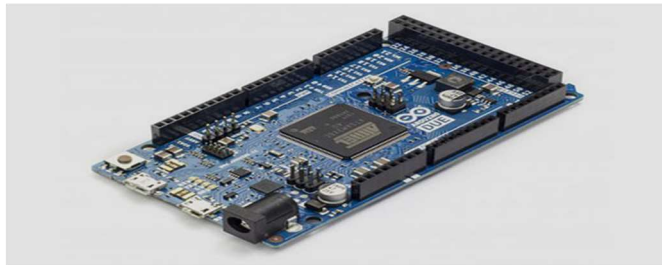


Figure 1.6 : La carte Arduino Due

f. La carte arduino nano :

La carte Arduino nano n'est ni plus ni moins qu'une carte Arduino Uno miniaturisée. Sa taille et son poids réduits la destinent à une utilisation dans des espaces réduits (en textile par exemple) ou dans des applications de robotique ou de modélisme pour lesquels le poids et la taille sont des facteurs déterminant (hélicoptères, drones...)

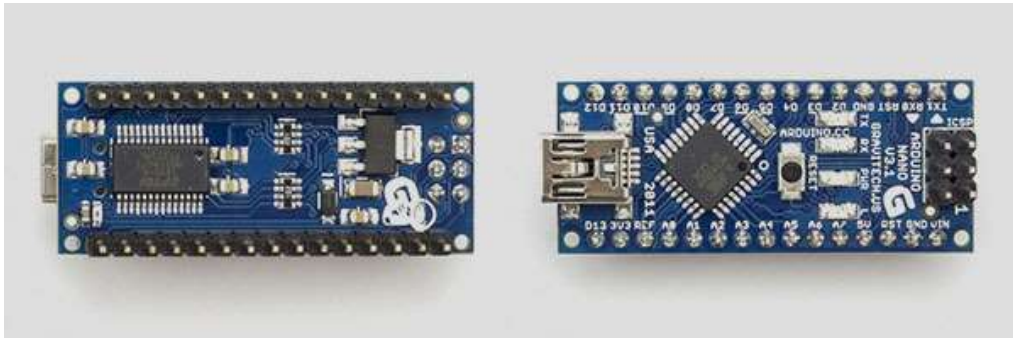


Figure 1.7 : La carte Arduino Nano

g. La carte arduino mini pro :

La carte arduino Mini Pro est une carte Arduino Uno simplifiée à l'extrême permettant néanmoins de piloter de petits projets ou certains éléments d'un projet. Attention, cette carte n'intègre pas de port USB ce qui rends sa connectivité délicate.

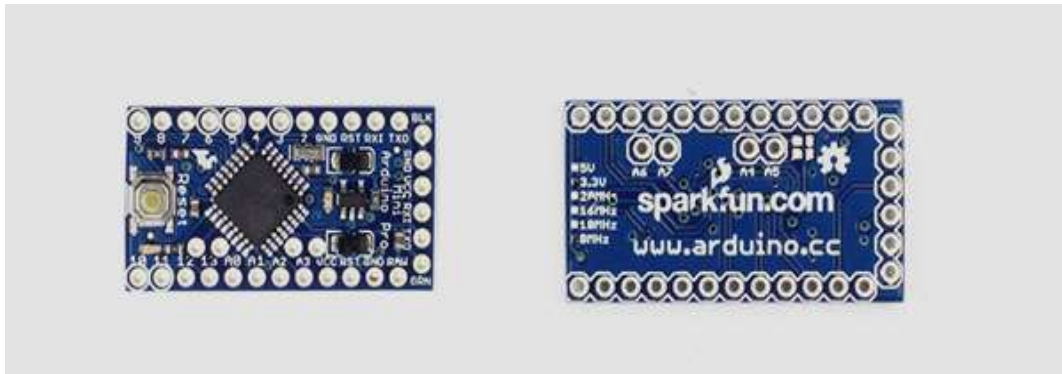


Figure 1.8 :La carte Arduino Mini Pro

h. La carte arduino yun :

La carte Arduino Yun, récemment proposée par Arduino, est conçue pour contrer les avantages de la carte Raspberry. Elle est un dérivé de la carte Leonardo et a pour objectif de combiner la puissance de Linux avec la facilité d'utilisation d'une carte Arduino. Elle est également la première carte Arduino à être dotée nativement d'un wifi intégré.



Figure 1.9 : La carte Arduino Yun

I.6 Les cartes d'extension « shield » :

Une carte Arduino seule, même si vous disposez de quelques LED, résistances, boutons poussoir, etc. n'est pas d'une grande utilité et peut rester limitée à des fins d'apprentissage.

Un des atouts majeurs de l'Arduino est qu'il existe justement une multitude de fabricants proposant des cartes d'interface aussi puissantes, appelées shield dans la terminologie Arduino, Ces extensions augmentent l'ergonomie et la facilité de programmation. Ils sont capables de couvrir la plupart des besoins d'une application embarquée : capteurs, relais de puissance, commande de moteurs, Internet, affichage sur matrice LED ou écran LCD, communication Wifi...

I.6.1 Les modules d'affichage :

Vous trouverez ci-dessous une sélection complète de platines shield pour Arduino (ou compatibles) conçues pour ajouter des possibilités d'affichage numérique sur digits 7 segments à leds, sur matrices à leds RGB, sur rubans à leds, sur écrans LCD alphanumériques, sur écrans graphiques N&B ou couleur [4]



Figure 1.10 : shield 7 segment (i2c)



Figure 1.11 : shield afficheur graphique

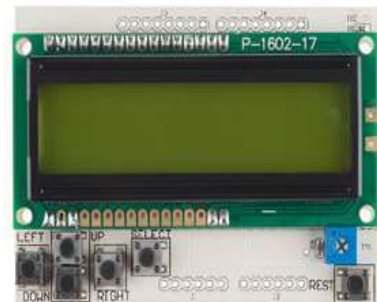


Figure 1.12: module shield "LCD 2 x 16 characters"

I.6.2 Les modules de prototypage :

Vous trouverez ci-dessous une sélection complète de connecteurs mâles/femelles et de platines shield destinées à faciliter les phases de prototypage sur platines Arduino Uno (ou Arduino Mega2560). Ces platines disposent d'une zone de prototypage vierge pastillée avec pour certaines des emplacements pour des circuits intégrés standards ou des circuits intégrés CMS ainsi que des reprises multiples des ports d'entrées/sorties de l'arduino sur des connecteurs ou des borniers à vis. [4]

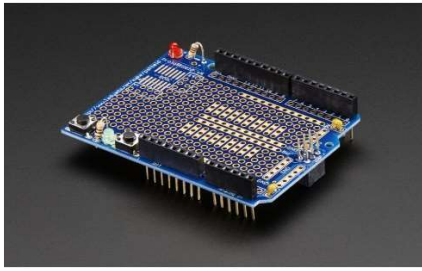


Figure 1.13 : proto shield pour arduino

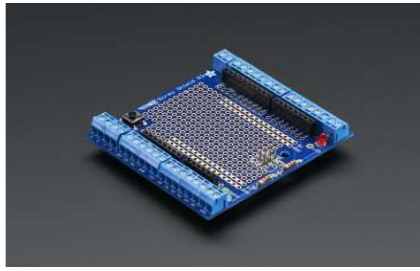


Figure 1.14 : proto-screwshield pour arduino

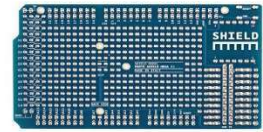


Figure 1.15: shield mega proto PCB r3

I.6.3 Les modules de commandes des moteurs :

Vous trouverez ci-dessous une sélection complète de platines shield pour Arduino™ ou compatibles conçues pour leur ajouter des possibilités de pilotage de moteurs "cc", de moteurs pas-à-pas et de servomoteurs ainsi que des platines shield dédiées à la réalisation de robot mobiles. [5]



Figure 1.16 : Dual Dc Motor Shield

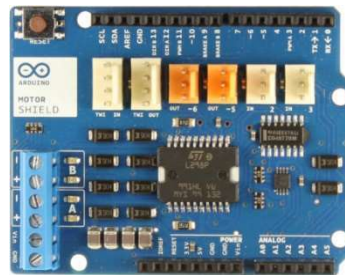


Figure 1.17 :Arduino Motor Shield R3

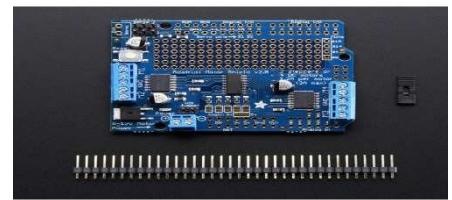


Figure 1.18: Adafruit Motor/Stepper/Servo Shield Pour Arduino

I.6.4 Les Modules de communication sans fil :

Modules Bluetooth, Wifi ou GPRS compatibles Arduino destinés à la communication sans fil.

a. Le module Bluetooth:

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.[6]

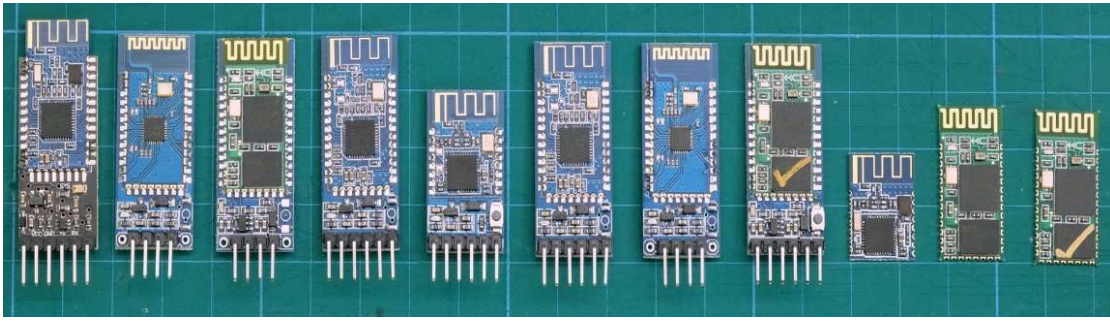


Figure 1.19 : Type de modules Bluetooth

b. Le module GSM :

Le kit d'Arduino GSM connecte Arduino à l'Internet en utilisant le réseau sans fil GPRS. Il suffit de brancher ce module sur une carte Arduino, et on branche une carte SIM d'un opérateur offrant une couverture GPRS et on suit quelques instructions simples pour commencer à contrôler notre monde à travers l'Internet. Nous pouvons également émettre/recevoir des appels vocaux à l'aide de la prise audio/MIC embarquée et envoyer/recevoir des messages SMS. [7]



Figure 1.20 : module GSM d'Arduino

c. Le Module Wifi (ESP 8266):

Wifi ESP8266 est le nom du microcontrôleur développé par Espressif Systems qui est une société basée à Shanghai. Ce microcontrôleur a la capacité d'effectuer des activités liées au Wifi donc il est largement utilisé comme un module Wifi.

Il existe de nombreux types de module ESP8266 disponibles allant d'ESP8266-01 à ESP8266-12. [8]



Figure 1.21 : Les modules ESP8266

I.7 L'environnement de développement Arduino :

Une carte Arduino, que sa construction se base sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

I.7.1 Présentation :

L'interface de l'IDE Arduino est plutôt simple voir figure 2.16, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console pour afficher les résultats de la compilation du code source, des opérations sur la carte ,etc.

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le **C**, le **C++**, le **Java** et le **Processing**. Le langage impose une structure particulière typique de l'informatique embarquée [9].

- ✓ La fonction « **Setup** » contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, débits de communications série, etc.).
- ✓ La fonction « **Loop** », elle est exécutée en boucle après l'exécution de la fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset). Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation. En effet, si l'on omettrait cette boucle, à la fin du

code produit, il sera impossible de reprendre la main sur la carte Arduino qui exécuterait alors du code aléatoire.

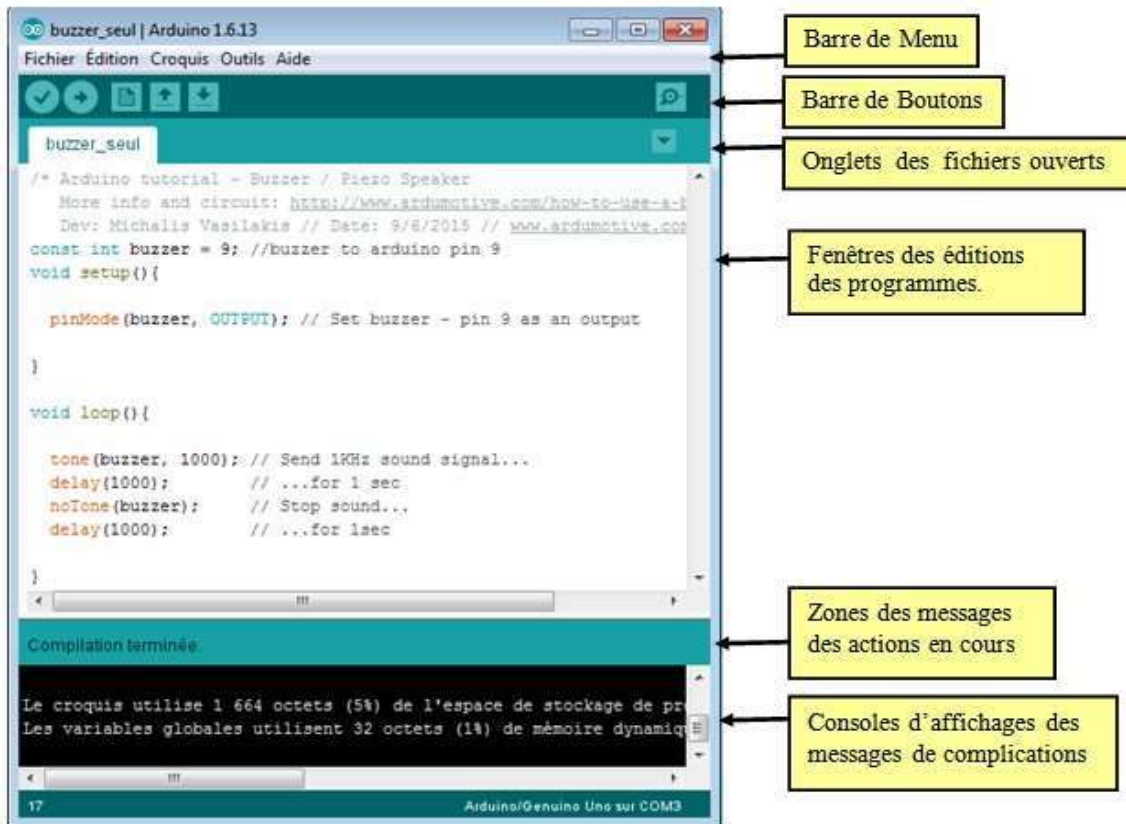


Figure 1.22 : Interface de la plateforme Arduino

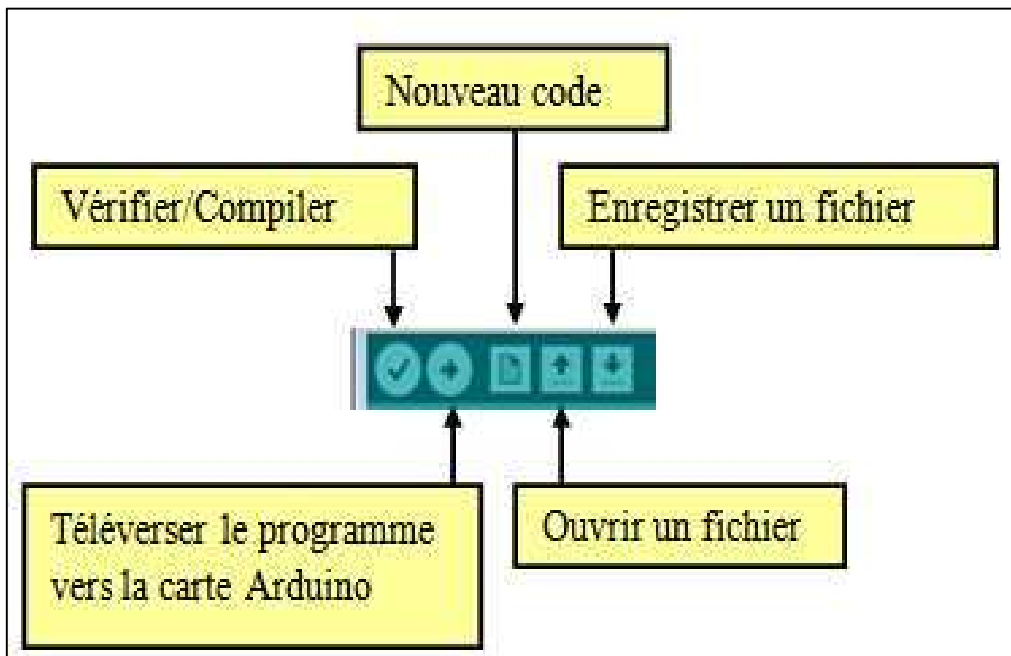


Figure 1.23 : Barre de boutons Arduino

Le logiciel comprends aussi un moniteur série (Equivalent à HyperTerminal) qui permet de d'afficher des messages textes émis par la carte Arduino et d'envoyer des caractères vers la carte

Arduino (en phase de fonctionnement):



Figure 1.24 : HyperTerminal de l'Arduino (Moniteur Série)

I.7.2 Structure générale du programme (IDE Arduino) :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code. La structure d'écriture d'un programme sous Arduino est de la forme suivante:

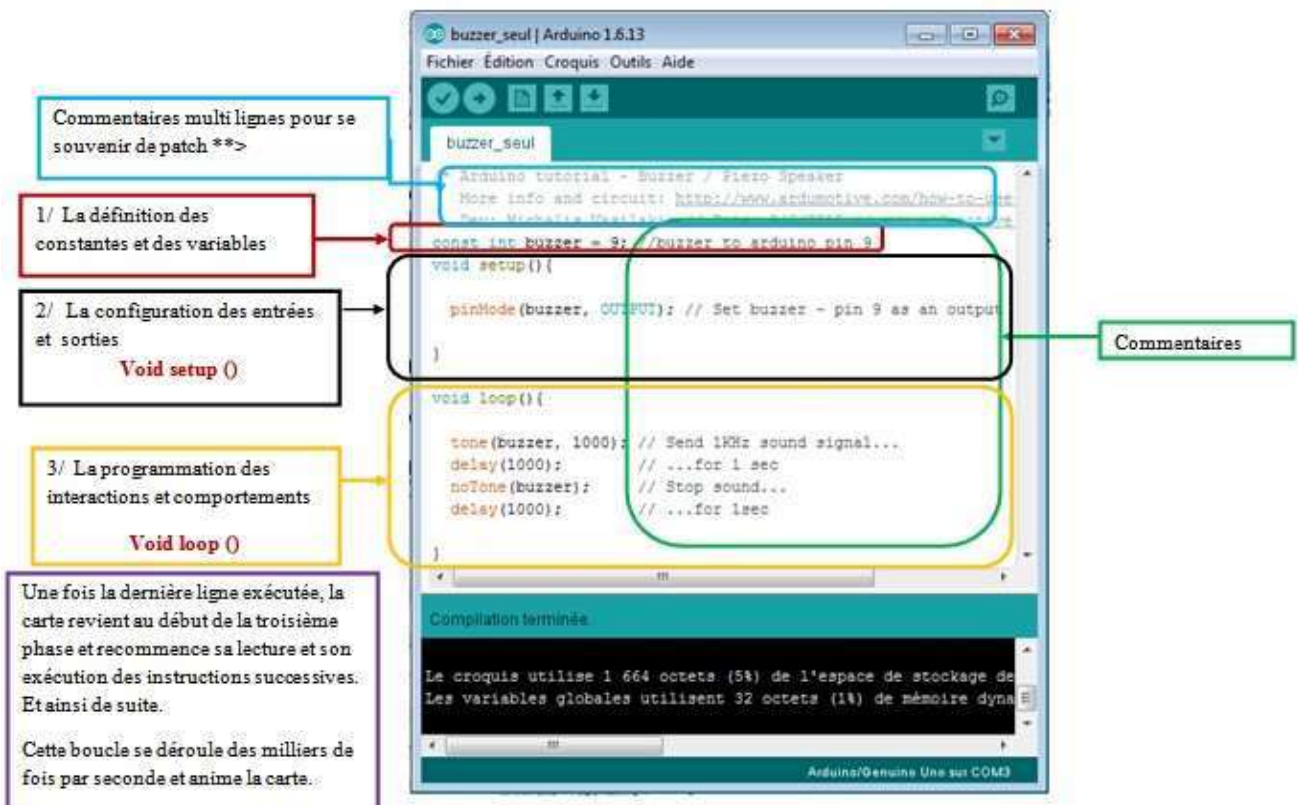


Figure 1.25 : Structure générale du programme (IDE Arduino).

I.7.3 Le langage de programmation :

Un programme Arduino (ou sketch) comporte trois parties (Figure 1.26):

1. La partie déclaration des variables (*optionnelle*).

2. La partie initialisation et configuration des entrées et sorties : la fonction **setup ()**.

3. La partie principale qui s'exécute en boucle : la fonction **Loop ()**.

Ses deux dernières parties sont représentatives de la manière de fonctionnement d'un microcontrôleur et sont appelées dans cet ordre. En effet, un microcontrôleur est destiné à exécuter les instructions contenues dans sa mémoire de manière cyclique et infinie.

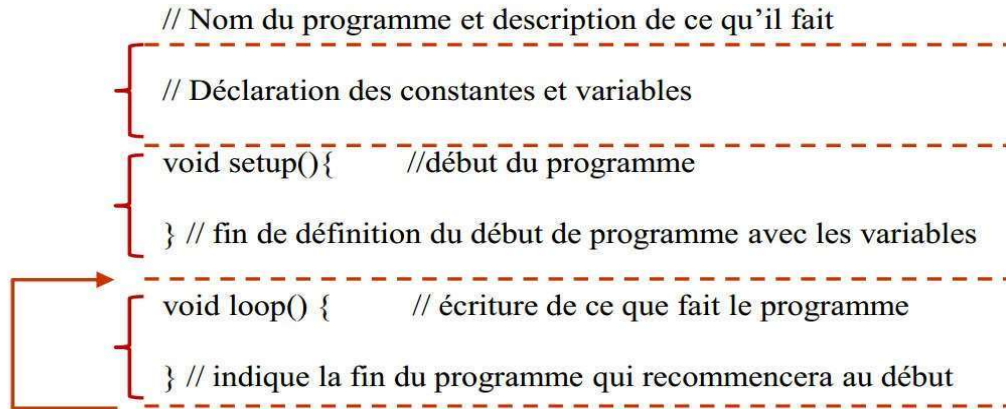


Figure 1.26: Structure d'un programme Arduino. [10]

I.7.3.1 Déclaration des constantes et variables:

Dans cette partie on déclare les constantes et les variables ; type, nom, valeur, broche.

Aussi, on déclare les bibliothèques nécessaires pour le programme, et crée les objets.

```

// Partie déclaration des constantes et variables
// et appel des bibliothèques

void setup() {
  
```

Figure 1.27 : Déclaration des constantes et variables.

1.7.3.2 Configuration des entrée et sorties :

Avant de s'exécuter, le microcontrôleur a besoin d'une phase de « configuration » de son environnement, notamment pour paramétrer des broches d'entrées/sorties selon le besoin de programme, cette phase correspond à la fonction setup ().

```

void setup() {
  // put your setup code here, to run once:
}
  
```

Figure 1.28 : La fonction de configuration.

Les instructions contenues dans cette fonction ne sont exécutées qu'une seule fois, et ce, lors de mise sous tension de la carte.

I.7.3.3 Programmation des interactions et comportements:

Une fois la configuration effectuée, le microcontrôleur débute l'exécution des instructions contenues dans sa mémoire. Cette phase correspond à la fonction `loop()`.

Cette phase est le cœur de programme, elle contient les instructions, les opérations, les boucles,...etc.

```
void loop() {
  // put your main code here, to run repeatedly:
}
```

Figure 1.29 : Programmation des instructions et des comportements.

