



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



N° Série : .....

## Mémoire de fin d'étude

*Pour l'obtention du diplôme de Master*

**Filière : Télécommunications**

**Spécialité : Télécommunications Avancées**

### THEME

# Réalisation d'un système de télésurveillance médicale

**Réalisé par :**

- Ferroudj ABDELGHANI.

**Dirigé par :**

M<sup>r</sup>. Sahour ABDELHAKIM

**Membres de jury :**

Mr.Saïga MOHAMED, Président

Mr.Boumaaraf ABDELAALI, Examineur

*Présenté le 01/07/2017*

# Dédicace

*Ma Mère, Mon Père*

*Affable, honorable, aimable : vous représentez pour moi*

*Le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.*

*Soyez sûrs que je continuerai mon chemin.*

*Je vous dédie ce travail en témoignage de mon profond amour.*

*Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.*

*A mes sœurs :*

*Amel, Saïda*

*A mon frère :*

*Yazid*

*En témoignage de l'attachement, de l'amour et de*

*L'affection que je porte pour vous.*

*A tous les membres de ma famille, petits et grands*

*A tous mes amis de proches :*

*Mehdi, Rached, Moomen, Salah,*

*Houcine, Hichem, Ibrahim*

*Veillez trouver dans ce modeste travail l'expression de mon*

*Affection*

**ABDELGHANI**

# *Remerciement*

*En préambule à ce mémoire mes remerciant ALLAH qui m'a aidé et m'a donné la patience et le courage durant ces longues années d'étude.*

*Ce mémoire n'aurait pas été possible sans l'intervention, consciente, d'un grand nombre de personnes. Je souhaite ici les en remercier. Je tiens d'abord à remercier très chaleureusement **DR SAHOUR ABDELHAKIM** qui m'a permis de bénéficier de son encadrement. Les conseils qu'il m'a prodigué, la patience, la confiance qu'il m'a témoignée a été déterminants dans la réalisation de notre travail. Mes remerciements s'étendent également à tous nos enseignants durant les années des études.*

*Sans oublier mes parents pour leur prière et leur douaa .*

*Enfin, je tiens à remercier tous mes amis ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.*

# المخلص

في هذا المشروع قمنا بتطوير تطبيق إلكتروني داخل في اختصاص الطب عن بعد يسمح بمراقبة المرضى عن بعد وفي المنزل وخاصة الذين يعانون من أمراض مزمنة. هذا التطبيق يمكنه أخذ إشارات من جسم الإنسان هذه الأخيرة تترجم تطورات حالة الجسم مثل ذلك إشارات القلب ECG، الضغط، السكري... الخ من الأمراض المزمنة وتحويلها إلى نظام معلوماتي قادر على عرضها لحظيا للطبيب المعين.

قمنا بعرض مختلف الجوانب الخاصة بتكنولوجيا المعلومات والاتصالات التي صادفناها في مشوارنا الدراسي

انقسم هذا المشروع إلى قسمين رئيسيين قسم أجهزة وتم فيه استعمال مختلف الأجهزة الإلكترونية مثل نظام أردوينو، الشبكات اللاسلكية، المستشعرات الطبية والقسم الآخر قسم برمجي استعملنا فيه البرمجة ممثلة في برمجة بطاقة أردوينو وصفحات الويب بالإضافة إلى تخصيص الخادم وإعداده .

**الكلمات المفتاحية :** مراقبة طبية عن بعد ، نظام معلومات، بطاقة أردوينو، المستشعرات الطبية، شبكات لاسلكية، إرسال البيانات ، لغة البرمجة ، صفحات الويب .

# Résumé

Dans ce projet, nous avons développé une application électronique dans le domaine de la télé surveillance médicale à domicile qui permet de surveiller à distance des patients, en particulier ceux qui souffrent de maladies chroniques. Cette application peut prendre des paramètres physiologiques du corps humain ces derniers donne un aperçu sur l'état du corps tels que les signaux cardiaques ECG, la tension artérielle, le diabète ... etc et les transmettes vers un système d'information capable de les présenter au médecin instantanément.

Dans ce projet on a exploité les différents aspects de la technologie de l'information et de la télécommunication que nous avons acquis dans notre parcours d'études.

Ce projet a été divisé en deux sections principales. Une section hardware, dans laquelle divers outils électroniques, tels que le système Arduino, les réseaux sans fils wifi, les capteurs biomédicaux ont été utilisés, et dans l'autre section dit software, ont a utilisés la programmation représentée par la programmation des cartes et des pages Web et la configuration du serveur pour servir le projet et enfin les systèmes d'information.

**Mot clé :** télésurveillance médicale, système d'information, carte Arduino, capteurs biomédicaux, réseaux sans fils, envoie des donnés, langage de programmation, pages web

# Abstract

In this project we have developed an electronic application in the field of medical telemonitoring at home that allows remote monitoring of patients, especially those who suffer from chronic illnesses. This application can take physiological parameters of the human body these latter gives a glimpse into the state of the body such as ECG cardiac signals, blood pressure, diabetes ... etc and transmit it to an information system able to present them to the doctor instantly.

In this project we have used the various aspects of information technology and telecommunications that we have acquired in our course of study

This project was divided into two main sections. A hardware section, in which various electronic tools, such as the Arduino system, wireless WiFi networks, biomedical sensors have been used, and in the other section called software, have used programming represented by programming card and Web pages and configuring the server to serve the project and finally the information systems.

**Keyword:** medical telemonitoring, information system, Arduino card, biomedical sensors, wireless networks, send data, programming language, Web pages

# Sommaire

## Liste des figures

## Liste des tableaux

## Introduction générale

I

### CHAPITRE I : Généralité sur le système Arduino

I.1) Introduction.....	(1)
I.2) Généralité sur les microcontrôleurs.....	(1)
I.3) Définition du système Arduino.....	(2)
I.4) Propriétés de système Arduino.....	(3)
I.5) Configuration de base d'une carte Arduino.....	(3)
I.6) Les gammes des cartes Arduino.....	(4)
I.7) Les accessoires de la carte Arduino.....	(5)
I.7.1) Le module Arduino Bluetooth.....	(5)
I.7.2) Le module ESP8266.....	(6)
I.7.3) Le kit Arduino WIFI.....	(6)
I.7.4) Le kit Arduino GSM.....	(7)
I.8) Partie programme (software).....	(7)
I.8.1) Présentation du logiciel.....	(9)
I.8.1.1) Les boutons.....	(10)
I.8.2) La structure d'un programme Arduino.....	(12)
I.8.3) Coloration syntaxique.....	(12)
I.8.4) La syntaxe du langage.....	(13)
I.8.4.1) Ponctuation.....	(13)
I.8.4.2) Les variables.....	(13)
I.8.4.3) Les fonctions.....