Cette fonction s'apparente à une boucle ne possédant aucune condition d'arrêt. Le seul moyen de l'arrêter est de couper l'alimentation de la carte ou de réinitialiser la carte. Les instructions contenues dans cette fonction sont donc répétées aussi longtemps que la carte sera alimentée.

1.7.3.4 Références du langage (jeu d'instruction):

Le langage Arduino possède un jeu d'instruction très riche, qui peut être divisé en trois parties valeurs (constantes et variables), structures, et fonctions.

1. **Les valeurs :** qui peuvent être constante ou variables, numériques (byte, int, double, word,), logiques (booleen,.....), tableaux (array), caractères ou chaînes de caractères (char, string), ou constantes particulières (HIGH/LOW, INPUT/OUTPUT, True / False, ...), ...etc.

2. **Les structures :**

Les structures de contrôle comme les boucle (for, while, if-else,....).

Les opérateurs arithmétiques simple (+, -, *, /), composés (+=, -=, /=), et logiques (&&, \!), et les opérateurs de comparaison (=, <, >,....).

3. **Les fonctions :**

Les fonctions mathématiques et trigonométriques (cos,sin,....).

Gestion du temps (delay, millis (),...)

Gestion des entrées et sorties numériques (pinMode, digitalWrite, digitalRead) et analogiques (analogWrite,analogRead)

1.8 Conclusion :

Dans ce chapitre, nous avons projeté la lumière sur le système de développement Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Le chapitre suivant sera consacré à l'étude des capteurs biomédicaux utilisé dans le domaine de la télémédecine.

CHAPITRE 2 :

Généralité Sur Les Capteurs Biomédicaux

II.1 Introduction :

Dans le domaine médical, les capteurs peuvent être utilisés pour assurer une surveillance permanente sur l'état de l'être humain. Les informations fournies par ces dispositifs suivant leurs types peuvent être exploités pour aider au diagnostic médical.

II.2 Généralités sur les capteurs :

Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable, exemple : une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille... On fait souvent (à tort) la confusion entre capteur et transducteur :

Le capteur est au minimum constitué d'un transducteur

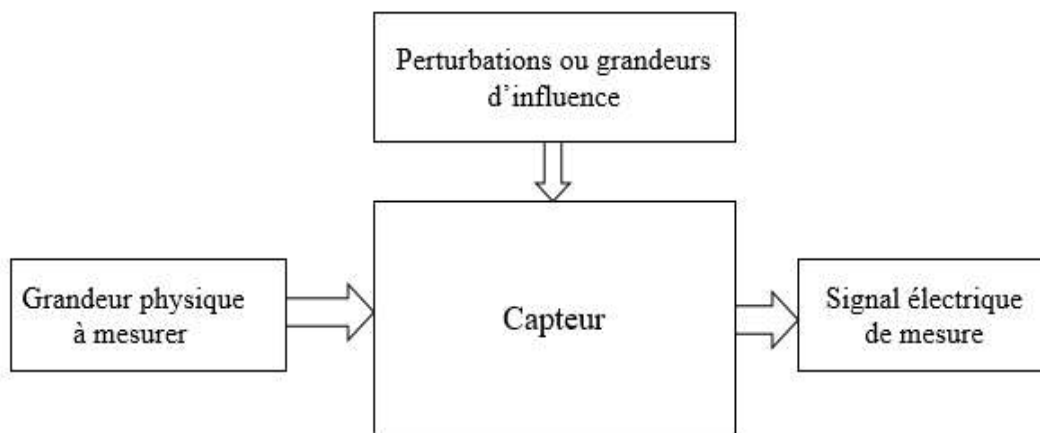


Figure 2.1: Définition d'un capteur.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation [1].

Généralement, on obtient une grandeur de sortie du type électrique. Elle peut être soit : Une charge, une tension, un courant et une impédance (R, L, C).

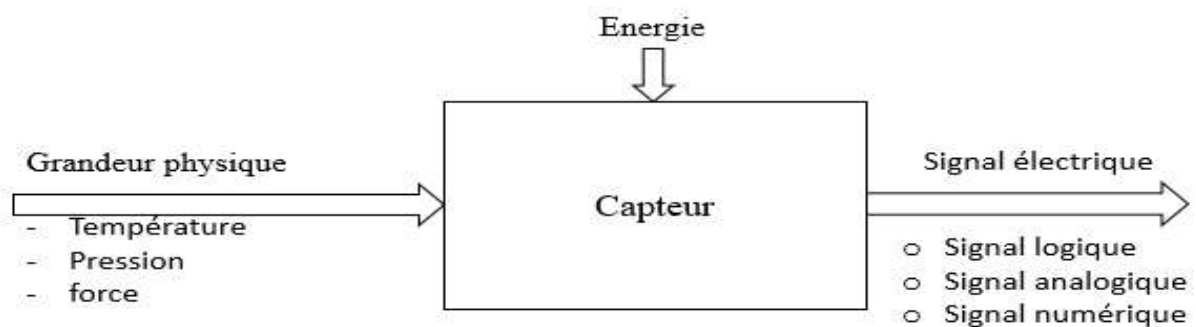


Figure 2.2: Schéma bloc d'un capteur.

II.2.1 Chaîne de mesure :

Pour obtenir une image d'une grandeur physique, on fait appel à une chaîne de mesure qui peut faire intervenir plusieurs phénomènes différents. Par exemple, la mesure d'un débit peut se faire en plusieurs étapes :

- Transformation du débit en une pression différentielle.
- Transformation de la pression différentielle en la déformation mécanique d'une membrane.
- Transformation de la déformation mécanique en une grandeur électrique (à l'aide d'un piézo- électrique) via un circuit électronique associé.

L'ensemble de ces étapes constitue la chaîne de mesure :

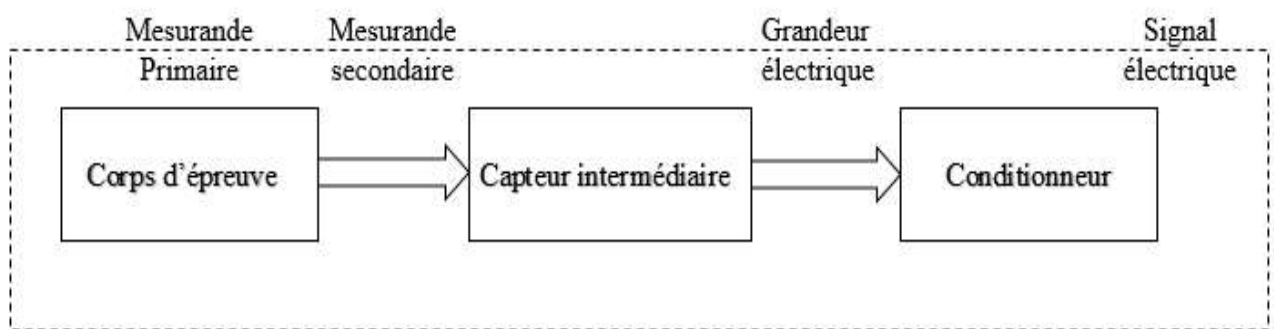


Figure 2.3 : schéma bloc de la chaîne de mesure classique.

De manière classique la sortie d'une chaîne de mesure est du type électrique. Si la chaîne de mesure fait intervenir plusieurs transducteurs.

On appelle corps d'épreuve celui qui est en contact direct avec la mesurande, le dernier transducteur est associé à un conditionneur qui fournit la grandeur électrique de sortie de manière exploitable.

Le choix de conditionneur est une étape importante dans le cadre de la chaîne de mesure car, associé au capteur, il détermine la nature finale du signal électrique et va influencer sur les performances de la mesure.

II.2.2 Types de grandeur physique :

- ✓ On peut classer les grandeurs physiques en 6 familles, chaque capteur s'associant à l'une d'entre elles :
- ✓ Mécanique : déplacement, force, masse, débit...
- ✓ Thermique : température, capacité thermique, flux thermique...
- ✓ Electrique : courant, tension, charge, impédance, diélectrique
- ✓ Magnétique : champs magnétique, perméabilité, moment magnétique...
- ✓ Radiatif : lumière visible, rayon Micro-onde...

✓ (Bio) chimique : humidité, gaz, sucre, hormone...

II.2.3 Modes de fonctionnement des Capteurs :

On classifie les capteurs en deux grandes familles en fonction de la caractéristique électrique de la grandeur de sortie

a. Capteurs actifs :

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre au mesurande : énergie thermique, mécanique ou de rayonnement.

Les plus importants parmi ces effets sont regroupés tableau 2.1; dans la suite du paragraphe, on en donne une description sommaire destinée à éclairer leur mode d'application. [2]

Mesurande	Effet utilisé	Grandeur de sortie
Température	Thermoélectricité	Tension
Flux de rayonnement optique	Pyroélectricité Photoémission Effet photovoltaïque Effet photoélectromagnétique	Charge Courant Tension Tension
Force Pression Accélération	Piézoélectricité	Charge
Vitesse	Induction électromagnétique	Tension
Position (aimant)	Effet Hall	Tension

Tableau 2.1 : Capteurs actifs : principes physiques de base.

b. Capteurs passifs :

Le capteur se comporte en sortie comme un dipôle passif qui peut être résistif, capacitif ou inductif. Le tableau ci-dessous résume, en fonction de la mesurande, les effets utilisés pour réaliser la mesure.

[2]

Mesurande	Caractéristique électrique sensible	Types de matériaux Utilisés
Température	Résistivité	Métaux : platine, nickel, cuivre.
Très basse température	Constante diélectrique	Semi-conducteurs Verres
Flux de rayonnement Optique	Résistivité	Semi-conducteurs
Déformation	Résistivité Perméabilité magnétique	Alliages de nickel, silicium dopé. Alliages ferromagnétiques
Position (aimant)	Résistivité	Matériaux magnéto-résistants : bismuth, antimoniure d'indium.
Humidité	Résistivité Constante diélectrique	Chlorure de lithium. Alumine ; polymères
Niveau	Constante diélectrique	Liquides isolants.

Tableau 2.2 : Capteurs passifs : principes physiques et matériaux.

c. Capteurs intelligents :

On désigne par capteur intelligent l'ensemble de mesure d'une grandeur physique constitué de deux parties (figure 2.4):

- une chaîne de mesure pilotée par microprocesseur ;
- une interface de communication bidirectionnelle.

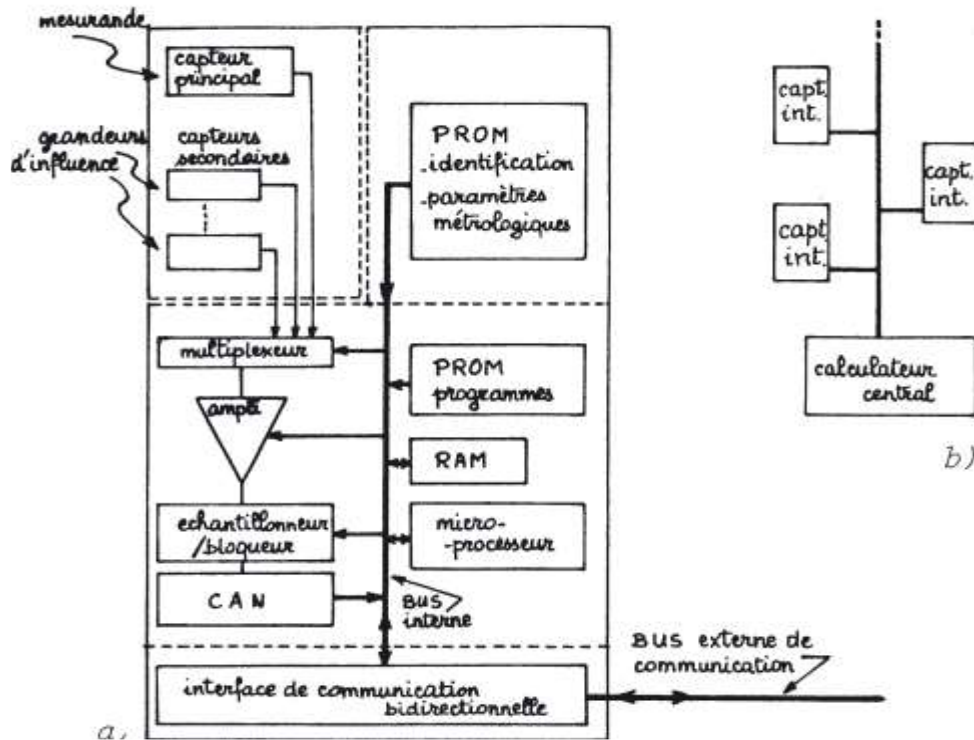


Figure 2.4 : Capteur intelligent

La chaîne de mesure comporte :

- le capteur principal spécifique du mesurande étudié, et identifiable par un code stocké en PROM (Programmable Read Only Memory: mémoire programmable à lecture seule) ;
- les capteurs secondaires propres aux grandeurs d'influence susceptibles d'affecter la réponse du capteur principal ;
- Les dispositifs classiques permettant l'obtention sous forme numérique de la grandeur de sortie de chaque capteur : conditionneur, multiplexeur, amplificateur, échantillonneur-bloqueur, convertisseur analogique-numérique ;
- un microprocesseur affecté aux tâches suivantes : gestion de l'acquisition, correction de l'effet des grandeurs d'influence au moyen des paramètres stockés en PROM et des données fournies par les capteurs secondaires, linéarisation, diagnostic des capteurs

II.2.4 Caractéristiques métrologiques du capteur :

On se propose, dans ce paragraphe de citer les principaux paramètres métrologiques qui constituent les liens effectifs entre un capteur et la grandeur qu'il mesure.

a. Etendue de mesure :

On va examiner la courbe d'entrée-sortie d'un capteur, qu'on appelle aussi courbe d'étalonnage. C'est une courbe qui exprime la relation d'évolution de la grandeur de sortie en

fonction de la grandeur d'entrée. Il s'agit d'une courbe en régime permanent qui ne donne pas d'informations sur les caractéristiques transitoires du capteur.

Sur cette courbe, on notera l'étendue de mesure. C'est la différence algébrique entre les valeurs extrêmes pouvant être prises par la grandeur à mesurer, pour laquelle les indications d'un capteur, obtenues à l'intérieur du domaine d'emploi en une seule mesure, ne doivent pas être entachées d'une erreur supérieure à celle maximale tolérée [2].

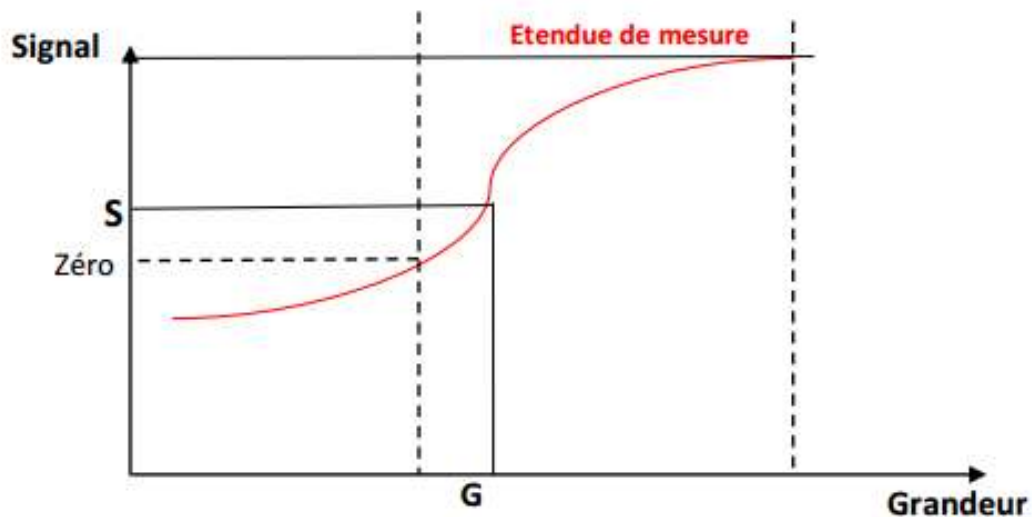


Figure 2.5 : Courbe d'étalonnage d'un capteur

On appelle portées les valeurs limites de la grandeur à mesurer correspondant à cette étendue de mesure ; et zéro la valeur de l'information de sortie du capteur correspondant à la portée minimale.

b. Sensibilité :

Elle détermine l'évolution de la grandeur de sortie en fonction de la grandeur d'entrée en un point donné. C'est la pente de la tangente à la courbe issue de caractéristique du capteur.

Dans le cas d'un capteur linéaire, la sensibilité du capteur est constante.

$$\text{Sensibilité} = (d(\text{Grandeur de sortie}))/d(\text{mesurande})$$

c. Finesse :

C'est la qualité d'un capteur à ne pas venir modifier par sa présence la grandeur à mesurer. Cela permet d'évaluer l'influence du capteur sur mesure. On la définit non seulement vis-à-vis du capteur mais aussi vis-à-vis de l'environnement d'utilisation du capteur. La finesse d'un capteur donnée ne peut donc être appréciée qu'en fonction de ses conditions effectives d'utilisation.

d. Linéarité :

C'est la zone dans laquelle la sensibilité du capteur est indépendante de la valeur mesurande. Cette zone peut être définie à partir de la définition d'une droite obtenue comme approchant au mieux la caractéristique réelle du capteur, par exemple par la méthode des moindres carrés.

On définit à partir de ce droit l'écart de linéarité qui exprime en % l'écart maximal entre la courbe réelle et la droite approchant la courbe.

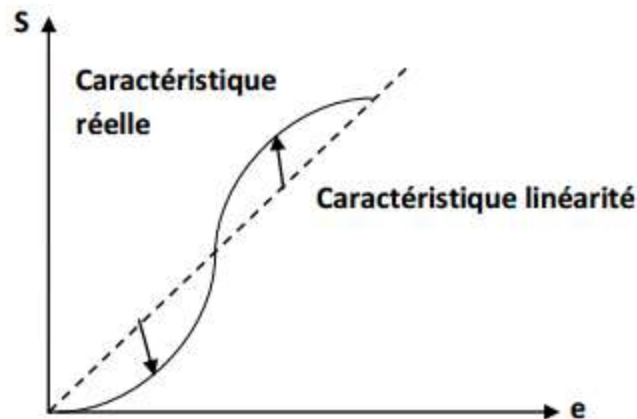


Figure 2.6 : Exemple de linéarisation de caractéristique.

II.2.5 Caractéristiques statiques d'un capteur :

Ces paramètres permettent de prendre en compte la notion d'erreurs accidentelles qui peuvent survenir sur un capteur.

On peut, pour cela rappeler les résultats suivants : soient n mesures effectuées sur un mesurande ; on définit à partir de ces n mesures :

- La valeur moyenne :

$$\langle m \rangle = \frac{\sum_i m_i}{n}$$

L'écart type (dispersion des résultats autour de la valeur moyenne) :

$$\sigma = \sqrt{\frac{\sum (m_i - \langle m \rangle)^2}{n - 1}}$$

a. Fidélité :

Elle définit la qualité d'un capteur à délivrer une mesure répétitive sans erreurs. L'erreur de fidélité correspond à l'écart type obtenu sur une série de mesure correspondant à une mesurande constante.

b. Justesse :

C'est l'aptitude d'un capteur à délivrer une réponse proche de la valeur vraie et ceci indépendamment de la notion de fidélité. Elle est liée à la valeur moyenne obtenue sur un grand nombre de mesure par rapport à la valeur réelle.

c. Précision :

Elle définit l'écart en % que l'on peut obtenir entre la valeur réelle et la valeur obtenue en

sortie du capteur. Ainsi un capteur précis aura à la fois une bonne fidélité et une bonne justesse.

d. Rapidité :

C'est la qualité d'un capteur à suivre les variations de la mesurande, on général est-elle représentée par temps de réponse (à x%) à un échelon de la mesurande

II.2.6 Choix d'un Capteur :

Les capteurs présentent deux parties distinctes. Une première partie ayant pour rôle la détection d'un événement et une deuxième partie servant à la conversion de cet événement en un signal analogique ou numérique. Pour bien choisir un capteur il faut citer quelque caractéristique:[3]

- Etendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : Temps de réaction du capteur. La rapidité est liée à la bande passante.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure

En fonction de ces paramètres on pourra effectuer un ou plusieurs choix pour un type de détection.

D'autres éléments viennent compléter la pertinence du choix du capteur comme :

- ses performances,
- son encombrement,
- sa fiabilité
- son prix

II.3 Les capteurs biomédicaux :

II.3.1 Introduction

Les capteurs biomédicaux sont des capteurs destinés à recueillir les signaux physiologiques générés par le corps humain. Ils créent un dialogue entre les organes du corps et la machine. Ils doivent obéir à un ensemble de critères qui sont :

- ✓ Très haute sensibilité.
- ✓ Supportant la stérilisation.
- ✓ Possibilité d'usage unique.
- ✓ Non invasivité.
- ✓ Bio- compatibilité.
- ✓ Résistance aux agressions du corps humain.

- ✓ Provocation d'un minimum de perturbation du signal physiologique mesuré.
- ✓ Fiabilité et stabilité de leurs caractéristiques.
- ✓ Haute sécurité.

II.3.2 Classement des capteurs biomédicaux :

Les capteurs biomédicaux peuvent être classés suivants plusieurs critères dans des catégories différentes. En fonction de leurs quantités de détection. Ils sont Classés par le principe de fonctionnement on trouve : Les capteurs comprennent des capteurs physiques, des capteurs chimiques et des capteurs biologiques. [4]

- a. *Capteurs physiques* : il se réfère au capteur fait en fonction de la nature physique et effet. Ce type de capteurs est principalement représenté par des capteurs tels que le métal capteurs de tension de résistance, capteurs piézorésistif semi-conducteurs, piézoélectriques photoélectriques, etc.
- b. *Capteurs chimiques* : il se réfère au capteur fait selon la nature chimique et l'effet. Ce type de capteurs utilise habituellement des pellicules sensibles aux ions pour transformer la non-électricité d'un composant chimique comme le contenu, la densité, etc. à la quantité électrique connexe, telle que diverses électrodes sensibles à l'ion, tubes sensibles à l'ion, capteurs d'humidité etc.
- c. *Capteurs biologiques* : ou biocapteurs il se réfère au capteur utilisant le matériel biologique actif comme un système de reconnaissance de molécules. Ce genre de capteurs utilise généralement l'enzyme pour catalyser certaines réactions biochimiques ou des examens du type et le contenu des substances organiques de grande molécule à travers une combinaison spécifique. Il s'agit d'un capteur nouvellement développé dans la seconde moitié du siècle, et des exemples comprennent des capteurs enzymatiques, des capteurs de micro-organismes, des capteurs d'immunité, des capteurs de tissus, des capteurs d'ADN etc

II.3.3 Technologie de mesure biomédicale :

Les signaux biomédicaux sont généralement faibles, aléatoires avec un fort bruit et des interférences, permettent un changement dynamique et présentent des différences individuelles significatives. Par conséquent, les technologies de mesure biomédicales sont plus complexes et rigides que la technologie de détection industrielle commune.

La mesure biomédicale est une technologie de guidage dans l'acquisition et traitement de l'information biomédicale est directement lié à la recherche de technologie de détection biomédicale, méthodes de mesure biomédicale, électronique et systèmes de mesure. Par conséquent, la recherche et le développement novateurs dans la mesure biomédicale ont un effet direct sur la

conception et l'application des capteurs et des instruments médicaux.

La technologie de mesure biomédicale implique la détection de signaux chimiques et biologiques à différents niveaux d'organismes. Par exemple, ECG, EEG, EMG sont des signaux physiologiques électriques; la pression artérielle, la température corporelle, le souffle, le débit sanguin sont des signaux physiologiques non électriques; le sang et l'urine sont des signaux chimiques ou biologiques; les enzymes, les protéines, les anticorps et les antigènes sont des signaux biologiques. De même, les systèmes de mesure biomédicale exigent une fiabilité et une sécurité particulières. [4]

Par ailleurs, on rencontre deux types de capteurs ; les capteurs invasifs et les capteurs non invasifs.

- a. *Les capteurs invasifs* sont implantés en partie ou en totalité à l'intérieur du corps humain, Ces capteurs délivrent des mesures de bonne qualité (très peu bruitées) parce qu'ils sont placés très près de la source des signaux qu'ils mesurent.
- b. *les capteurs non invasifs* à l'inverse des invasifs, sont placés à l'extérieur du corps humain. Cependant, les mesures qu'ils génèrent sont plus bruitées que celles des capteurs invasifs parce qu'ils sont plus éloignés que ces derniers de la source des signaux qu'ils mesurent. [5]

II.3.4 les différents capteurs médicaux

Pratiquement, il existe plusieurs capteur utiliser dans le domaine médicale on cite :

- Le capteur d'électrocardiogramme ECG
- Le capteur de pouls cardiaques et la saturation d'oxygène dans le sang (SpO2)
- Capteur de température
- Capteur de pression artérielle
- Capteur de position et les chute
- Capteur de glucimie
- Capteur de débit d'air
- Capteur de muscle / électromyographie

a. Le capteur d'électrocardiogramme ECG :

Le cœur est un organe possédant un automatisme permettant la contraction et le relâchement périodique. Certaines de ses cellules, regroupées dans le tissu nodal (ou nœud sinusal), sont responsables d'un influx électrique se propageant aux cellules cardiaques (myocardiques) avoisinantes. Suite à cela, le cœur se contractera et pourra alors éjecter le sang aux autres organes. Physiologiquement, l'influx électrique est dirigé vers le bas et vers la gauche. Tout le

fonctionnement de l'électrocardiogramme (ECG) repose sur la direction et la répétition de cet influx.

Le principe de l'ECG est d'enregistrer les impulsions électriques à l'origine des contractions cardiaques. Les impulsions électriques sont enregistrées à distance du cœur, à travers la peau, au moyen d'électrodes. L'électrocardiogramme (ECG ou ECG) est un outil de diagnostic couramment utilisé pour évaluer les fonctions électriques et musculaires du cœur. Il est devenu l'un des tests médicaux les plus couramment utilisés en médecine moderne.

Il existe deux « types » d'électrodes : les électrodes frontales au nombre de trois (ou quatre, afin d'affiner le signal) se plaçant sur les membres et les électrodes précordiales au nombre de six se positionnant sur le torse.

L'électrode la plus utilisée est celle en Ag/AgCl. Cette électrode de surface (Figure 2.6) est une simple plaque métallique reliée à la peau du patient par une pâte électrolytique (gel conducteur). Afin de réduire les artefacts du mouvement, une isolation est réalisée au niveau de la périphérie de l'électrode [6].



Figure 2.7 : (a) et (b) les électrodes de l'ECG à l'usage unique et (c) le jeu de fils d'électrodes ECG M1934A



Figure 2.8 : Les électrodes d'ECG utilisés.

Le principe d'une mesure électro-cardiographique consiste à capter, amplifier, mettre en forme et visualiser l'électrocardiogramme.

b. Le capteur de La saturation en oxygène et de pouls (SpO2) :

Ce type de capteur permet de mesurer la fréquence du pouls et la saturation en oxygène dans le sang. L'oxymétrie repose sur une lecture du taux d'oxygène par une simple émission lumineuse. L'oxymètre de pouls est utilisé quotidiennement par les médecins, infirmiers, urgentistes...

- **Capteur en sonde :**

Chaque mesure est assurée par l'intermédiaire d'un capteur, dans le cas de la photo-pléthysmographie, la conception de la mesure est élaborée selon la technique ou la méthode physique (absorbance ou réflectance). Les capteurs existent sous différentes formes et différentes tailles afin de s'adapter à toutes les situations. Le plus souvent, ils se présentent sous l'apparence de pinces à placer sur un doigt. Ce dernier est constitué d'une source lumineuse qui est assez puissante pour pénétrer plus d'un centimètre dans le tissu, d'un photo détecteur pour détecter la lumière transmise [7]. Ce type de capteur se présente sous plusieurs formes

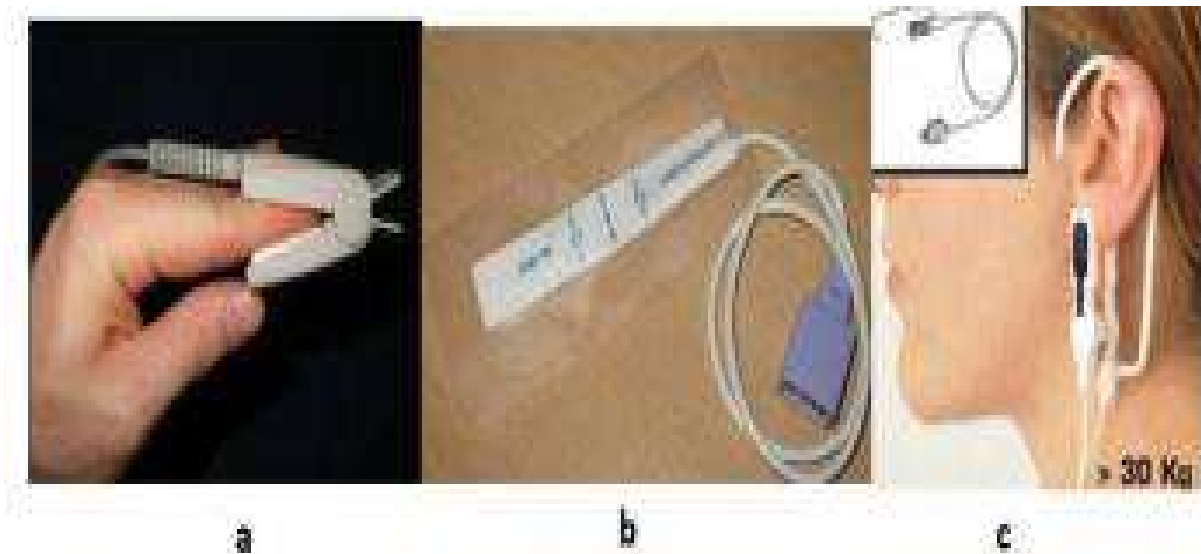


Figure 2.9 : les différentes formes du capteur du PPG, (a) pince semi-rigides s'adaptant à toutes les morphologies, (b) pince autocollante utilisée au niveau du pied, (c) pince à l'oreille.

- **Source lumineuse :**

La source lumineuse est constituée d'une diode émettrice de lumière infrarouge à 920 nm. Un photo-détecteur (phototransistor), placé à l'opposé de la diode, mesure l'intensité de la lumière transmise à travers le lit vasculaire. Les variations de l'intensité lumineuse transmises au photo-détecteur, dues aux variations de l'absorption de la lumière par les hémoglobines oxygénées (HbO₂) dans ce lit vasculaire permettent de déterminer la variation pulsée de l'oxygène dans le sang



Figure 2.10 : capteur de rythme cardiaque (pulse sensor)

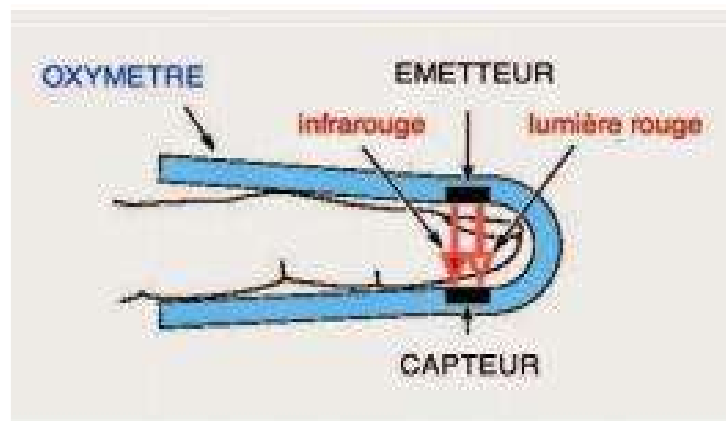


Figure 2.11 : principe de fonctionnement d'un capteur

c. Capteur de température :

La température corporelle dépend de la place dans le corps où la mesure est faite, et l'heure du jour et le niveau d'activité de la personne. Différentes parties du corps ont des températures différentes. La température moyenne du corps de base (prise interne) est de 37°C (98.6°F).

Chez les adultes sains, la température corporelle fluctue environ $0,5^{\circ}\text{C}$ ($0,9^{\circ}\text{F}$) tout au long de la journée, avec des températures plus basses le matin et des températures plus élevées en fin d'après-midi et en soirée, à mesure que les besoins et les activités du corps changent.

Il est d'une grande importance médicale pour mesurer la température corporelle. La raison en est qu'un certain nombre de maladies sont accompagnées par des changements caractéristiques de la température corporelle. De même, le cours de certaines maladies peut être surveillé en mesurant la température corporelle, et l'efficacité d'un traitement initié peut être évaluée par le physicien. [8]



Figure 2.12 : Le capteur de température

d. Capteur de pression artérielle :

La pression artérielle est la pression du sang dans les artères car il est pompé autour du corps par le cœur. Quand notre cœur bat, il contracte et pousse le sang à travers les artères pour le reste de notre corps. Cette force crée de la pression sur les artères. La pression artérielle est enregistrée comme deux nombres: la pression systolique (comme le battement de cœur) au-dessus de la pression diastolique (comme le cœur détend entre les battements).

La surveillance de la pression artérielle à la maison est importante pour beaucoup de personnes, particulièrement pour les patients ayant la pression artérielle élevée. La pression artérielle ne reste pas le même tout le temps. Il change pour répondre aux besoins de notre corps. Il est affecté par divers facteurs, y compris la position du corps, la respiration ou l'état émotionnel, l'exercice et le sommeil [9]



Figure 2.13 : Capteur de pression artérielle Particularités:

- Mesure automatique de la tension systolique, diastolique et impulsion avec Time &

Date

- Grand écran LCD avec rétro-éclairage LED
- Touche Pad

- 80 résultats de mesure avec Time & Date stockés dans l'appareil

e. Capteur de position et de chutes :

Le capteur de position du patient (accéléromètre) surveille cinq positions différentes des patients (debout/assis, couché, couché, gauche et droite.)

Dans de nombreux cas, il est nécessaire de surveiller les positions et les mouvements du corps en raison de leurs relations avec des maladies particulières (p. ex., l'apnée du sommeil et le syndrome des jambes agitées). L'analyse des mouvements pendant le sommeil aide également à déterminer la qualité du sommeil et les habitudes de sommeil irrégulières. Le capteur de position du corps pourrait également aider à détecter les évanouissements ou la chute des personnes âgées ou des personnes handicapées.

Le capteur de position du corps utilise un accéléromètre triple axe pour obtenir la position du patient.[9]



Figure 2.14 : Capteur de position et les chutes

Les positions de corps :

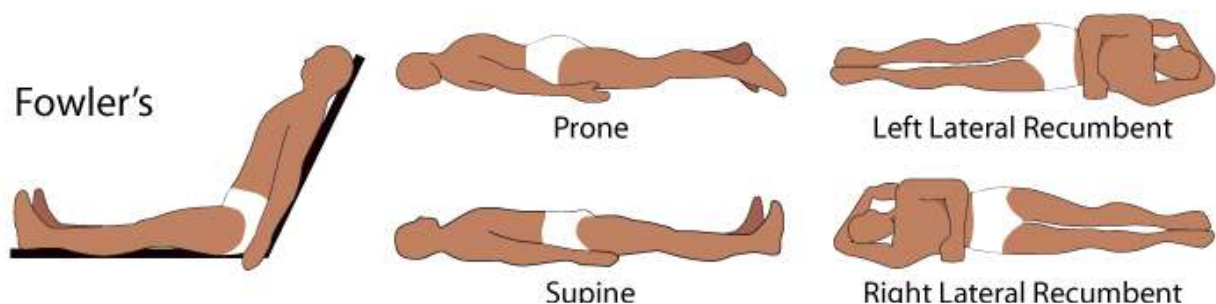


Figure 2.15 : les positions de corps

f. Capteur glycémie :

Un glucomètre ou lecteur de glycémie est un appareil permettant de mesurer rapidement le taux de glucose dans le sang (la glycémie), le plus souvent sur du sang capillaire. Ce type d'appareil est utilisé à l'hôpital par les soignants ou à domicile par les patients, notamment en cas de diabète sucré.[10]



Figure 2.16 :Capteur de glycémie

g. Capteur de débit d'air :

Les taux respiratoires anormaux et les changements dans la fréquence respiratoire sont un indicateur général d'instabilité physiologique majeure, et dans de nombreux cas, la fréquence respiratoire est l'un des premiers indicateurs de cette instabilité. Par conséquent, il est essentiel de surveiller la fréquence respiratoire en tant qu'indicateur de l'état du patient. Le capteur peut fournir un avertissement précoce de l'hypoxémie et de l'apnée.

Le capteur de débit d'air nasal / bouche est un appareil utilisé pour mesurer le rythme respiratoire chez un patient ayant besoin d'une aide respiratoire ou d'une personne. Ce dispositif consiste en un fil flexible qui se place derrière les oreilles et un ensemble de deux dents qui sont placées dans les narines. La respiration est mesurée par ces branches. [9]

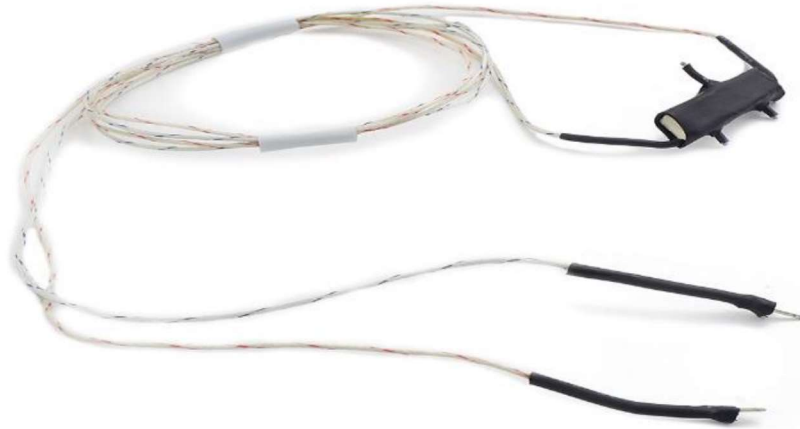


Figure 2.17 : Capteur de débit d'air

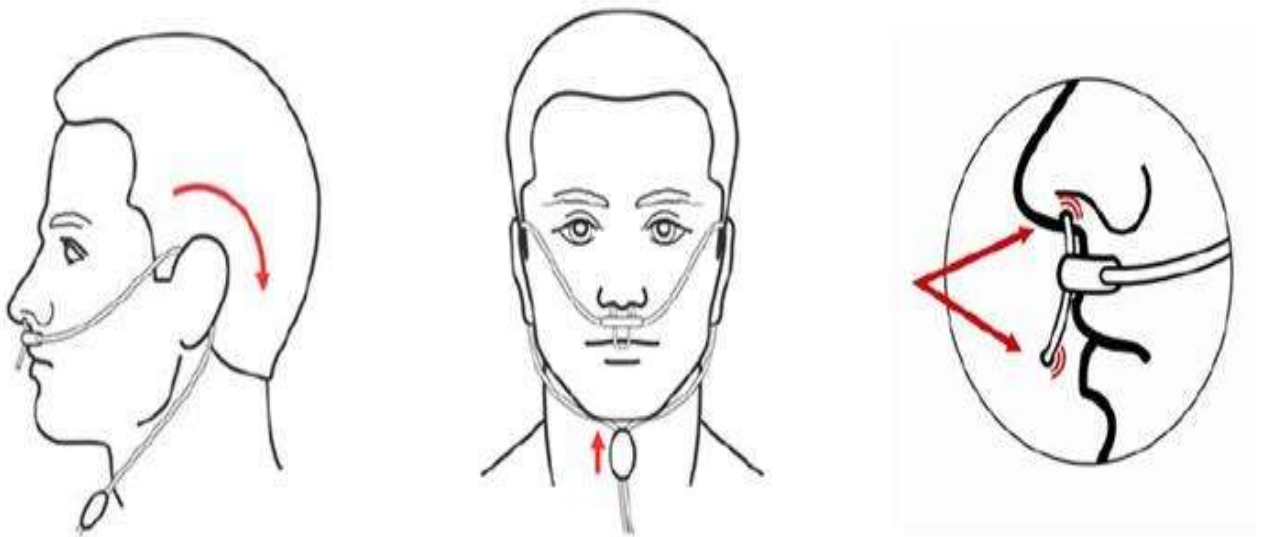


Figure 2.18 : Placez le capteur comme indiqué dans l'image

II.4 Conclusion

Ce chapitre été concentré sur l'étude des capteurs biomédicaux, où nous avons présenté quelques types de capteurs biomédicaux qui peuvent être exploités dans le domaine de télémédecine. Nous avons toutes fois présenté les caractéristiques et les propriétés de chaque type son mode fonctionnement sa fiabilité et ces performances.

CHAPITRE 3 :

Méthodologie de La conception

III.1 Introduction :

Le système de télésurveillance médicale (ou télémédecine) proposé est principalement conçu pour les personnes souffrant de maladies chroniques ou de personnes dites « fragiles ». Le terme « fragile », volontairement général, englobe des populations présentant des aptitudes physiques ou psychologiques dégradées par le vieillissement, des personnes présentant un handicap exigeant des moyens et une organisation adaptés à leurs besoins ou encore des sujets atteints d'une maladie neurodégénérative (maladie d'Alzheimer par exemple). Ou tout simplement toutes personnes nécessitant un suivi médical. Toute personne sachant qu'on les laisse mener leur vie quotidiennes de façon habituelle. Sans recourir à une présence à l'hôpital.

III.2 présentation du système :

Dans ce projet, nous avons développé une application électronique de l'internet des objets (internet of things IOT) ou les objets connectés. Elle est dédiée à la surveillance médicale à distance des personnes sans une présence à un hôpital. Le système est également conçu pour capturer des paramètres physiologiques pour des personnes éloignées et de les envoyer au Cloud afin d'effectuer un stockage permanent ou visualisées en temps réel en envoyant les données directement sur un ordinateur portable ou un Smartphone. Des capteurs biomédicaux sont utilisés pour capturer les caractéristiques d'entrée et le calculateur les accueille pour les envoyer par internet à un serveur en temps réel. Envoyer des alertes par SMS au praticien concerné. Comme indiqué sur la figure 3.1.

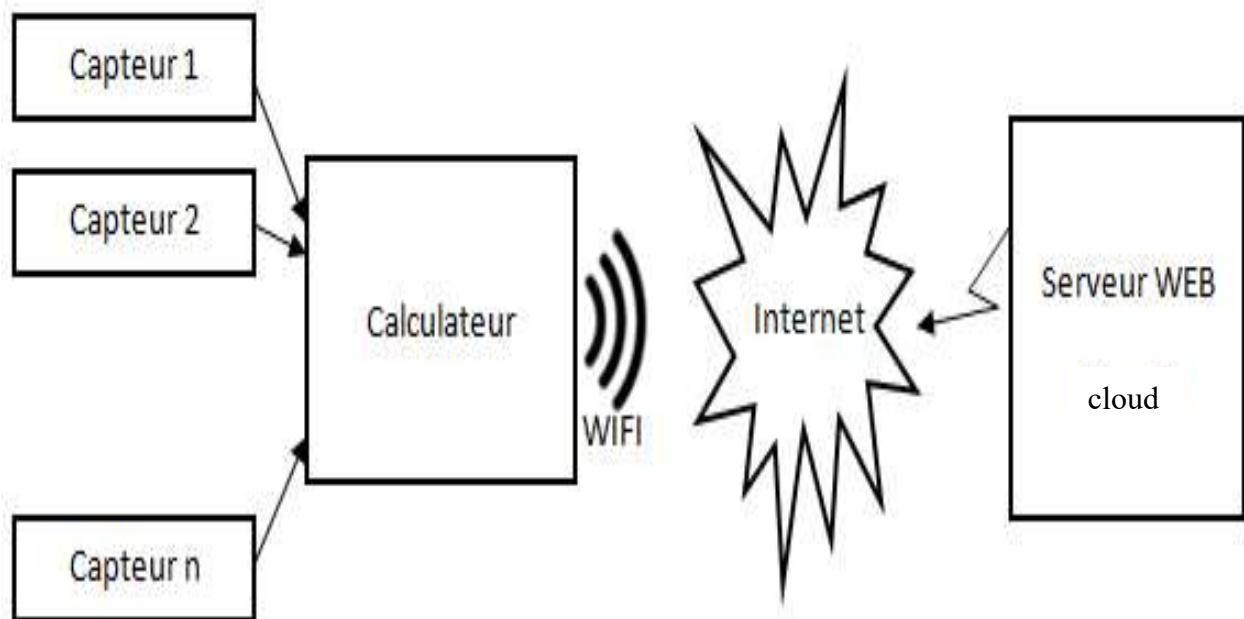


Figure 3.1 : schéma bloc du système à réaliser.

III.3 Architecture globale du système :

L'architecture générale de notre application est présentée sur la figure (3.2).

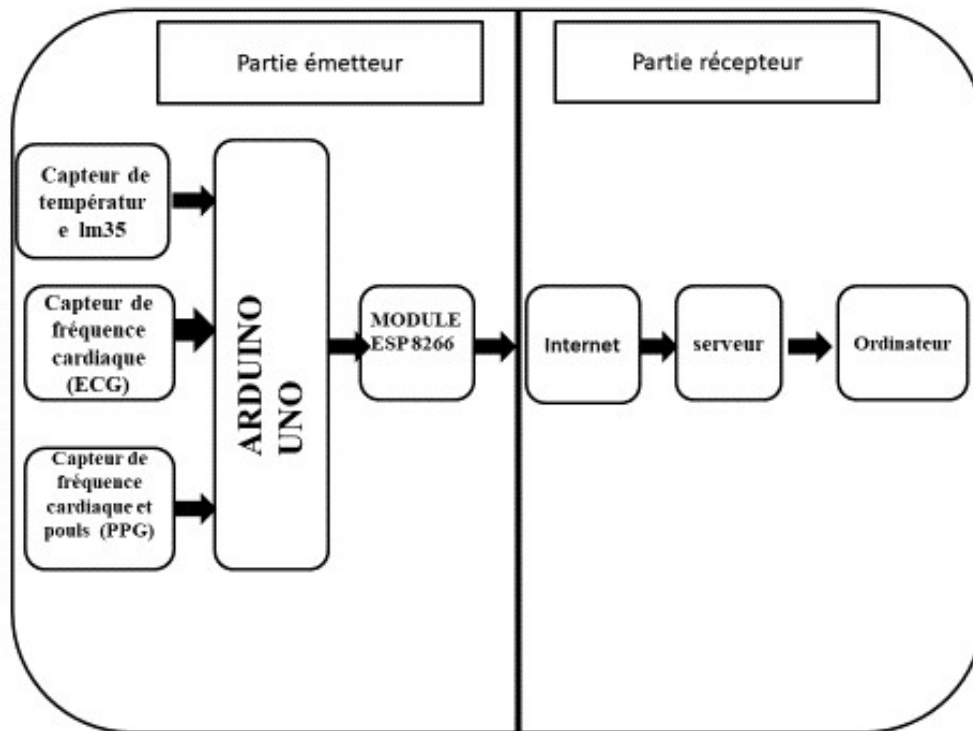


Figure 3.2 : Architecture globale du système.

La solution proposée permet d'expérimenter la surveillance à distance, à l'aide d'un ensemble de dispositifs IoT biomédicaux non intrusifs, des personnes et surtout ceux qui atteints des maladies. Elle assiste le corps humain en automatisant le traitement des informations issues de ces capteurs afin de détecter éventuellement et de signaler de manière précoce les situations à risques possible.

Le schéma ci-dessus montre les différentes parties décrivant l'application à réaliser. Chaque partie sera bien décortiquée dans les sections qui suivent. Pour faciliter la méthodologie de réalisation, nous avons décomposé notre travail en deux principales parties :

Partie émetteur (au niveau du patient) : cette partie est le principal pilier de notre application. Elle se présente comme un système embarqué comporte les deux aspects de la réalisation de l'électronique programmée : le soft et le hard

Partie récepteur (côté médecin) : dans cette partie nous trouvons un ordinateur reliée à internet qui représente le server local (on a utilisé XAMP) il héberge un système d'information capable d'acquérir les données, les traiter et enfin les présenter sur un terminal au médecin superviseur.

III.4 Partie émetteur (au niveau du Patient) :

Le module matériel du système de surveillance (Le bouclier), qui doit être porté par le patient, est conçu comme il est présenté sur le schéma bloc de la figure 3.3. Le système est contrôlé par une carte Arduino de type UNO. Nous rappelons qu'il existe plusieurs d'autres types de cartes Arduino. Les plus connues sont : la carte Due, Méga ... etc. Notre choix de cette carte est justifié par le fait que ces caractéristiques sont largement suffisantes à notre application (vitesse, nb de pin, espace mémoire disponible...) et elle est pratique, facilement programmable et donne la possibilité de brancher plusieurs composants électroniques grâce à ses connexions multiples (pins).

Quant à l'ensemble des capteurs physiologiques, on a été dans l'obligation de sélectionner un nombre limité de capteurs en raison du coût élevé et le manque de disponibilité. Ainsi que techniquement généralement pour chaque personne on suit un nombre limité de paramètres physiologiques. Notre choix été sur la mesure de température par capteur de température lm35, pour l'activité électrique du cœur ECG. On a utilisé une carte électronique qui est le Moniteur de Fréquence Cardiaque AD8232. Un capteur non invasif qui est Le PPG de Pouls et du taux d'oxygène dans le sang (SPO2).

Et enfin la connexion WIFI est d'assurer par l'utilisation d'un module WIFI l'ESP-8266.

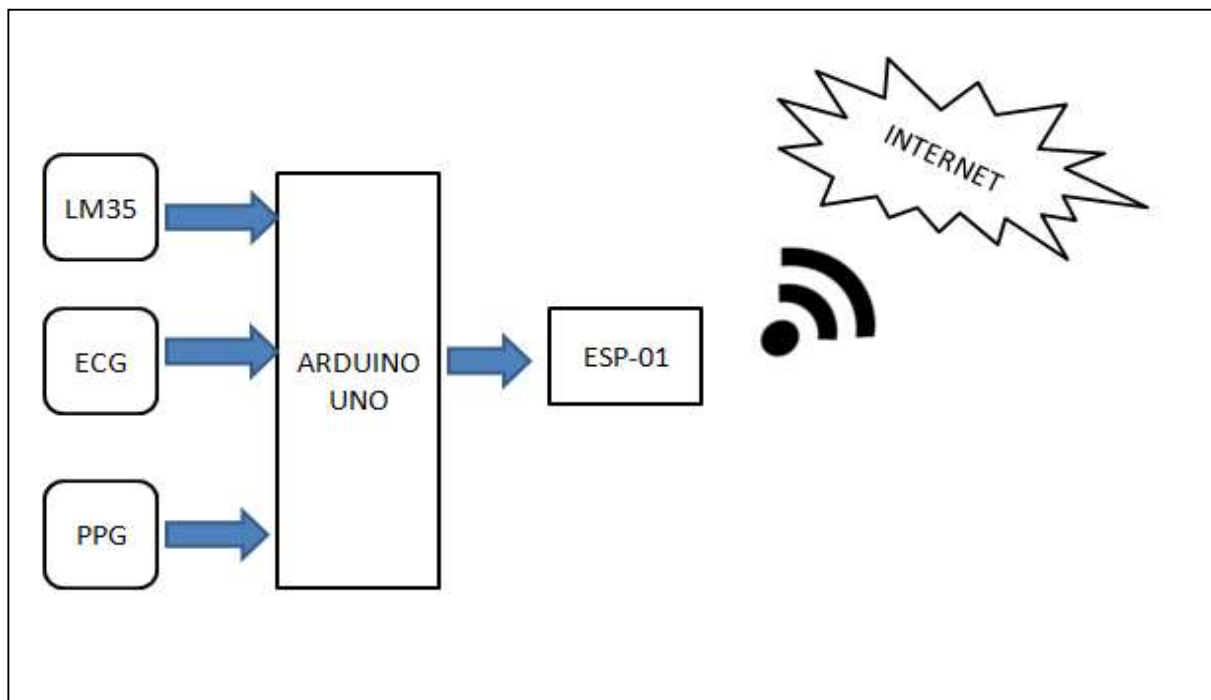


Figure 3.3 : Schéma bloc de la Partie émetteur (au niveau du patient).

III.4.1 Description du matériel utilisés :

III.4.1.1 La carte arduino uno :

Parmi ces types, nous avons choisi une carte Arduino UNO (carte Basique). L'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure si dessous.



Figure 3.4 : Arduino Uno.

III.4.1.1.1 Pourquoi arduino uno :

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- ✓ Le prix (réduits) : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 Dinars).
- ✓ Multi plateforme : Le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- ✓ Un environnement de programmation clair et simple : L'environnement

de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

- ✓ Logiciel Open Source et extensible : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (Fonctionne sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme à travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- ✓ Matériel Open source et extensible : Les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA328, les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût [1].

III.4.1.1.2 Technologie de la carte arduino uno :

La carte Arduino Uno est basée sur un Microcontrôleur ATmega328 cadencé à 16 MHz. C'est la plus récente et la plus économique carte à microcontrôleur [2]. Les caractéristiques techniques de la carte Arduino UNO sont présentées dans le tableau 3.1

Microcontroller	ATMega328
Tension de fonctionnement	5V
Tension d'alimentation – entrée - (Recommandée)	7-12V
Tension d'alimentation – entrée - (Limites)	6-20V
Broches E/S numériques	14 (Dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (Utilisables en broches E/S numériques)
Intensité maximum disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maximum disponible pour la sortie 3.3V	50 Ma
Intensité maximum disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader
Mémoire SRAM (Mémoire volatile)	2 KB (ATmega328)

Tableau 3.1 : Caractéristiques de la Carte Arduino UNO.

III.4.1.1.3 Microcontrôleur ATmega328 :

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit. Aujourd'hui, en soudant un grand nombre de composants encombrants; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C [3].

La figure (2.4) montre deux types de microcontrôleur ATmega328, qu'on trouve sur les cartes Arduino.

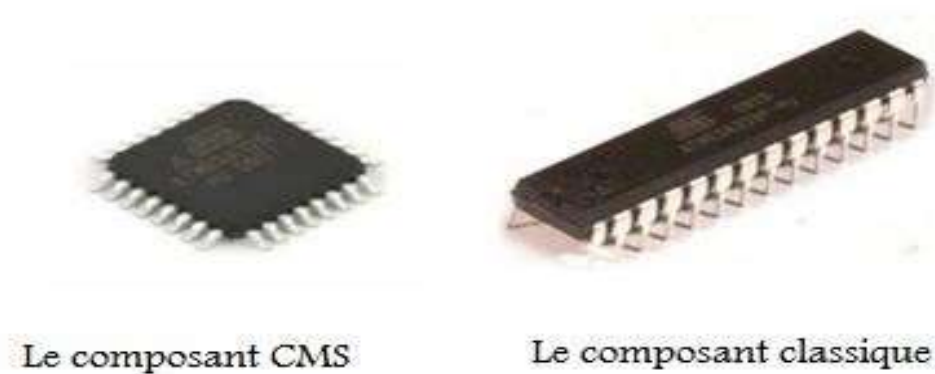


Figure 3.5: Microcontrôleur ATmega328.

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

- ✓ Mémoire Flash : C'est celle qui contiendra le programme à exécuter. Cette mémoire effaçable et réinscriptible est une mémoire programmée de 32Ko (dont boot loader de 0.5 ko).
- ✓ RAM : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- ✓ EEPROM : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme [4].

III.4.1.1.4 Sources de l'alimentation :

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte [5].

L'alimentation externe peut être soit un adaptateur secteur (Pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles. L'adaptateur secteur peut être connecté en branchant une prise (2.1mm) positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de (6 à 20 volts). Cependant, si la carte est alimentée avec moins de (7V), la broche (5V) pourrait fournir moins de (5V) et la carte pourrait être instable. Si on utilise plus de (12V), le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino Uno est entre (7V et 12V).

Les broches d'alimentation sont les suivantes :

- ✓ **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (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, 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 à 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 tout autre source d'alimentation régulée.
- ✓ **3.3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (Circuit Intégré Faisant l'Adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : Ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V).
- ✓ **GND** : Broche de masse (Ou 0V) [8].

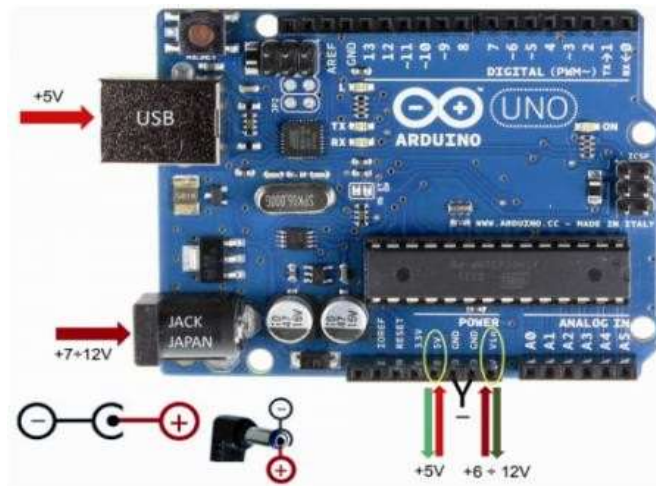


Figure 3.6: Sources de l'Alimentation de la carte Arduino UNO.

II.4.1.1.5 Entrées & sorties :

L'Arduino UNO possède 14 broches d'entrée/sortie digitale (Numérotées des 0 à 13), ces broches peuvent être utilisées soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` 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 de "Rappel au plus" (Pull-up) (Déconnectée par défaut) de (20-50 KOhms). Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH) [6].

De plus, certaines broches ont des fonctions spécialisées :

- ✓ **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.
- ✓ **Interruptions Externes** : Broches (2 et 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.
- ✓ **Impulsion PWM (largeur d'impulsion modulée)** : Broches (3, 5, 6, 9, 10, et 11). Fournissent une impulsion PWM8-bits à l'aide de l'instruction `analogWrite()`. Les applications de modulation de largeur d'impulsion (PWM) peuvent être trouvées dans le nombre

D'applications, par exemple les télécommunications, le contrôle des servomoteurs, la régulation de la tension, la remise en puissance, etc. mesurer la largeur de PWM à l'aide d'un microcontrôleur, De plus, comment les capteurs à ultrasons (Qui peuvent être utilisés pour la mesure à distance) peuvent fonctionner conjointement avec PWM.

- ✓ **SPI (Interface Série de Périphérique)** : Broches 10(SS) ,11(MOSI) ,12(MISO) ,13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI.
- ✓ **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire Interface - Interface "2 fils")
- ✓ **LED**: Broche 13. Il y a 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.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd 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 le 0V (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.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'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 appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [9]

III.4.1.1.6 Les ports de communications :

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL).

Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 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. Comme un port de

communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil:

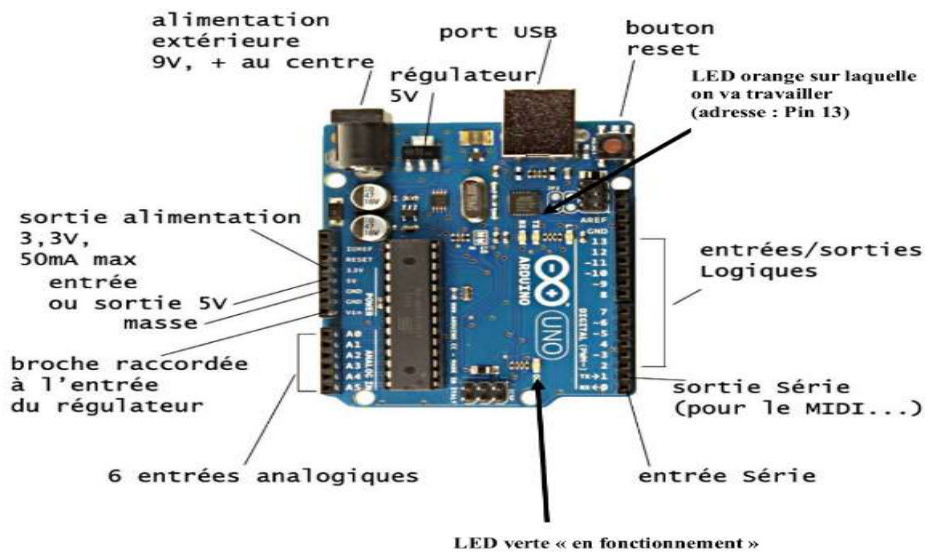


Figure 3.7 : Constitution de la carte Arduino UNO.

III.4.1.2 Le module Wifi ESP 8266 :

L'ESP8266 est un composant électronique permettant de rajouter la communication Wifi, avec une utilisation simple, d'apporter une connectivité wifi par ligne série à tout équipement notamment les cartes à base de microcontrôleur comme: Arduino. [7]

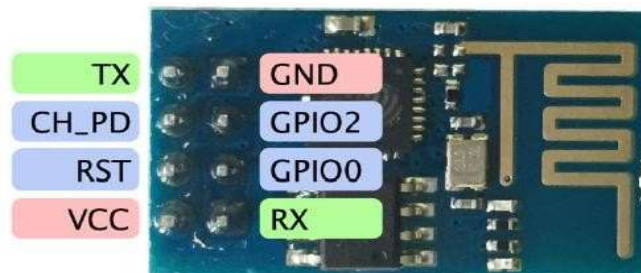


Figure 3.8: le module wifi ESP8266.

Dans notre travail, nous avons choisi l'élément esp 01

III.4.1.2.1 Principales caractéristiques d'ESP-01 :

- ✓ Norme sans fil: IEEE 802.11.
- ✓ Gamme de fréquence: 2.4 GHz ~ 2.5 GHz.
- ✓ Tension de fonctionnement: 3.3v.
- ✓ Courant de fonctionnement moyenne 80MA.
- ✓ Taille de paquet: 14.3 mm × 24.8 mm × 3 mm
- ✓ Sans fil mode réseau: station/softAP/SoftAP + station.

III.4.1.3.1 Avantage du capteur LM 35:

Parmi les avantages de ce capteur, mentionnons sa consommation très faible (de l'ordre de $60\mu\text{A}$), d'où une puissance dissipée également très faible, et sa linéarité qui demeure excellente sur toute sa plage de sensibilité. Cette plage va de -40°C à $+110^{\circ}\text{C}$ pour les LM35C et de 0°C à 100°C pour les LM35D. [9][10]

III.4.1.3.2 Caractéristiques du capteur LM 35:

- Proportionnelle à la température (en $^{\circ}\text{C}$).
- Ce capteur ne nécessite pas de calibrage externe.
- Précision de capteur (lm35) = 0.5°C .
- Sur une gamme de température de -55°C à $+150^{\circ}\text{C}$.
- Son coefficient est de $10\text{mV}/^{\circ}\text{C}$.
- Le capteur peut être alimenté entre 4V et 30V, il est alimenté en 05 V. [9]

[10]

III.4.1.3.3 Branchement du capteur LM35 :

- ✓ Vout doit être branché à une alimentation de 5V.
- ✓ GND doit être branché à la masse
- ✓ Tension de sortie Vs branche sur le pin analogique de l'Arduino

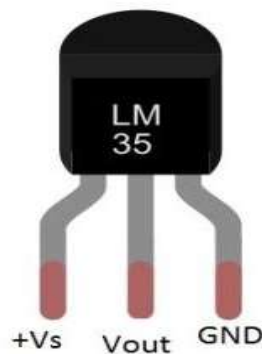


Figure 3.10 : Brochage du lm 35.

III.4.1.4 Le capteur du signal cardiaque AD8232 :

L'électrocardiogramme (ECG ou ECG) est un outil diagnostique couramment utilisé pour évaluer les fonctions électriques et musculaires du cœur. Nous avons utilisé Le Moniteur de Fréquence Cardiaque AD8232. Le moniteur de fréquence cardiaque AD8232 à fil unique agit comme un amplificateur opérationnel pour aider à obtenir facilement un signal clair des intervalles QT et PR. L'AD8232 est un bloc intégré de conditionnement de signal pour les

ECG et d'autres applications de mesure du biopotentiel. [11]

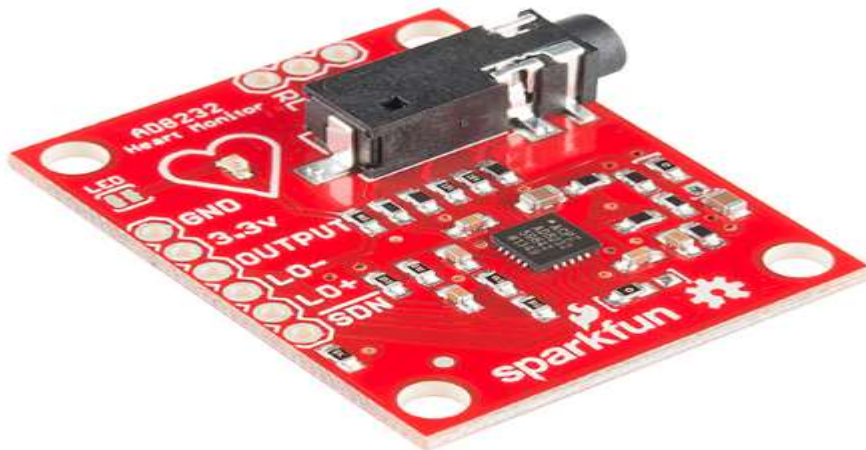


Figure 3. 11 : Le Moniteur de Fréquence Cardiaque AD8232.

Le moniteur de fréquence cardiaque AD8232 répartit neuf connexions du CI auxquelles vous pouvez souder des broches, des fils, ou d'autres connecteurs. SDN, LO+, LO-, OUTPUT, 3,3 V, GND procurent les broches essentielles au fonctionnement de ce moniteur avec un Arduino ou toute autre carte de développement. Des broches RA (bras droit), LA (bras gauche), et RL (jambe droite) sont également disponibles sur cette carte pour y fixer et utiliser vos propres capteurs personnalisés.

Board Label	Pin Function	Arduino Connection
GND	Ground	GND
3.3v	3.3v Power Supply	3.3v
OUTPUT	Output Signal	A3
LO-	Leads-off Detect -	11
LO+	Leads-off Detect +	10
SDN	Shutdown	Not used

Tableau 3.2 : connecter le capteur ECG avec Arduino.

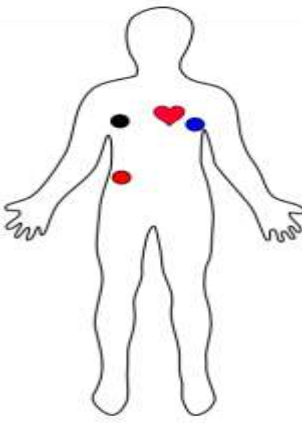
Cable color	Signal	
Black(noire)	RA(RightArm)	
Blue (blue)	LA (Left Arm)	
Red(rouge)	RL (Right Leg)	

Tableau 3.3 : les couleurs de câble et position

III.4.1.4.1 Les électrodes

Le signal est représenté par l'apparition d'une grandeur physique mesurable (courant, tension, Température, pression, etc.) en fonction du temps, et est issu d'une mesure délivré par un capteur.

Les électrodes sont le capteur du signal ECG., Elles sont les points de connexion entre l'appareillage et le patient. Elles sont des pièces de métal reliées par des fils conducteurs, de Différentes formes ; généralement rectangulaires pour l'application sur la surface du corps selon la Figure 3.12



Figure 3.12 : les électrodes, ECG

III.4.1.5 Le Capteur de pouls et la saturation en oxygène dans le sang (SPO2):

Le SPO2 est un capteur non invasif utilisé pour surveiller le taux de saturation en oxygène. Il se base sur une technique optoélectronique qui mesure les changements d'intensité de lumière lors du passage du sang. Le capteur optoélectronique est composé d'une source de lumière (LED) et d'une photodiode. La source de lumière éclaire une zone spécifique et les variations de sang produisent ainsi un changement dans l'intensité de lumière mesurée avec la photodiode. Ce capteur est souvent utilisé dans des endroits diverses

comme le front, le doigt ou l'oreille car la vascularisation dans ces endroits est très importante. [12]

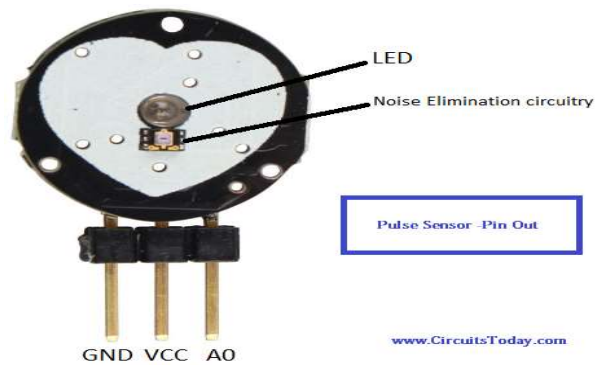


Figure 3.13 : Le Capteur (spo2).

Board Label	Pin Function	Carte Arduino
GND	Ground	GND
VCC	3.3v Power Supply	3.3v
OUTPUT	Output Signal	A1

Tableau 3.4 : connexion du capteur spo2 avec Arduino.



Figure 3.14 : la position d'un spo2.

III.4.2 Réalisation pratique du système :

Maintenant que nous avons tout présenté sur le système, et ces différentes composantes, il est grand temps de le mettre à l'œuvre. Pour cela, nous allons réaliser notre montage. Comme il a été mentionné au par avant le but de notre application sera de mesurer un certain nombre de paramètre physiologique (Dans notre cas, nous avons utilisé trois capteurs : température, ECG, SPO2) et d'envoyer cette mesure au could via un réseau d'internet WIFI. L'utilisation de l'ensemble des capteurs nécessite des bibliothèques de haut niveau pour une gestion facile.

Pour commencer notre réalisation, nous avons câblés et testé chaque module appart comme suit.

III.4.2.1 Montage et programmation du module WIFI

Première étape avant de pouvoir programmer l'ESP8266 : on doit réaliser le montage d qui va permettre de vérifier le bon fonctionnement d'un programme d'émission.

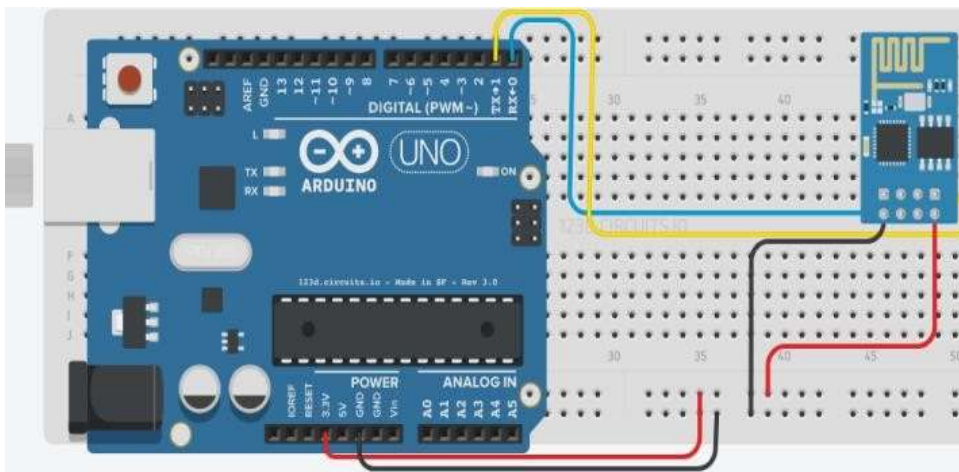


Figure 3.15: vue de prototypage du Montage d'esp8266 avec L'Arduino

Connecter le montage à l'ordinateur (via le convertisseur USB <=> série). Le témoin de l'alimentation de l'ESP8266 doit s'allumer.

Pour adapter le code de module Wifi avec l'Arduino il faut utiliser la bibliothèque SoftwareSerial.h: `#include <SoftwareSerial.h>`

Pour mettre en œuvre cette bibliothèque. Il faut tout d'abord falloir l'inclure dans le projet. Au choix, soit en cliquant sur «Library/Import/SoftwareSerial» dans l'IDE Arduino (inclus de base) ou alors en ajoutant la ligne suivante en haut du programme:

Une fois que la bibliothèque est là, alors il nous faudra l'utiliser. Cette bibliothèque nous donne l'accès à un objet nommé SoftwareSerial. Nous pouvons donc créer une instance

de cet objet en écrivant :

Software Serial esp8266 (Rx, Tx);

Rx et **Tx** seront les numéros des broches sur lesquelles sont connectées les broches d'émission/réception de notre module wifi. Dans notre cas la broche **Tx** du module branché sur le pin 10 de l'Arduino et la broche **Rx** reliée à pin 12. On a donc dans le programme sur la page suivant:

Pour plus de détail de la programmation de langage commande en c, Les fonctions de la bibliothèque SoftwareSerial esp8266 utilisée sont les suivant :

- **esp8266.begin (speed);** // démarre la voie série à la vitesse speed
- **esp8266.available ();** // retourne le nombre de caractère à lire
- **esp8266.read ();** // retourne le prochain caractère reçu
- **esp8266.print (val);** // envoie le char "val" sur la voie série

Notre module fonctionnant par défaut en 115200 bauds, nous allons pouvoir le préparer en écrivant :

```
#include<SoftwareSerial.h>
int led=8;
SoftwareSerial esp8266(10,11);

void setup()
{
  Serial.begin(9600);
  esp8266.begin(115200);
  pinMode(led, OUTPUT);
}
```

III.4.2.1.1 Configuration d'ESP 8266 avec l'environnement Arduino

D'abord il faut ajouter dans l'environnement Arduino/Génuine la bibliothèque permettant de compiler pour l'ESP8266, ensuite en suivant ces étapes :

« Ajouter l'url http://arduino.esp8266.com/package_esp8266com_index.json » à la liste de dépôts : “fichier” > “préférences” ; champ “AdditionnalBoard Manager URLs”

- ✓ Aller dans la sélection des cartes :
(menu“outils”>typedecartesxxxx>“Boardsmanager” (audessus de la liste des cartes)

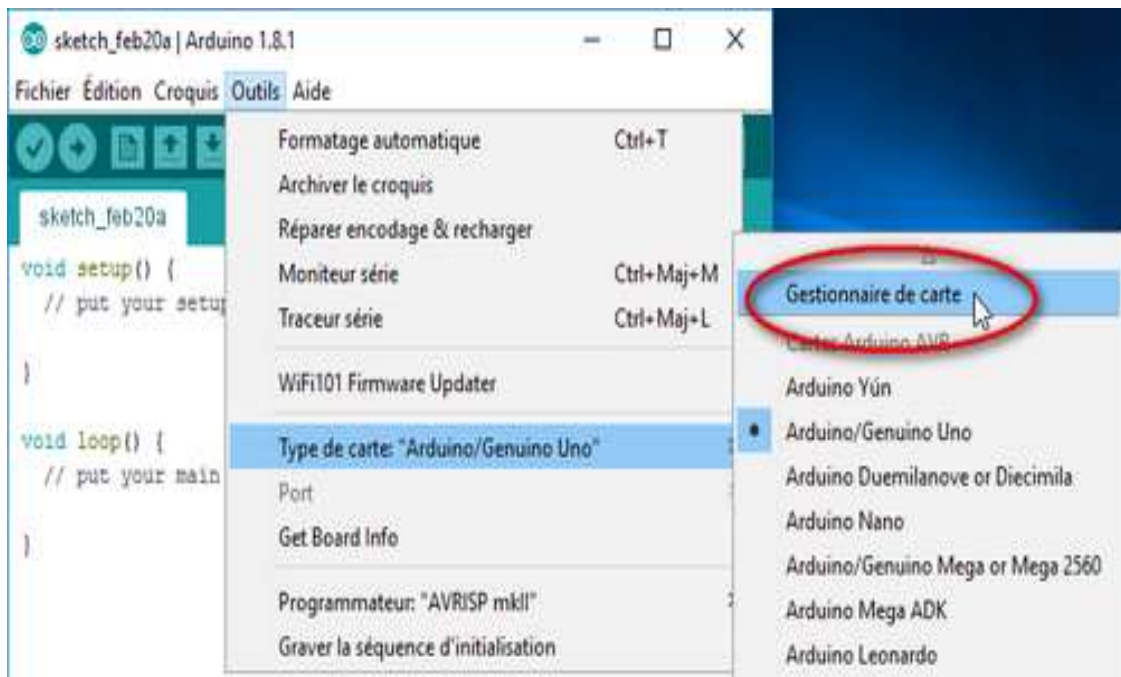


Figure 3.16: Configuration d'ESP 8266 avec l'environnement Arduino

- ✓ Tapper “esp8266” dans le champ de recherche
- ✓ Sélectionner le module “**esp8266 by ESP8266 Community**” et cliquez sur le bouton Install

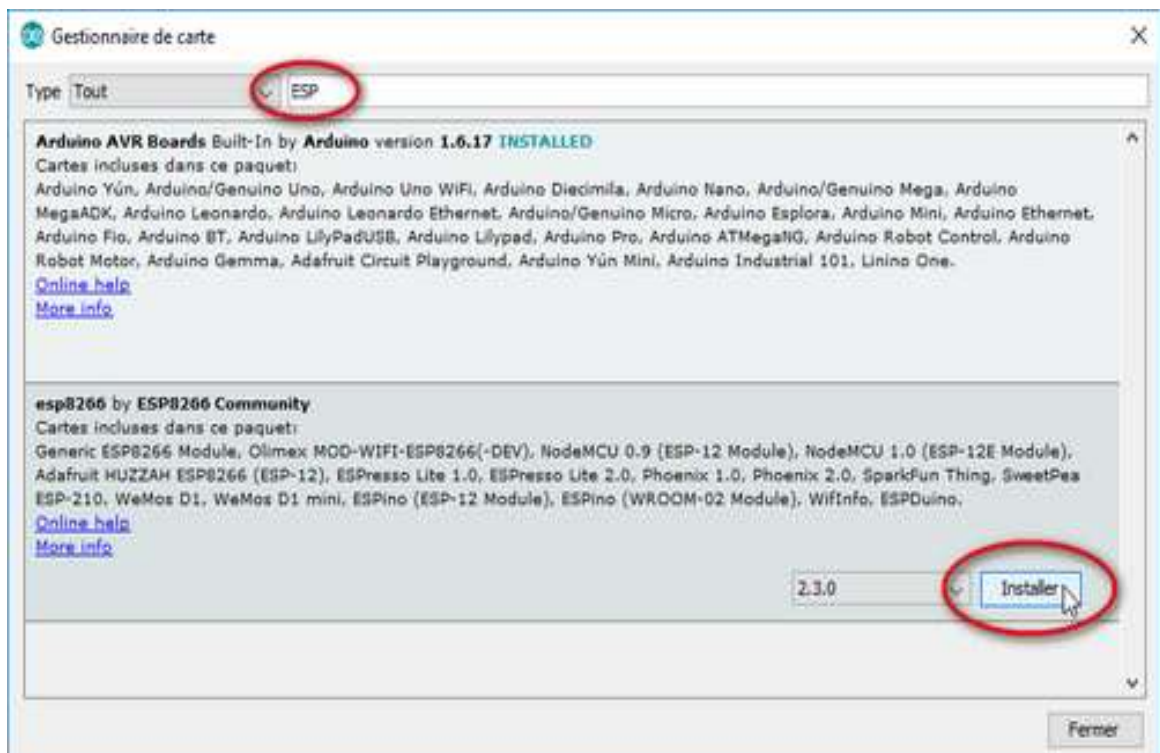


Figure 3.17: recherche les bibliothèques de module esp

- ✓ Fermer la fenêtre

Après on va entrer les commandes AT pour bien configuré notre module : les commandes AT constituent un langage de commandes développé à l'origine pour le modem Hayes Smart

modem 300. Ce jeu de commandes s'est ensuite retrouvé dans tous les modems produits. L'ESP8266 est un SoC (System on Chip) qui contient un microcontrôleur et un système WiFi. Il est pourvu d'un port RS232 qu'on peut utiliser pour lui envoyer des commandes AT. Afficher l'adresse IP, sélectionner le module comme une station, préciser le cout.

Les commands	Les Reponses	Functions
AT+CWMODE=?	+CWMODE:(1-3) OK	Choisir le mode de l'opération
AT+CWJAP=ssid,pwd	OK	Création le nom de wifi et le mot de passé
AT+CIPSTA?	+CIPSTA:ipOK	Sélectionner Adresse IP d'ESP
AT+CIPSEND=length	SEND OK	Préciser la taille de data qu'il va envoyer par ESP
AT+CIPCLOSE=id	OK	Quitter le TCP
AT+CIFSR	OK	Présenter l'adresse IP
AT+GMR	<pre>wdt reset load 0x40100000, len 2 4444, room 16 tail 12 chksum 0xe0 ho 0 tail 12 room 4 load 0x3ffe8000, len 3 168, room 12 tail 4 chksum0x93 load 0x3ffe8c60, len 4 956, room4 tail 8 chksum 0xbd csum 0xbd ready</pre>	Afficher la version de module
AT+RST	OK	Restarter le module

Tableau3.1: les AT commandent de module WIFI ESP8266

- Vous pouvez ensuite écrire vos commandes dans le moniteur série:
- Si vous écrivez "AT", l'ESP8266 devrait répondre "OK".
- La commande "AT+RST" cause un redémarrage de l'ESP8266.
- "AT+GMR" permet de connaître le numéro du firmware de l'ESP8266, ce qui peut être bon à savoir.
- La commande "CWMODE" permet de définir l'ESP8266 comme une station (STA), un point d'accès (AP) ou les deux, selon qu'on lui assigne les valeurs 1, 2 ou 3. Alors écrivons la commande "AT+CWMODE=2". L'ESP8266

répondra "OK" s'il a fait le changement, ou "no change" si l'ESP8266 était déjà dans l'état demandé.

- Puis, on demande au ESP8266 de joindre le réseau WiFi en écrivant :
AT+CWJAP="SSID","PSK"
- En suite on 'écrit la commande "AT+CIFSR", l'ESP8266 afin d'avoir l'adresse IP qu'on lui a assigné :

```
sendData("AT+RST\r\n", 2000, DEBUG); // restart du module
sendData("AT+CWMODE=2\r\n", 1000, DEBUG); // configurer comme une station
sendData("AT+CIFSR\r\n", 1000, DEBUG); // présenter le IP adress de ESP
sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); // configurer pour multiples connections
sendData("AT+CIPSERVER=1,80\r\n", 1000, DEBUG); // préciser le port en 80
```

Figure 3.8 : les commandes AT de module WIFI ESP8266 en environnement Arduino

III.4.2.2 Montage et programmation du capteur de température Lm 35:

Le capteur de température utilisé est un circuit intégré spécialisé LM 35. Ce capteur utilisé appartient à la classe des capteurs actifs, il est alimenté par une source de tension. Lorsqu'il est alimenté sous une tension de 5 V, la tension de sortie est proportionnelle à la température du milieu dans lequel est plongé ce capteur.

Pour réaliser ce montage, il va nous falloir :

- Un capteur LM35
- Une carte Arduino UNO
- Une plaque d'essai et des fils pour câbler notre montage.

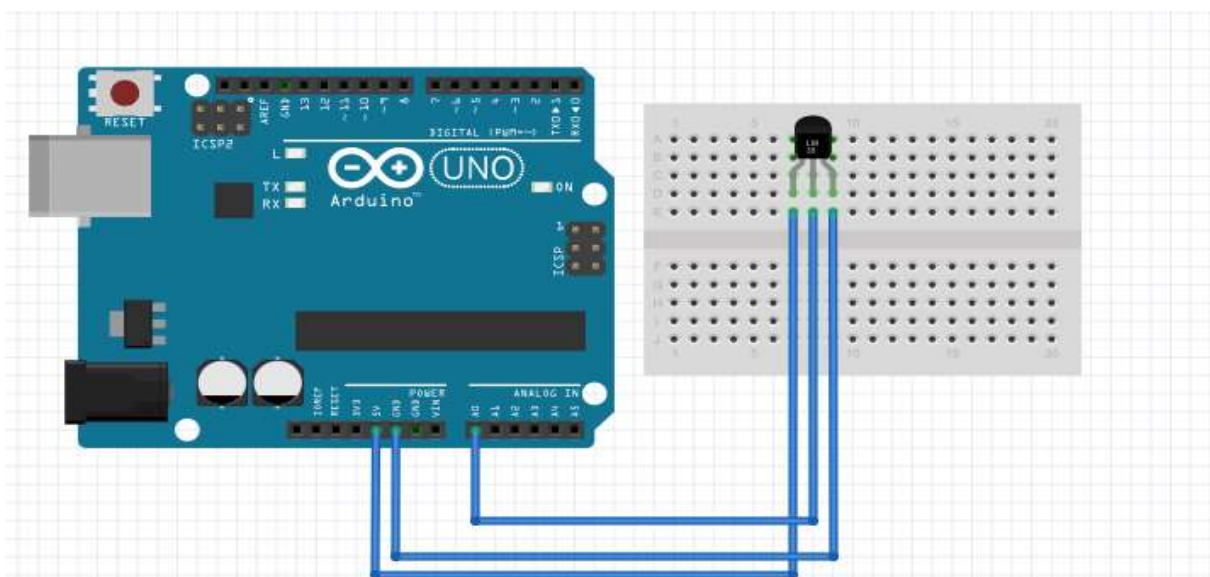


Figure 3.18: Vue prototypage du montage LM35 avec l'Arduino UNO

Pour commencer notre montage, nous allons câbler la broche VCC du capteur à

l'alimentation 5V de la carte Arduino au moyen d'un fil. On fait ensuite de même avec la broche GND du capteur qui vient se câbler sur la broche GND de la carte Arduino. On achève ensuite le circuit en reliant la sortie du capteur à la broche A0 de la carte Arduino avec un fil.

Le programme doit établir une connexion, lire les données reçues, s'il reçoit un "LM35.température" il lira la valeur analogique du détecteur de température (LM35) et transformera cette valeur à son correspondant en température (°C), puis envoyer cette température au module WIFI pour l'affichage via la page de réception sur le serveur. Ci-dessous le programme dans l'Arduino :

```
void loop() {
  String chk ;

  // Mesure la tension sur la broche A0
  int valeur_brute = analogRead(A0);

  // Transforme la mesure (nombre entier) en température via un produit en croix
  float temperature_celcius = valeur_brute * (5.0 / 1023.0 * 100.0);

  delay (2000);
```

Figure 3.19 : le code de LM 35

Dans la fonction loop (), nous allons faire trois choses :

1. Mesurer la tension sur la broche A0 avec analogRead ().
2. Transformer le résultat de la mesure en un nombre à virgule (type float) en faisant un simple produit en croix. Rappel : 5V = 5000mV = 1023 en sortie de analogRead (), 10mV = 1°C, par conséquent, $température = valeurmesurée * (5.0 / 1023.0 * 100.0)$
3. Envoyer la valeur au server et attendre quelques millisecondes

Après avoir envoyé le programme dans la carte Arduino, en ouvrant le moniteur série (onglet outils), puis en sélectionnant la bonne vitesse de communication (ici 9600 bauds), vous devriez voir apparaître en temps réel la température en sortie du capteur.

Si le montage est correct, en pinçant le capteur ou en soufflant dessus, les valeurs dans le moniteur série doivent normalement changer.

III.4.2.3 Montage et programmation PPG pour Arduino :

Le PPG vise à mesurer la saturation en oxygène du sang et le pouls. Un émetteur constitué d'une diode rouge et infrarouge envoie deux rayons à travers le doigt. Le récepteur capte le rayon dont une partie a été absorbée selon la couleur du sang, qui varie en fonction de son oxygénation (seules certaines longueurs d'ondes sont absorbées). En exploitant les variations de couleur du sang, on en déduit son taux d'oxygénation, ce qui permet de mettre en évidence les différentes contractions du cœur, donc le rythme cardiaque.

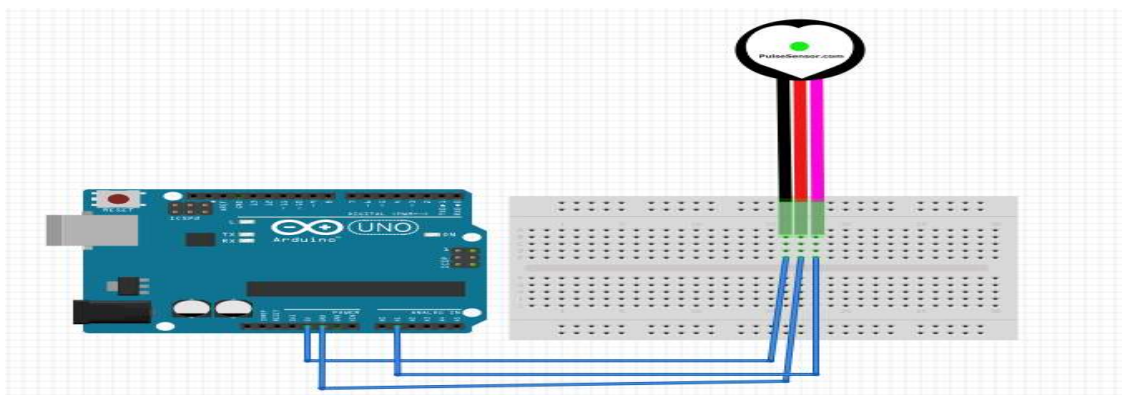


Figure 3.20 : Vue prototypage du montage de PPG avec Arduino Uno

Maintenant que nous avons notre montage, passons au code !

Le but de notre code va être de :

4. Lire la tension sur la broche A1
5. Convertir la valeur mesurée en une Fréquence (pour l'affichage)
6. Envoyer la valeur au server et attendre quelques millisecondes

```
String pulse_value() {
  String pulse_data = "";
  delay(50);
  while(pulse_data.length() < 20){
    Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's value.
    delay(50);
    if (Signal<600){
      pulse_data +=( String(Signal) + '|' ) ;
    }
  }
  delay(50);
  return pulse_data;
}
```

Figure 3.21 : le code PPG

III.4.2.4 Montage et programmation ECG pour Arduino :

L'électrocardiogramme (ECG) est un signal qui représente l'activité électrique du cœur. L'ECG est un élément essentiel que ce soit dans la surveillance des patients ou dans le diagnostic des maladies cardiovasculaires.

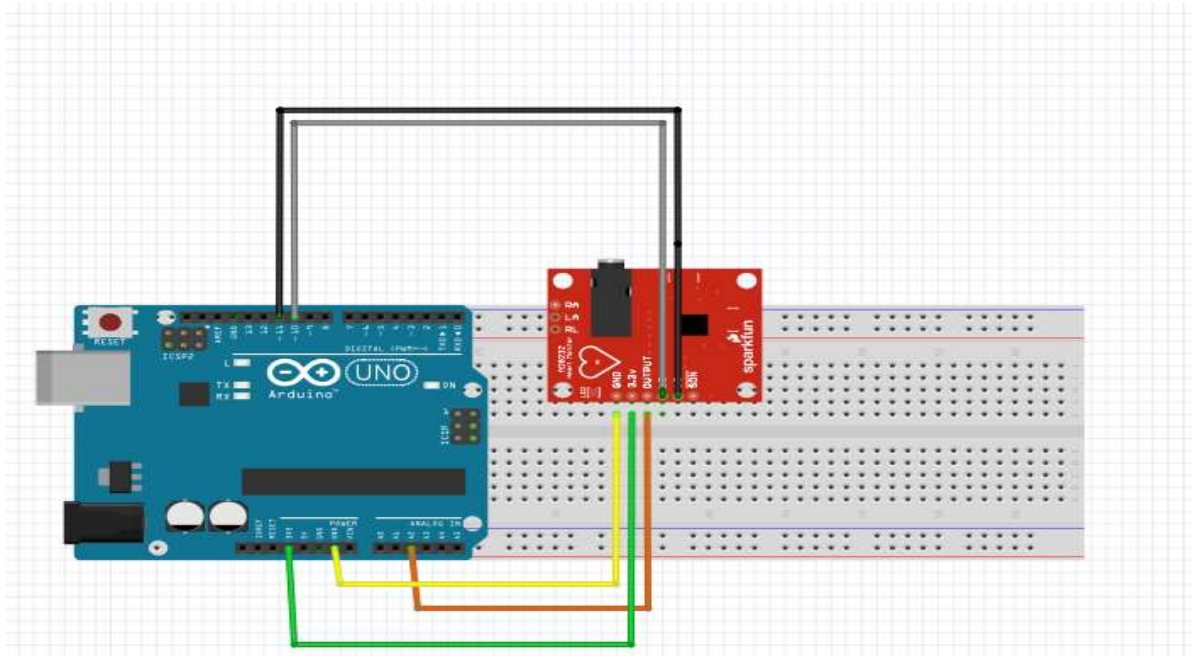


Figure 3.22 : Branchement ECG avec Arduino Uno

Le code de démonstration capteur ECG :

Maintenant que nous avons notre montage, Le but de notre code va être de :

1. Lire la tension sur la broche A2
2. Convertir la valeur mesurée en une Fréquence (pour l'affichage)
3. Envoyer la valeur au server et attendre quelques millisecondes

```
String ecg_value() {
    String pulse_data = "";

    delay(50);
    while(pulse_data.length() < 20) {
        Signal = analogRead(ECGPin); // Read the PulseSensor's value.
        delay(50);

        //if (Signal<600) {
            pulse_data +=( String(Signal) + '|' );
        // }
    }
}
```

Figure 3.23 : le code d'ECG.

III.4.3 Montage globale :

Toute action réalisée par la partie client ne peut être exécutée qu'avec la permission d'un système de commande intégré dans la carte "Arduino".

Une fois que tous les composants électroniques (LM35, ECG, PPG et EPS) sont bien connectés à la carte Arduino, un programme informatique est téléchargé obligatoirement pour assurer la bonne communication du système. Via un câble USB reliant le Pc à la carte Arduino.

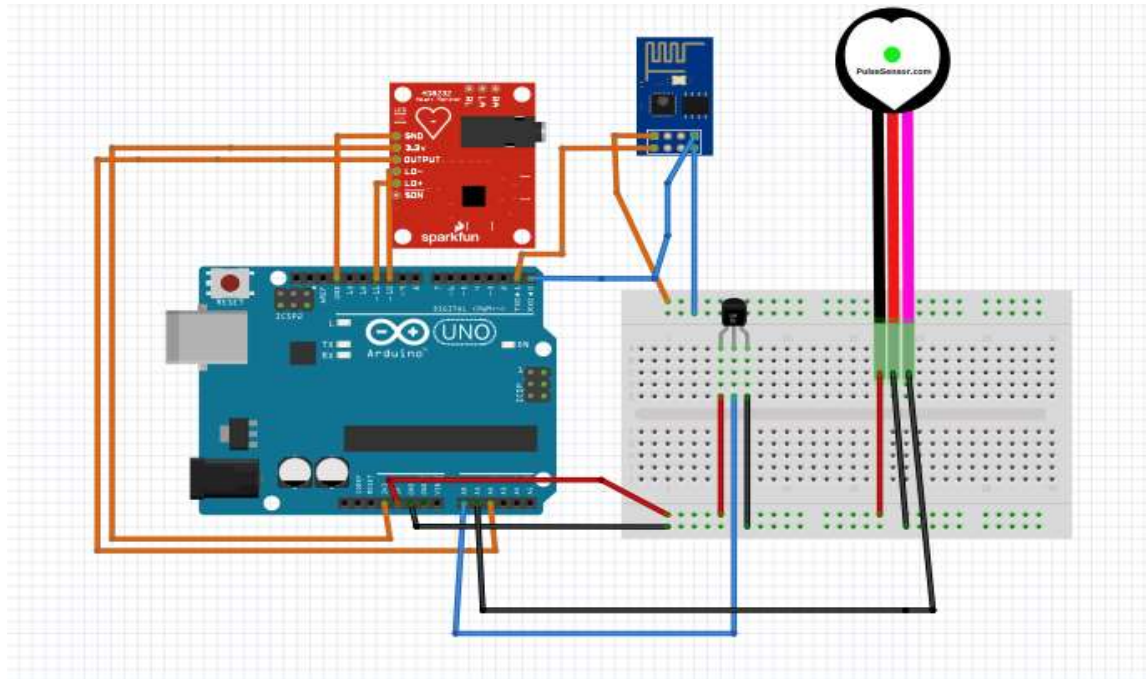


Figure 3.24 : montage prototypage

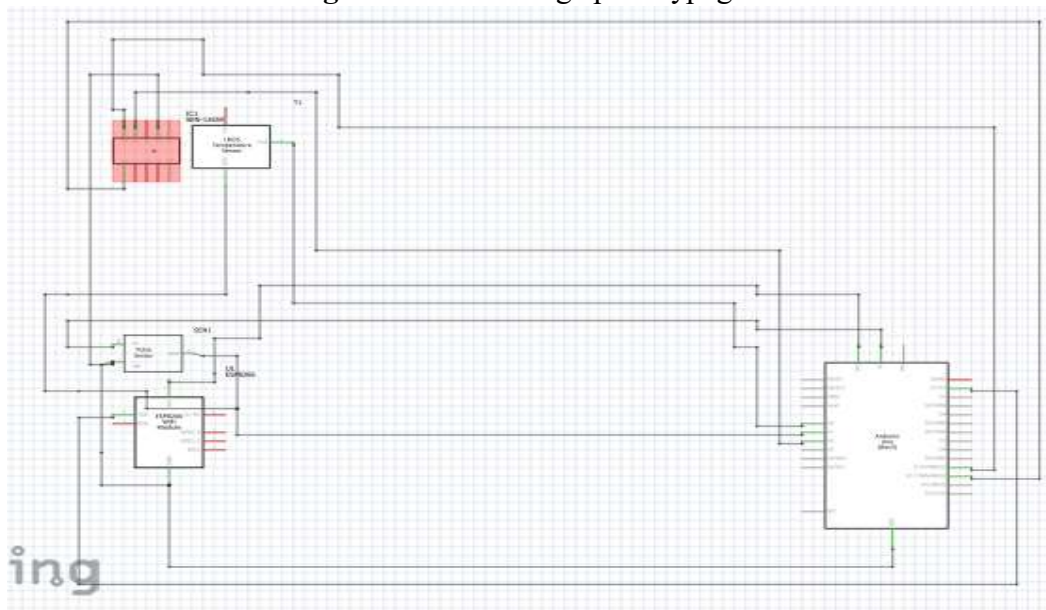


Figure 3.25 : Schéma électronique du montage

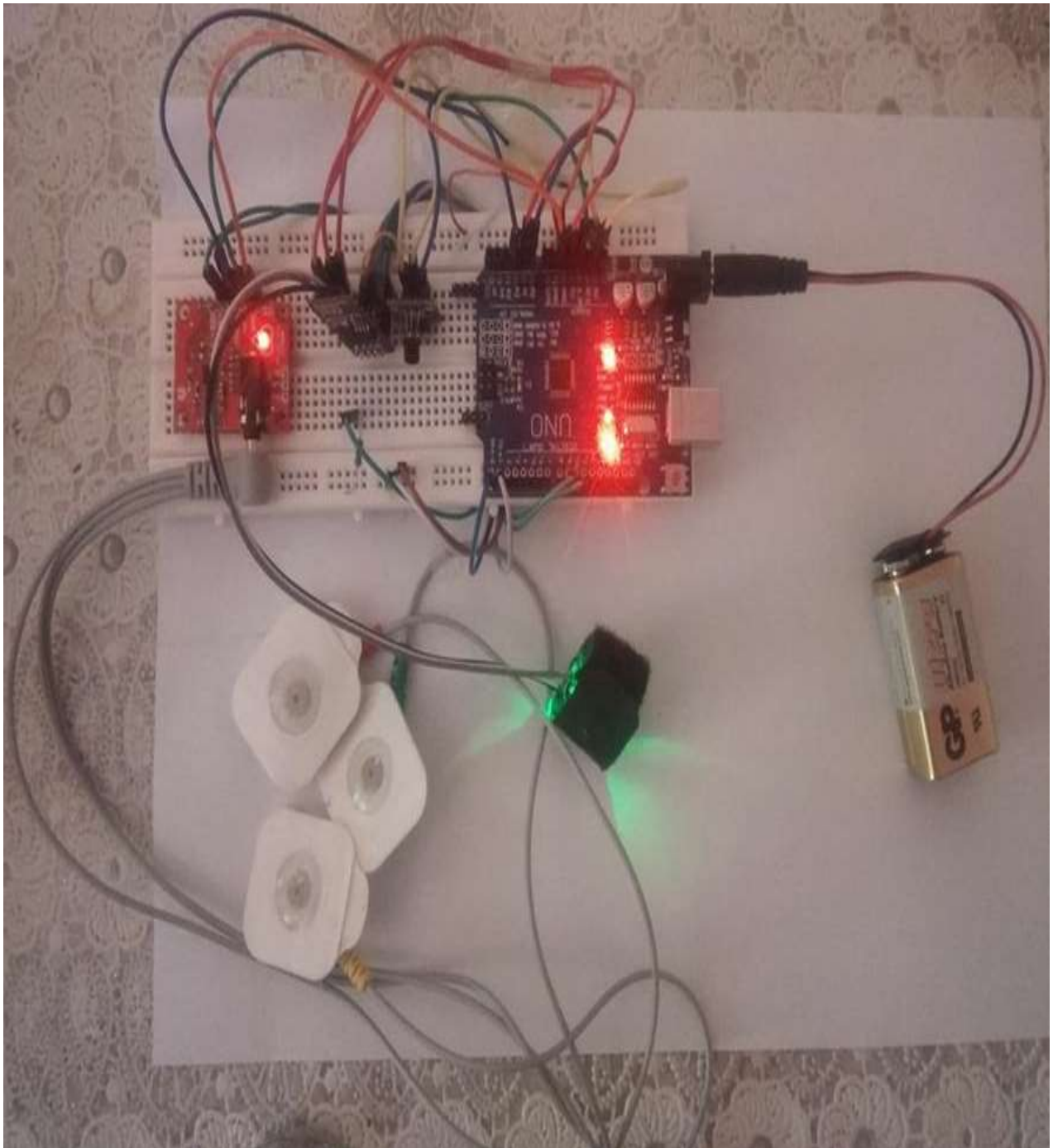


Figure 3.26 : Photo de montage fini

III.4.3.1 Les Tests avec Commande AT

Une fois nous avons terminé le chargement du programme avec succès. Nous passerons nécessairement à la vérification de la connectivité du module WIFI avec un ensemble de commande AT. ces commandes constituent un langage de commandes développé à l'origine pour le modem Hayes Smart modem 300. Ce jeu de commandes s'est ensuite retrouvé dans tous les modems produits. L'ESP8266 est un SoC (System on Chip) qui contient un microcontrôleur et un système WiFi. Il est pourvu d'un port RS232 qu'on peut

utiliser pour lui envoyer des commandes AT.

III.4.3.1.1 Connexion à un réseau

Après la mise sous tension de la carte, le système commence par la recherche d'un réseau WIFI accessible (ouvert) ou un réseau sécurisé qu'on a son mot de passe. Une fois la carte trouve un réseau elle se connecte. Et elle envoie un témoin on allumant en permanence une LED. Si ce n'est pas le cas elle prévoit un stockage directe dans une carte mémoire externe pour éviter la perte des informations voire la figure 3.47.

```
AT+CWJAP="DJAWEB_0858","80A1D7F40858"
WIFI CONNECTED
AT+CIPMUX=1
busy p...
WIFI GOT IP

OK
```

Figure 3.27 : la connexion de l'esp8266 avec le modem.

III.4.3.1.2 Envoie d'une donnée :

Après la connexion, les données sont envoyées via le protocole TCP à notre destination dans notre site web comme indiqué dans la figure 3.48 sur la page suivante.

```
AT+CIPSTART=4,"TCP","192.168.1.4",80
4,CONNECT

OK
AT+CIPSEND=4,256

OK
>
Recv 256 bytes

SEND OK

+IPD,4,62:New record created successfullyNew record created successfully4,CLOSED
AT+CIPSTART=4,"TCP","192.168.1.4",80
4,CONNECT

OK
AT+CIPSEND=4,256

OK
```

Figure 3.28 : la transmission des données dans le protocole TCP.

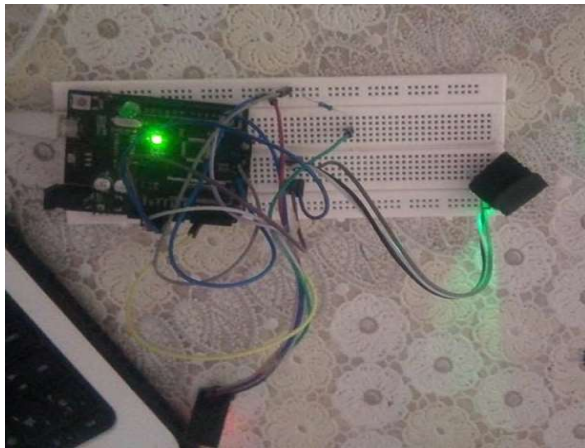


Figure A : Avant transmission de donnée

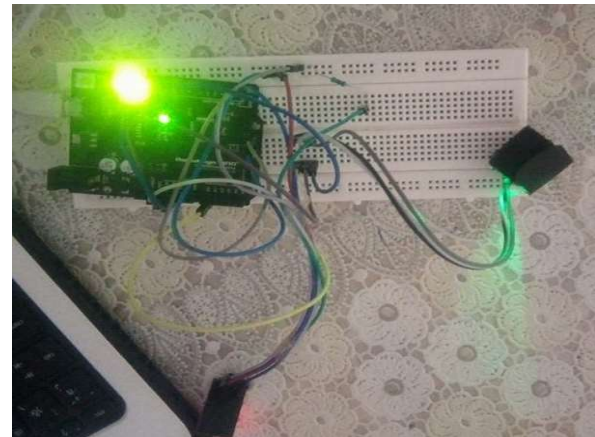


Figure B : instant de envoie de donnée

Figure 3.29 : la transmission des données.

III.4.3.1.3 Recevoir des données :

Nous avons reçu des données envoyées via TCP dans la base de données.

Les valeurs température et ECG dans la base de données.

	id	temp	createddate
<input type="checkbox"/> Éditer Copier Supprimer	12	21.02	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	13	19.55	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	14	27.37	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	15	19.55	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	16	35.68	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	17	19.55	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	18	26.39	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	19	21.99	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	20	20.53	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	21	19.06	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	22	21.51	2018,04,23
<input type="checkbox"/> Éditer Copier Supprimer	23	26.88	2018,05,31

	id	ecg
<input type="checkbox"/> Éditer Copier Supprimer	3009	238 238 239 239 239
<input type="checkbox"/> Éditer Copier Supprimer	3008	238 239 239 239 239
<input type="checkbox"/> Éditer Copier Supprimer	3007	239 239 239 239 238
<input type="checkbox"/> Éditer Copier Supprimer	3006	238 239 239 239 239
<input type="checkbox"/> Éditer Copier Supprimer	3005	153 137 160 151 164
<input type="checkbox"/> Éditer Copier Supprimer	3004	137 149 150 160 159
<input type="checkbox"/> Éditer Copier Supprimer	3003	182 117 183 137 178
<input type="checkbox"/> Éditer Copier Supprimer	3002	138 147 152 157 162
<input type="checkbox"/> Éditer Copier Supprimer	3001	132 153 145 160 158
<input type="checkbox"/> Éditer Copier Supprimer	3000	145 142 155 153 162
<input type="checkbox"/> Éditer Copier Supprimer	2999	141 142 158 150 168
<input type="checkbox"/> Éditer Copier Supprimer	2998	140 148 148 163 155
<input type="checkbox"/> Éditer Copier Supprimer	2997	139 149 147 163 155
<input type="checkbox"/> Éditer Copier Supprimer	2996	132 154 145 163 157
<input type="checkbox"/> Éditer Copier Supprimer	2995	134 147 148 155 165

Figure 3.30 : la réception des valeurs de température et ECG dans la base de données.

III.4.3.2 Organigramme de la carte de commande

Dans cette partie nous avons développé un programme en langage arduino.

Le programme commence par la déclaration des bibliothèques, l'initialisation des variables et le chargement des consignes de régulation depuis la mémoire interne. Selon la disponibilité d'un réseau Wifi et d'une connexion internet. Les capteurs prennent de nouvelles données pour différentes périodes et les envoient au panneau Arduino, Le panneau Arduino envoie les nouvelles données reçues par les capteurs au serveur via Wi-Fi. La figure 3.31 présente l'organigramme du programme principal de la carte de commande.

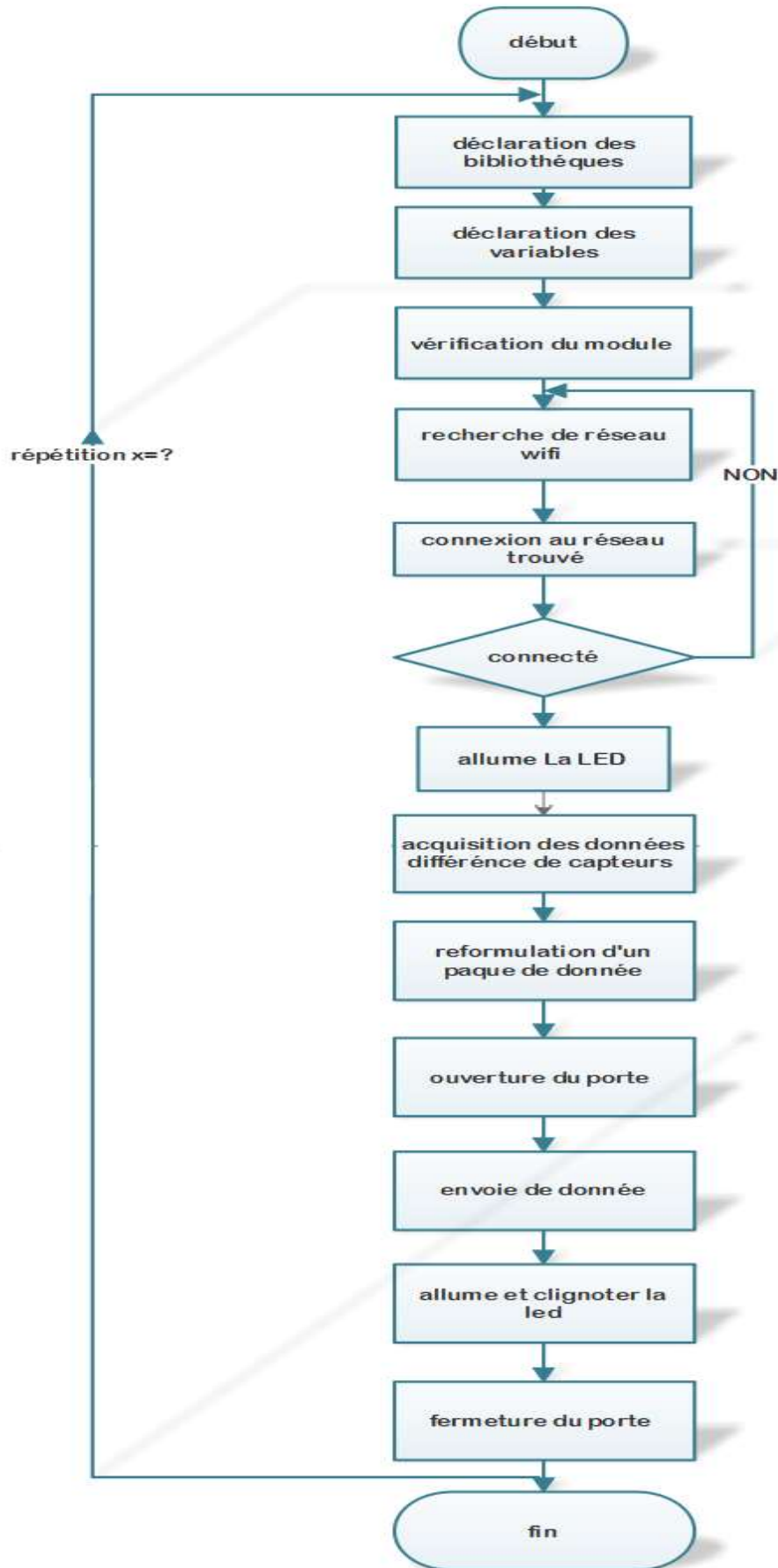


Figure 3.31 :L'organigramme du programme

III.5 Partie récepteur :

Dans cette partie, nous avons utilisé un ordinateur qui doit être configuré Comme un serveur local, dont nous avons développées deux pages Web, une pour recevoir les données de la carte Arduino à partir du réseau internet. Et la deuxième pour l'enregistrement et le traitement des données pour qu'elles puissent être exploités

Le superviseur (le médecin) peut accéder à ces information par internet à travers son smart phone tablette ou simplement son PC. Notre chois été l'utilisation d'un serveur local de type XAMP

III.5.1 Définition d'un serveur :

Un serveur est un ordinateur relié à Internet. Son rôle est de servir, d'où son nom, des données à celui qui lui en demande. Ce demandeur peut être un autre serveur ou l'ordinateur d'un utilisateur final. Les données servies peuvent être de toute nature : sons, images, texte, résultats mathématique. [13]

III.5.2 Serveur http :

Le serveur HTTP utilisé pour gérer votre site va dépendre notamment du système d'exploitation sur lequel il sera déployé (Linux, Windows, etc.). Les serveurs HTTP conseillés pour notre script sont : • Apache en version 2 au minimum. C'est le serveur le plus populaire car il est libre, fiable et multiplateforme. • Microsoft IIS, pour Windows uniquement, dans sa version 7 au minimum. [14]

III.5.3 L'environnement de développement XAMPP :

Maintenant que vous connaissez les prérequis, vous pouvez mettre en place votre environnement de développement local en toute connaissance de cause. Dans cette section, nous allons décrire l'installation et la configuration du serveur XAMPP, qui inclut tous les composants requis pour le bon fonctionnement de notre script et est disponible pour Windows, Linux et Mac OS X. [14]

III.5.3.1 Présentation de server local XAMP [14] :

XAMPP est un paquetage logiciel contenant tout ce dont vous avez besoin pour mettre en place facilement et rapidement un serveur web. Il comprend notamment :

- un serveur web, Apache.
- l'interpréteur PHP.
- le système de gestion de bases de données MySQL.
- un serveur FTP, FileZilla.
- un serveur de courrier électronique, Mercury.

- des outils d'aide à la configuration et à la gestion du serveur.

Installation

1. Téléchargez le programme d'installation de XAMPP depuis la page <http://www.apachefriends.org/fr/xampp-windows.html> vers un répertoire local.
2. Une fois le fichier récupéré, faites un double-clic dessus pour lancer le programme d'installation. Selon votre version de Windows et le niveau de sécurité du compte utilisateur, une boîte de dialogue apparaît et vous demande d'autoriser explicitement le logiciel à apporter des modifications sur votre ordinateur : cliquez sur le bouton Oui.
3. Choisissez la langue d'installation parmi l'anglais ou l'allemand. Suivant votre version de Windows, un avertissement apparaît concernant le contrôle des comptes d'utilisateurs (voir Figure 3.32) sur la page suivante.



Figure 3.32 : choix de la langue d'installation

4. Vous accédez maintenant à l'assistant d'installation de XAMPP. Cliquez sur le bouton NEXT pour lancer le processus.
5. Le programme d'installation décompresse alors tous les fichiers requis et le processus d'installation est terminé. Cliquez sur le bouton FINISH pour quitter l'installateur. Une boîte de dialogue vous demande si vous souhaitez lancer le panneau de contrôle XAMPP : cliquez sur le bouton Oui.

Le panneau de contrôle illustré à la Figure 3.33 s'ouvre alors : votre serveur web est installé et fonctionnel.

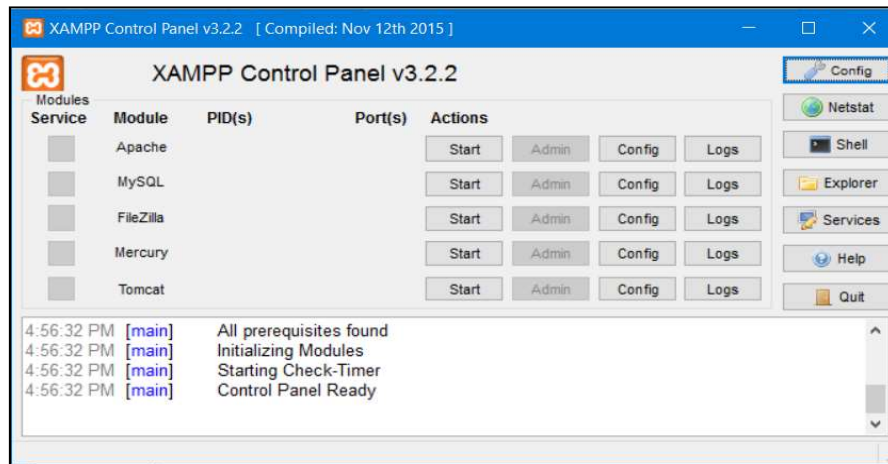


Figure 3.33 : Panneau de contrôle de XAMPP

III.5.4 La base de données :

Une base de données est un ensemble de données structurées correspondant généralement à un domaine fonctionnel (facturation, ressources humaines, etc.).

Physiquement, une base de données se matérialise par un ensemble de fichiers stockés sur un périphérique de stockage.

Dans notre cas nous utilisons une base de données nommé « Arduino » constitué des trios tableaux (ECG, temp, pulse) (voir Figure 3.34).

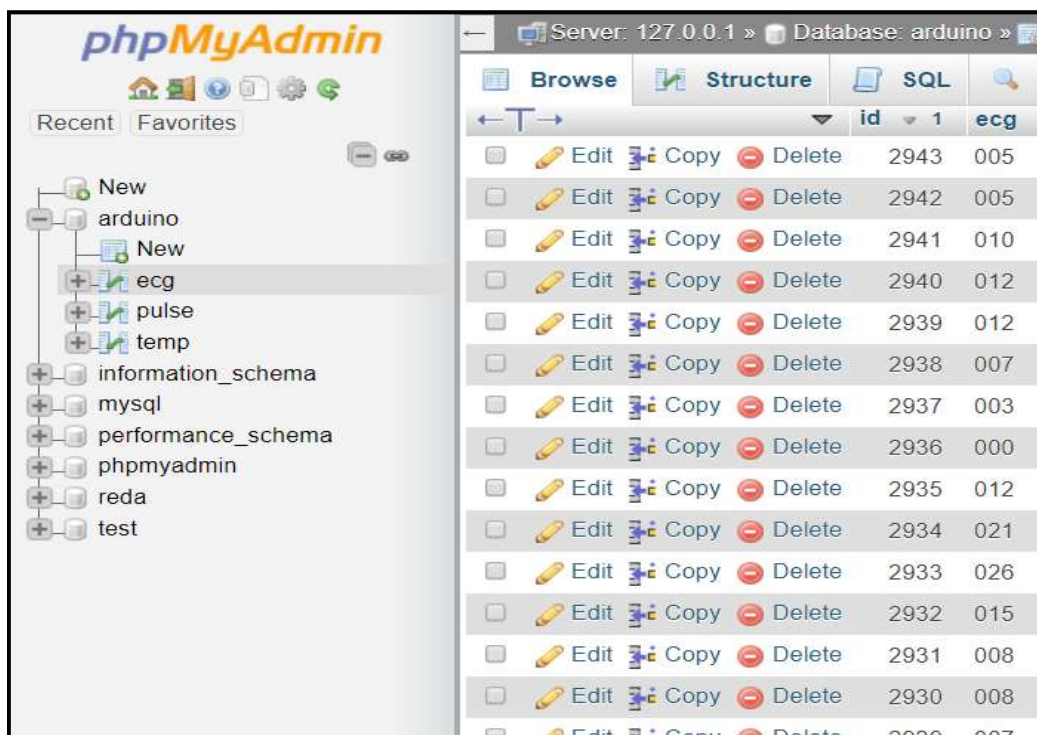


Figure 3.34 : Une base de données permet de stocker la totalité des informations relatives à une activité.

III.5.5 La programmation de la page de la réception :

PHP est l'abréviation de PHP HyperText Preprocessor. C'est le langage de programmation utilisé pour écrire notre script et ses extensions complémentaires. Notre script nécessite au minimum la version 5.2.4 de PHP. Il est recommandé d'utiliser si possible la version 5.3. Dans tous les cas, les versions antérieures de PHP ne sont pas compatibles.

Le but du programme de cette page est de recevoir des données émis par la carte Arduino a parti du réseau internet et les enregistre dans des variables prédéfinies [\$temp, \$pulse, \$ecg], puis les stocke en contenu dans notre base de données « **Arduino** » avec la instruction « **INSERT INTO** » (Voir Figure 3.35) sur la page suivante.

```
1 <?php
2 $temp = "";
3 include_once("include/config.php");
4 if(isset($_GET["pulse"]) || isset($_GET["ecg"])){
5 $pulse = $_GET["pulse"];
6
7 $date = date('Y,m,d');
8 $sql = "INSERT INTO pulse (pulse) VALUES ('$pulse')";
9 if ($mysqli->query($sql) === TRUE) {
10 echo "New record created successfully";
11 } else {
12 echo "Error: " . $sql . "<br>" . $mysqli->error;
13 }
14 if(isset($_GET["temp"])){
15 $temp = $_GET["temp"];
16 $sql1 = "INSERT INTO temp (temp,createddate) VALUES ('$temp', '$date')";
17
18 if ($mysqli->query($sql1) === TRUE) {
19 echo "New record created successfully";
20 } else {
21 echo "Error: " . $sql1 . "<br>" . $mysqli->error;
22 }
23 }
24 if(isset($_GET["ecg"])){
25 $ecg = $_GET["ecg"];
26 $sql2 = "INSERT INTO ecg (ecg) VALUES ('$ecg')";
27
28 if ($mysqli->query($sql2) === TRUE) {
29 echo "New record created successfully";
30 } else {
31 echo "Error: " . $sql2 . "<br>" . $mysqli->error;
32 }
33 }
34 }
35 }
36 }
37 }
38 ?>
```

Figure 3.35 : Code PHP de page de réception

III.5.6 Le serveur de bases de données MySQL :

MySQL est un système de gestion de bases de données (SGBD). C'est l'un des logiciels les plus utilisés au monde, notamment par les applications web. Il est disponible pour de nombreux systèmes d'exploitation et les bases de données peuvent être exploitées par

différents langages de programmation, nous utilisons l'instruction « `newMySQL` » pour connecter à la base de données. Le code correspond à la figure suivante (Voir Figure 3.36).

```

1  <?php
2  /* Open connection to "zing_db" MySQL database. */
3  $mysqli = new mysqli("localhost", "root", "", "arduino");
4
5  /* Check the connection. */
6  if (mysqli_connect_errno()) {
7      printf("Connect failed: %s\n", mysqli_connect_error());
8      exit();
9  }
10 ?>

```

Figure 3.36 : Code PHP de la page de réception

III.5.7 La programmation de la page de l’affichage :

Le but principal de cette page est de se connecter à la base de données et puis lire les valeurs stockées de chaque tableau (Nous savons que chaque tableau représente des valeurs pour un capteur, par exemple l’ECG sensor) et d’afficher les résultats à un moment donné. L’affichage des résultats sous forme de valeurs numériques (voir Figure 1.6) ou bien des courbes graphiques ce qui permet au superviseur (le médecin) d’analyser facilement les résultats obtenus (voir Figure 3.37), cette page est hébergée directement dans le PC du médecin.

```

<?php

header ("refresh:2;");
echo "<br />";
$data = file_get_contents("status.txt");
echo ($data);
?>

```

Figure 3.37 : La programmation de la page de l’affichage :

III.5.8 Organigramme de la partie serveur :

La figure présente l'organigramme du programme du serveur web, il permet de faire le lien entre la carte de commande et l'interface web. Il permet aussi de gérer la base de données.

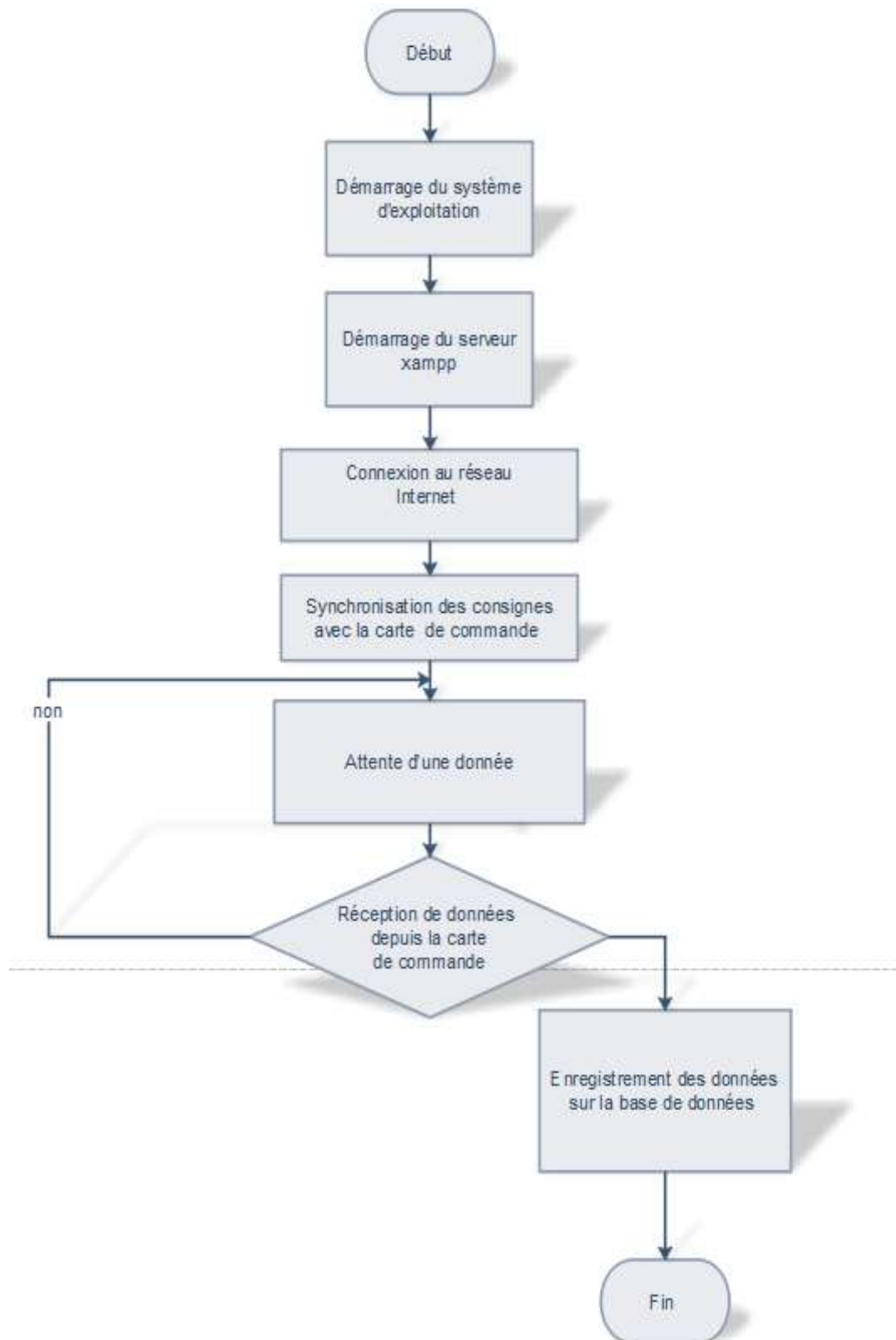


Figure 3.38: Diagramme du programme de la partie serveur

III.6 Présentation de l'interfaçage :

Les interfaces graphiques de l'application sont très importants, car ils facilitent le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application. Dans la conception des interfaces de notre application nous avons respecté un ensemble des choix ergonomiques comme la lisibilité, la compréhensibilité, etc. Dans ce qui suit une présentation des captures écrans des interfaces de l'application

III.6.1 Interface principale :

Afin de pouvoir visualisé les données stoker dans la base de données, nous avons développé un interface Homme-Machine sous forme d'application web (figure 3.39), ce type d'interface est très utilisé de nos jours car il offre la possibilité de contrôler un système depuis n'importe quel ordinateur connecté à internet sans avoir besoin d'installer des logiciels ou pilotes supplémentaires.

Pour avoir accès à cette application, il suffit juste d'ouvrir le navigateur web et d'y entrer le nom de domaine de l'application (dans notre cas on introduit l'adresse du serveur). Pour la publication du serveur sur internet il nous faut un nom de domaine. L'interface de la figure 3.39 s'ouvre. Cette page est divisée en plusieurs zones.

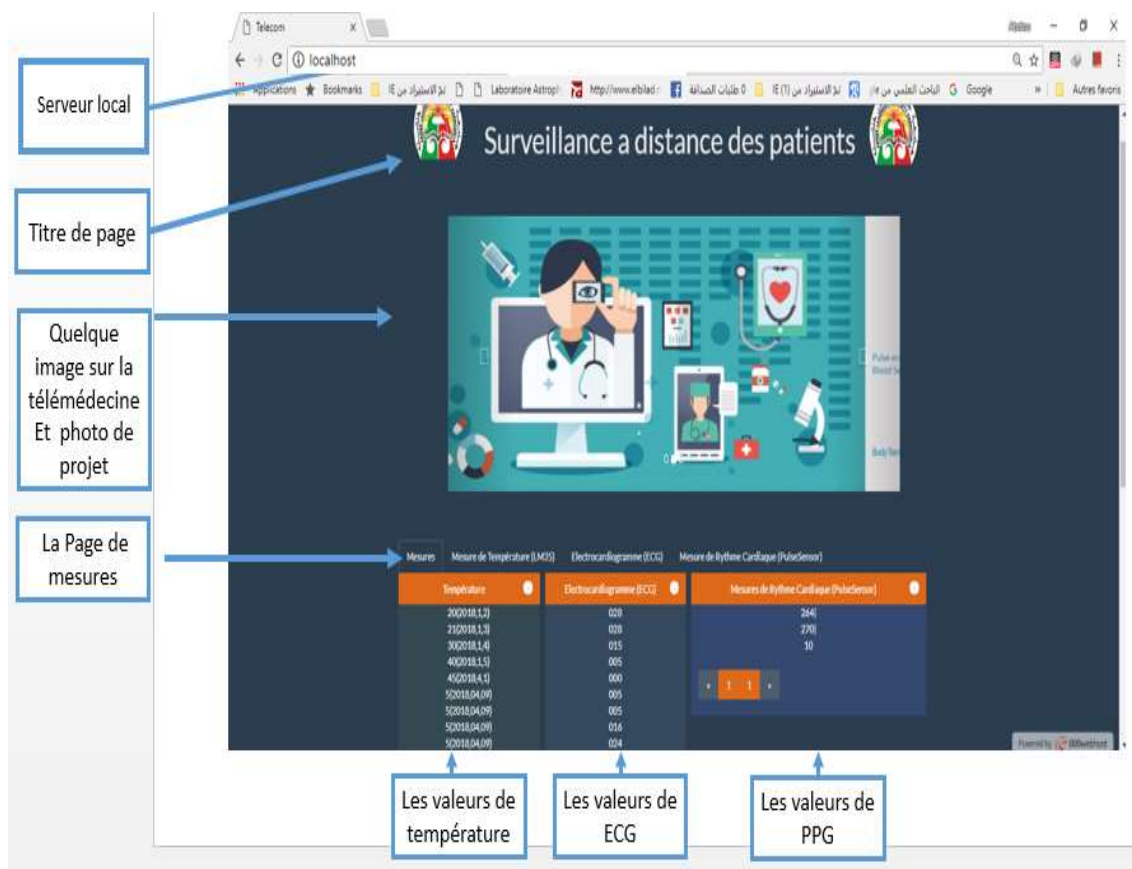


Figure 3.39: Page principale de notre interface Homme-Machine.



Figure 3.40 : Les données de température

Sur cette zone de l'interface principale figure les données de la température reçue.

Ces mesures ont été prises à différents intervalles de temps Comme vous le verrez dans Figure la donnée est représentée sous la forme suivante :

`T° en degré (la date, l'heure)`

Exemple :

`20 °C 6:51 PM (2018, 1,2)`

20 °C : Cette valeur représente une température en degré
6 :51 l'heure

2 : Représente le jour

1 : Représente le mois

2018 : Représente l'année

Les données sont affichées en page de 10 valeurs. Si en veut visualiser d'autre valeur il suffit de sélectionner le numéro de la page sur pagination



Figure 3.41: Les données de signal D'ECG

Pour cette zone de l'interface sont présentées les données du signal de l'ECG

Ces mesures ont été prises aussi à différents intervalles de temps contrairement à la température les données sont reçu en paqué de 10 échantillons.

Comme vous le verrez dans Figure

028 ; 028 ; 015 ; 005 ; 011 ; 000 ; 016 ; 024 ; 026 ; 022 :

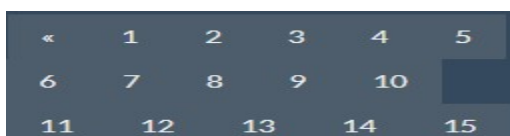
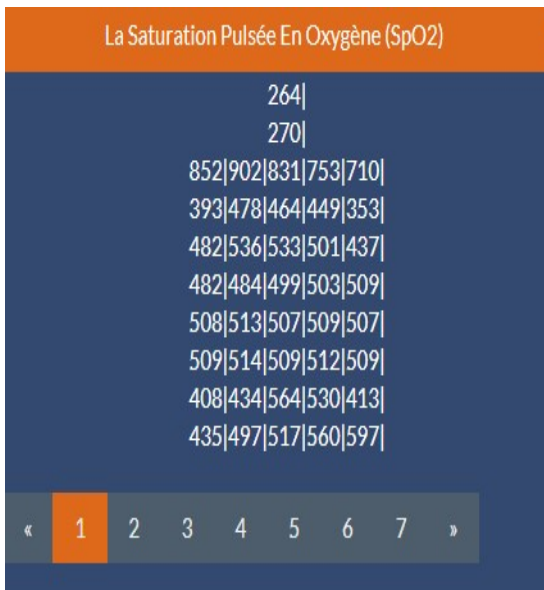


Figure 3.42 : Pagination de signal ECG

En raison du grand nombre d'échantillons ECG, nous avons Création de pagination Chaque page contient plusieurs échantillons du signal ECG.



Cette zone de l'interface représente différentes mesures de rythme du battement cardiaque

Ces mesures ont été prises à différents intervalles de temps

Comme vous le verrez dans Figure

852|902|831|753|710|393|478|464|449|353| nombre de battements cardiaques (ou pulsations) par unité de temps

Figure 3.43 : Les données de pouls

III.6.2 Interface de la température :

On peut accéder à l'interface de la température à partir de l'interface principale par un simple clic ; L'interface suivant permet l'affichage de la température du patient sous forme d'un graphe (ou courbe)

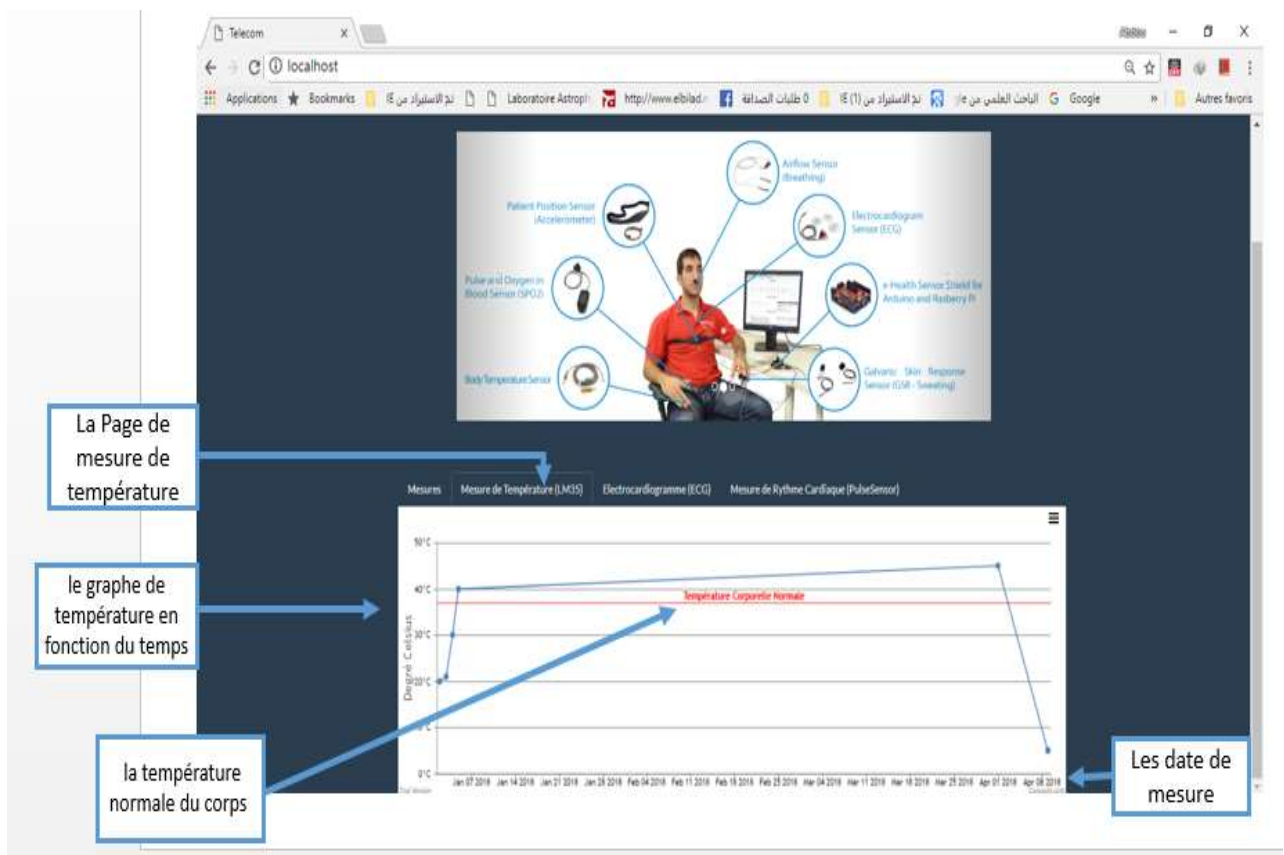


Figure 3.44: interface de température

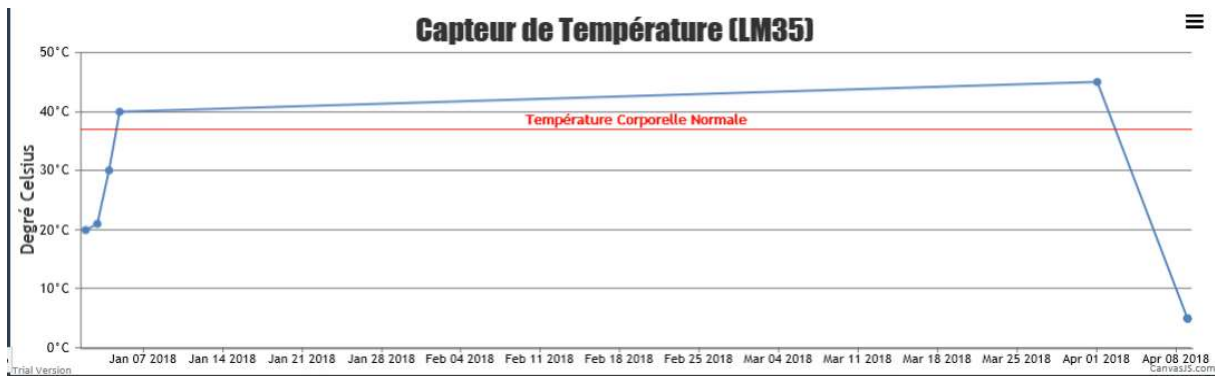


Figure 3.45 : La courbe de température

Cette figure représente une courbe de la température du patient à un intervalle de temps.

Dans cet exemple le capteur de mesure de la température a été fixé sur la main du patient. Le patient ont été surveillés pendant 4 mois (7 janvier 2018 ; 8 avril 2018). La figure 3. Montre les résultats pour la mesure de la température.

III.6.3 Interface ECG :

On peut accéder à l’interface de l’électrocardiogramme à partir de l’interface principale par un simple clic L’interface suivant permet l’affichage de le signal ECG du patient sous forme d’un graphe (ou courbe)

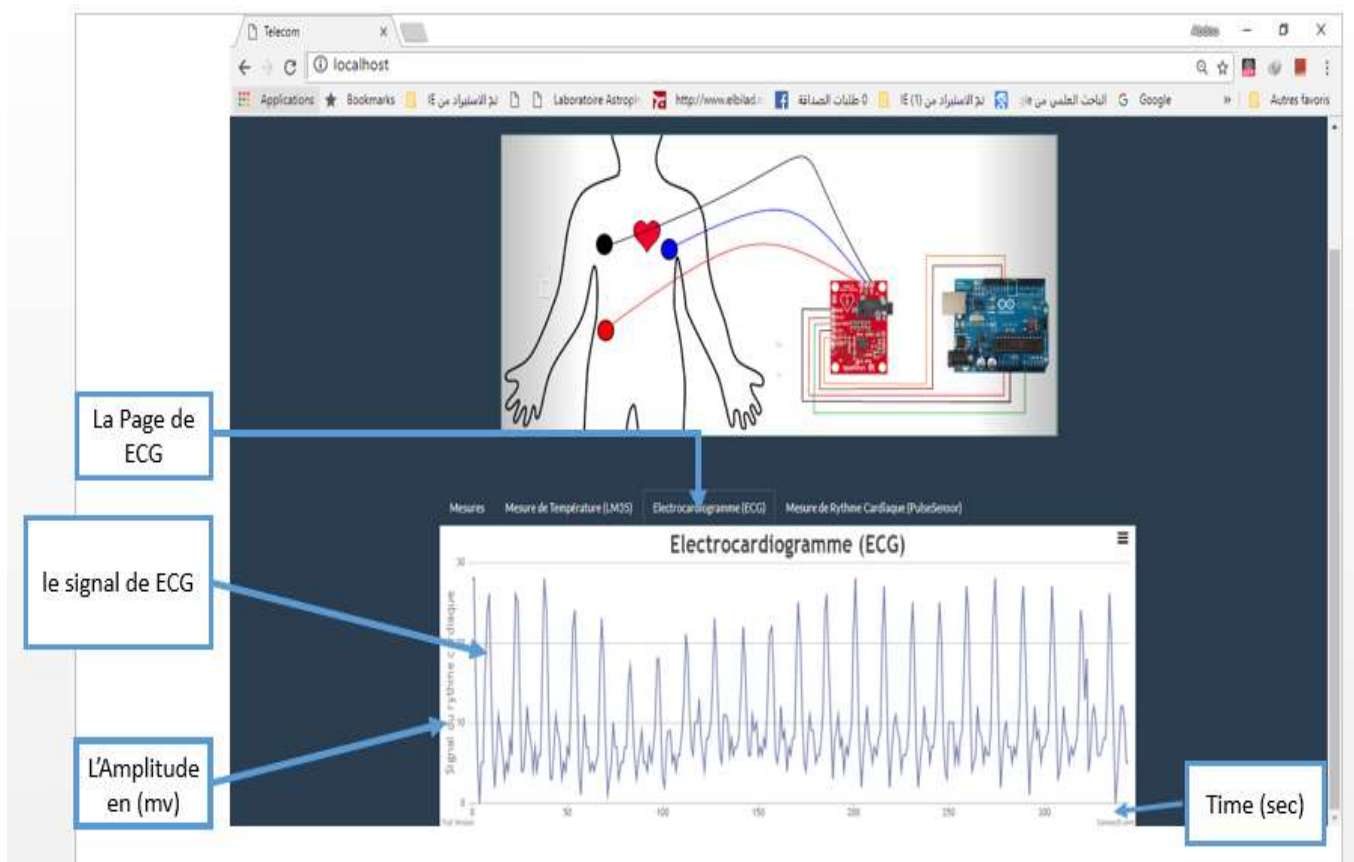


Figure 3.46: Interface ECG

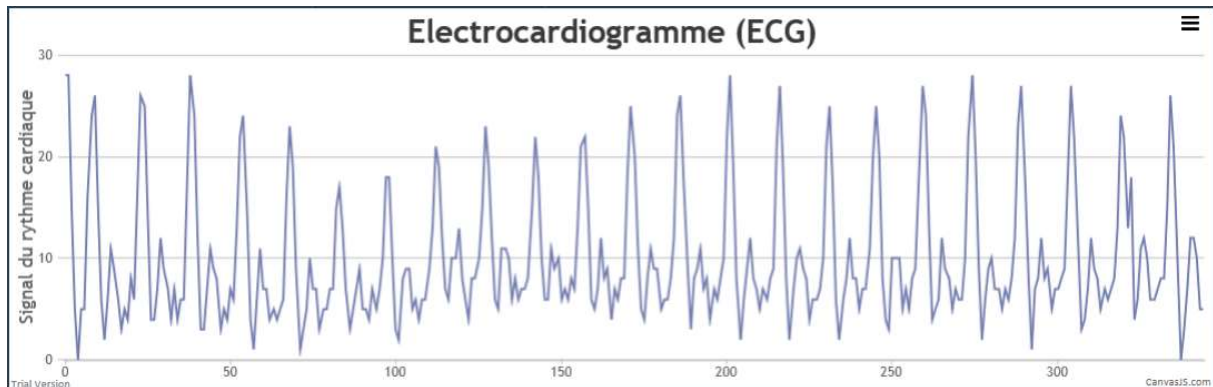


Figure 3.47 : Le signal d'ECG

Cette courbe représente l'ECG du patient sur une période de temps définie.

Le capteur ECG est connecté au corps du patient à la maison, comme le montre la figure 3.

Il représente la variation de l'amplitude en fonction du temps.

III.6.4 Interface La saturation pulsée en oxygène (SpO2):

On peut accéder à l'interface de l'électrocardiogramme à partir de l'interface principale par un simple clic

L'interface suivante permet l'affichage de le signal ECG du patient sous forme d'un graphe (ou courbe)



Figure 3.48: Interface La saturation pulsée en oxygène (SpO2)

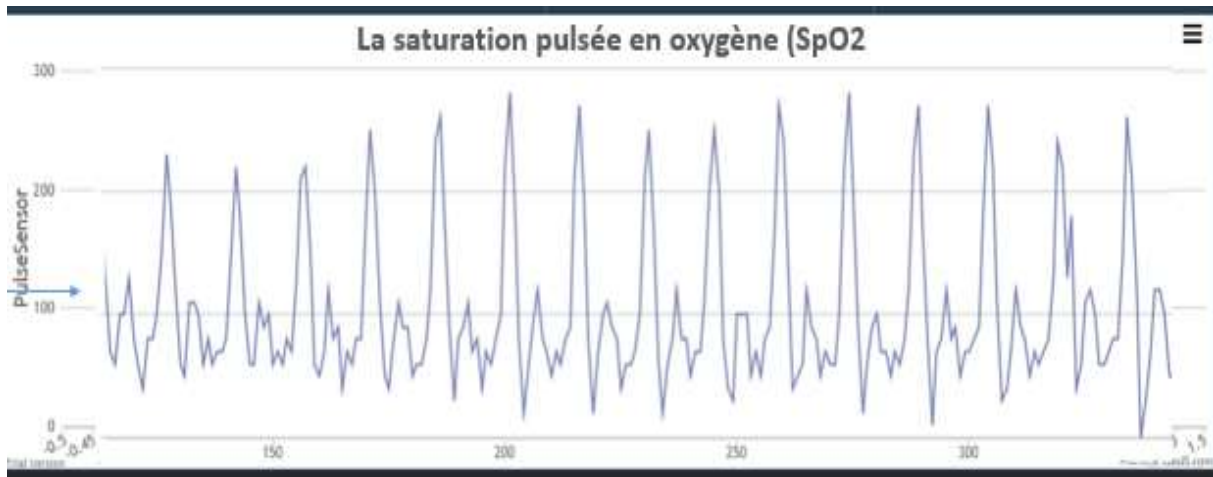



Figure 3.49 : Signal de La saturation pulsée en oxygène (SpO2)

Cette courbe représente la saturation pulse en oxygène du patient sur une période de temps définie.

Le capteur spo2 est connecté au corps du patient, comme le montre la figure 3. Il représente la variation de impulse en fonction du temps.

III.6.5 Éléments de graphique

A. Le Zoom

Cette option sert à la mise en point du tracé graphe. Pour zoomer le signal il faut sélectionner la partie souhaité puis cliquer sur 

Par Exemple le graphe ECG :

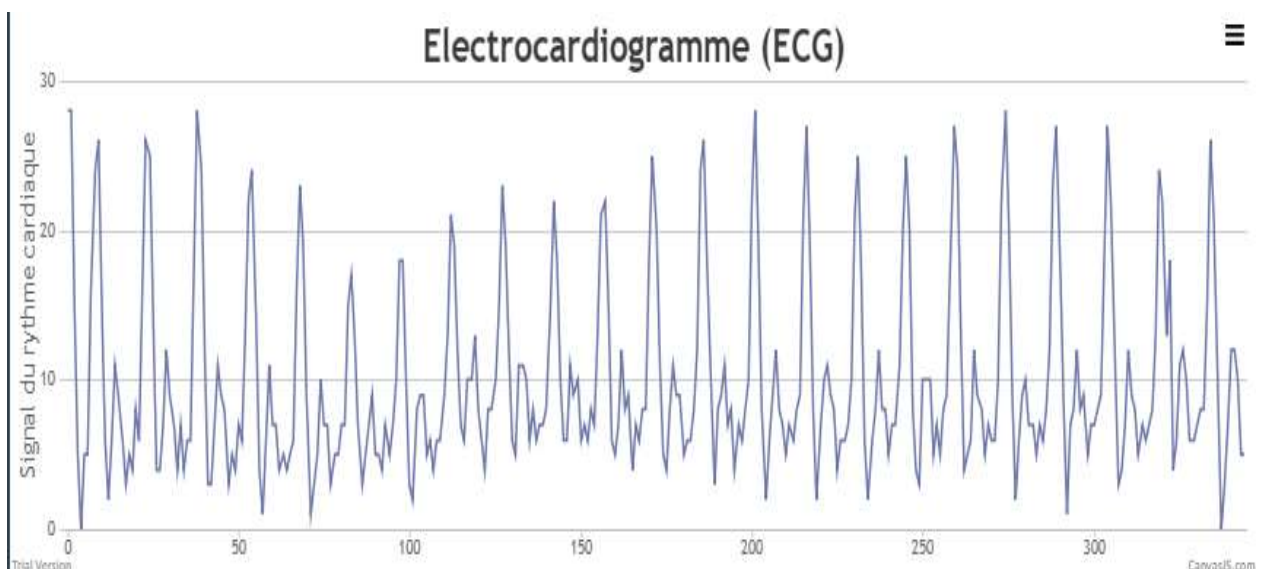


Figure 3.50: le signal ECG

La figure suivant montre une partie agrandie d'un signal 'ECG

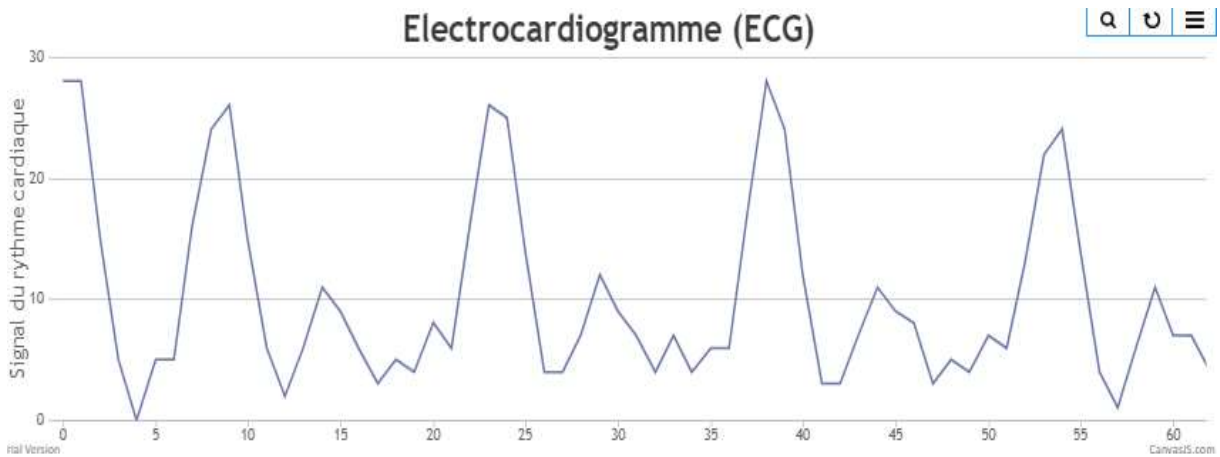


Figure 3.51: Partie agrandie de l'ECG.

B. Position

Ces boutons **+** servent à positionner le tracé de température

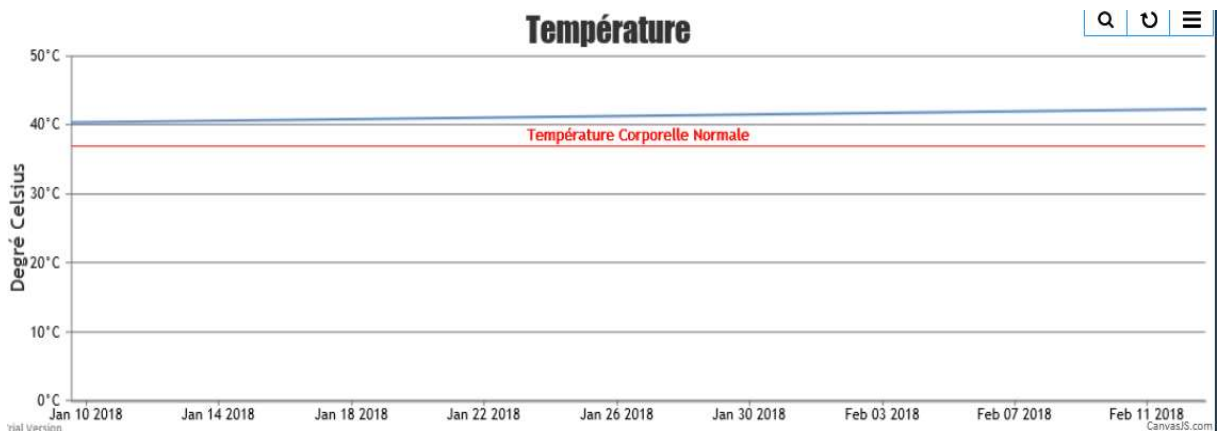


Figure 3.52 : Positionner de température.

C. Enregistrer le graph de l'ECG sur format image

Cette option permet d'enregistrer le graphique sous forme d'image (JPEG ou PNG)

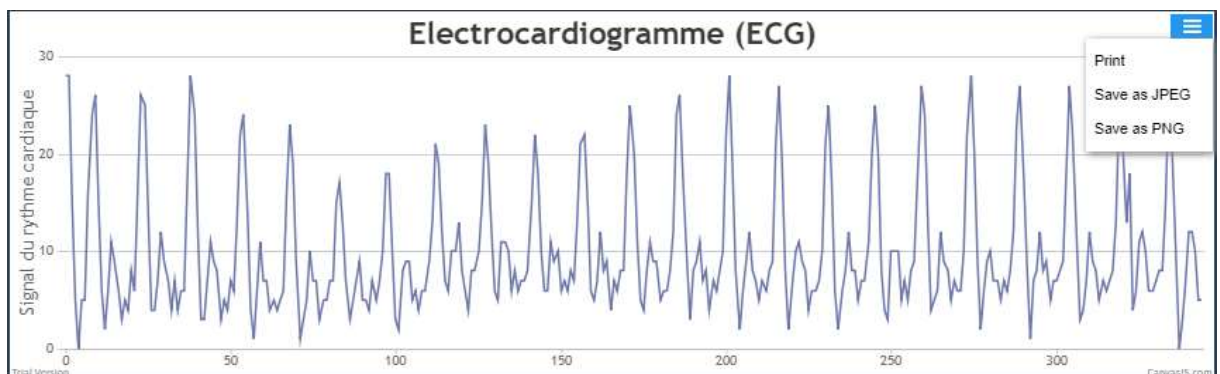


Figure 3.53 : Enregistrer le graph de l'ECG forme JPEG ou PNG

D. Imprimer les courbes sur papier

Nous pouvons également imprimer l'ECG comme indiqué dans la figure ci-dessous



Figure 3.54 : imprimer l'ECG sur papier.

III.7 Conclusion

Au début de ce chapitre, nous avons donné une description détaillée tous les éléments utilisés (carte Arduino, capteurs, module WIFI), puis nous avons fourni une explication détaillée de la méthode de conception du système de télésurveillance médicale.

Le système de surveillance à distance des patients proposé est l'intégration d'une application embarquée et web, fournit une plate-forme de manière rentable, solution pour le patient et le médecin situé dans un endroit éloigné. Le médecin peut arriver à une conclusion par examiner et surveiller les paramètres de santé de patients à des endroits éloignés. Le changement anormal dans les valeurs de la santé des patients paramètres peuvent alerter le médecin et aider à prendre les actions nécessaires qui sont possibles. Ce système fournit une lecture en temps réel des patients physiologiques paramètres avec sa démographie, ce qui aidera à diagnostic de santé du patient

CONCLUSION GÉNÉRALE

Conclusion Générale

Dans ce mémoire nous nous sommes concentrés à la Réalisation d'un système de télésurveillance médicale de l'acquisition des données grâce aux trois capteurs: de Température, ECG, de PPG, utilisant un protocole de communication sans fil (wifi) avec développement d'une application de télémédecine qui collecte les données d'un patient.

Ce mémoire présente le système de surveillance de la santé utilisant système embarqué. Les caractéristiques associées sont le besoin de les médecins sont moins, facile à utiliser, système polyvalent utilisé pour mesurer les différents paramètres. Avec son utilisation potentielle dans les hôpitaux et les domaines de la santé à domicile, capteur sans fil les réseaux ont un rôle important dans l'amélioration de la vie des patients .Dans ce cadre on a proposé et développé un prototype d'une application de télémédecine.

Notre implémentation est basée sur trois parties essentielles :

1. L'installation et la programmation de la carte Arduino avec le capteur utilisé.
2. La programmation des pages web coté server pour la réception le traitement et l'affichage des données.
3. L'envoi des données à travers les réseaux WIFI vers le server.

Problème rencontré :

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous sommes confrontés à plusieurs problèmes parmi eux :

- Pas disponibilité de matériel au niveau du laboratoire de l'université.
- Pas disponibilité des capteurs biomédicaux sur le marché.
- Le cout élevé des différentes composantes surtout les capteurs biomédicaux.
- La grande quantité des données envoyées par les capteurs et difficile à traiter « Le big data ».
- Problème avec la programmation et la configuration de base de données phpmyadmin.
- Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie de la connexion sans fil. Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études. Nous espérons que ce mémoire sera une référence aux personnes désirant développer et réaliser des projets et systèmes à base des cartes Arduino

Perspective du projet

Pour apporter une solution cohérente dans le domaine de la télésurveillance médicale, notre système de suivi peut prendre en compte différentes pathologies. Il doit par conséquent être ouverts et évolutifs. Il doit aussi assurer une transmission, et présente au médecin toutes les informations utiles dans un contexte simple et explicite. Cependant, différentes problématiques restent encore ouvertes en perspectives,

- Nous pouvons améliorer la fonctionnalité de ce projet en utilisant la technologie GSM ou Zigbee pour la transmission de données.
- Introduire des traitements des données dans la partie client (coté capteur) afin de ne pas envoyer les données non utiles.
- Nous ajoutons le GPS en cas d'épilepsie chez les patients épileptiques.
- Développer une application mobile Android pour permettre le médecin de voir les données des patients à tout moment.
- ajouter un écran LCD au l'application pour afficher des notifications au patient (client)
- Ajouter une option dans la page web pour envoyer un message SMS au médecin (superviseur) en cas d'urgence

RÉFÉRENCES

BIBLIOGRAPHIQUES

Références bibliographiques

Introduction générale

- [1]. Real time wireless health monitoring application using mobile devices, International Journal of Computer Networks & Communications (IJCNC) Vol.7, No.3, May 2015, Amna Abdullah, Asma Ismael, Aisha Rashid, Ali Abou-ElNour, and Mohammed Tarique
- [2]. Secured Smart Healthcare Monitoring System Based on Iot, International Journal on Recent and Innovation Trends in Computing and Communication Volume: 3 Issue: 7, Bhoomika.B.K, Dr. K N Muralidhara
- [[3]. Home Based Health Monitoring System Using Android Smartphone, International Journal of Electrical, Electronics and Data Communication, Vol-2, Issue-2, Feb-2014, Sushama Pawar, P.W.Kulkarni

Chapitre 1 : Généralité sur le système Arduino

- [1] Organisation fonctionnelle d'un système à microcontrôleur (G BERTHOME – Lycée Mireille GRENET – COMPIEGNE)
- [2] http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterIntroduction
- [3] <http://blewando.dlinkddns.com/elv/Promo2018/th14/pag6.html>
- [4] <https://www.lextronic.fr/5608-moteurs-robotique>
- [5] <https://www.generationrobots.com/fr/174-shield-arduino>
- [6] [https://www.arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoard Bluetooth](https://www.arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth)
- [7] <https://www.arduino.cc/en/Main/ArduinoGSMShield>
- [8] [https://en.wikipedia.org/wiki/ESP8266 Kit wifi](https://en.wikipedia.org/wiki/ESP8266_Kit_wifi)
- [9] Hamid HAMOUCI. Conception et réalisation d'une centrale embarquée de la domotique « Smart Home ».06/07/2015, mémoire de master en génie électrique, Université Mohammed V École Normale Supérieure d'Enseignement Technique – Rabat.
- [10] : « Electronique avec Arduino », cour de Pascal Masson, édition 2015/2016, v32.

Chapitre 2 :

[1] <http://fr.wikipedia.org/wiki/capteur>.

[2] Georges Asch Bernard Poussery LES CAPTEURS EN INSTRUMENTATION INDUSTRIELLE
Préface de Marc Desjardins, ancien président du comité « Capteurs » au ministère de la Recherche et de l'Industrie 8^e édition

[3] http://gte.univlittoral.fr/sections/documents-pdagogiques/chapitre-8_mesure/downloadFile/file/Les_capteurs.pdf?nocache=1289041293.82

[4] Biomedical Sensors and Measurement auteur Ping.W and Qingjun.L edition
SPRING

[5] http://www-master.ufr-info-p6.jussieu.fr/2005/IMG/pdf/GRETR_biocapteurs-haduong-carlier.pdf

[6] J. G. Webster, « Medical Instrumentation Application and Design», Edition: 4, ISBN-10: 0471676004, ISBN-13: 978-0471676003, February 3, 2009.

[7]. <http://dSPACE.univ-tlemcen.dz/handle/112/1186>

[8]. <https://www.generationrobots.com/fr/402175-capteur-de-temp%C3%A9rature-corporelle-pour-plateforme-e-health.html>

[9] <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>

[10] <https://fr.wikipedia.org/wiki/Glucom%C3%A8tre>

[11]: B. BENLADGHAM et S. BAHRI « la télésurveillance cardiaque », D'ingénieur d'état en électronique biomédical, Université Abou Bekr Belkaid, Tlemcen, Algérie, Juin 2003.

Chapitre 3 :

[1] X.HINAULT. www.mon-club-elec.fr.

[2] B. Cottenceau . Carte ARDUINO UNO Microcontrôleur ATmega328[en ligne].Microcontrôleurs EI3 Option AGI.

[3] Christian Tavernier. Arduino Applications avancées[en ligne]. Dunod, Paris, 2012. ISBN 78-2-10-058205-1

[4] Jean-Luc. Les écrans LCD alphanumériques [en ligne].Le 3 mars 2015.

[5] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».

[6] <http://www.generationrobots.com/fr/152-arduino>. consulter le: mars 2015

[7] <http://air.imag.fr/index.php/ESP8266>

[8] <https://sites.google.com/site/arduinoencore/home/esp8266>

[9] Ibrahim, D. (2002). Microcontroller-based temperature monitoring and control. John Wiley & Sons.

[10] Sensors, L. P. C. T. datasheet, National Semiconductor. www.ti.com/lit/ds/symlink/lm35.pdf

[11] <https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide>

[12] http://www.odpf.org/images/archives_docs/13eme/memoires/gr-7/memoire.pdf

[13] <http://www.comprendre-internet.com/Qu-est-ce-qu-un-serveur.html>

[14] livre : Joomla!: Créez un site web complet ,De Mihàly Marti, Sandrine Burriel consulte le 31/05/2018 a 2.30 pm

ANNEXE

Le code final Arduino

```
#include <stdlib.h>
#include "SoftwareSerial.h"
int count = 0;
String url = "GET /rec.php?pulse=";
String ssid ="red";
String password="001978001978";
int Signal;
int PulseSensorPurplePin = A1 ;
int ECGPin = A2 ;
SoftwareSerial esp(1, 0); // RX, TX
bool done = false;
void setup() {
  esp.begin(115200);
  delay (1000);
  pinMode(13, OUTPUT);
  esp.println("AT+RST");
  delay(2000);
  esp.println("AT");
  done = esp.find("OK");
  if(!done){
    delay(2500);
    done=esp.find("OK");
  }

  delay(2000);
  esp.println("AT+CWQAP");
  delay(2000);
  esp.println("AT+CWJAP=\"" +ssid+"\", \""+password+"\"");
  done = esp.find("OK");
  if(!done){
    delay(1000);
    done=esp.find("OK");
  }

  delay(2500);
  esp.println("AT+CIPMUX=1");
  done=esp.find("OK");
  if(!done){
    delay(2000);
    done=esp.find("OK");
  }

  delay(5000);
  esp.println("AT+CIFSR");
  done = esp.find("OK");
  if(!done){
    delay(5000);
    done=esp.find("OK");
  }

  }
  dht.begin();
  confirmenvoion();
  //confirmenvoioff();
  confirmenvoion1();
}
```

```
void loop() {
    String chk ;

    // Mesure la tension sur la broche A0
    int valeur_brute = analogRead(A0);

    // Transforme la mesure (nombre entier) en température via un produit en
    croix
    float temperature_celcius = valeur_brute * (5.0 / 1023.0 * 100.0);

    delay (2000);
    esp.println("AT+CIPSTART=4,\"TCP\", \"192.168.111.12\",80");
    delay(2000);
    confirmenvoion1();
    if(esp.find("OK")){
        digitalWrite(13,HIGH);
    }
    delay(1000);
    delay(1000);
    esp.println("AT+CIPSEND=4,256");
    confirmenvoion();
    delay(1000);

    if(count < 3){
        chk = pulse_value()+ "&ecg="+ecg_value();
        count++;
    }else {
        chk = pulse_value()+ "&ecg="+ecg_value()+ "&temp="+
String(temperature_celcius);
        count = 0;
    }
    esp.println(url + chk);
    delay(1000);
    esp.println("AT+CIPCLOSE");
    esp.println("AT");
    delay(1000);
    esp.println("AT");
    delay(1000);
    digitalWrite(13,LOW);
    esp.println("AT+CIPMUX=1");

    delay(1000);
}

String ecg_value(){

    String pulse_data = "";

    // Serial.println(Signal); // Send the Signal value to
Serial Plotter.

    delay(50);

    while(pulse_data.length() < 20){

Signal = analogRead(ECGPin); // Read the PulseSensor's value.

    delay(50);
```

```
//if (Signal<600){
    pulse_data +=( String(Signal) + '|' ) ;
// }

}
delay(50);
return pulse_data;

}
String pulse_value() {
    String pulse_data = "";
    delay(50);
while(pulse_data.length() < 20){
Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's
value.
delay(50);
    if (Signal<600){
        pulse_data +=( String(Signal) + '|' ) ;
    }
}
delay(50);
return pulse_data;
}

void confirmenvoion1(){
    if (esp.find("ERROR")){
        digitalWrite(13,LOW);
        return;
    }

}

void confirmenvoion(){
    if (esp.find("OK")){
        digitalWrite(13,HIGH);
        return;
    }
}
```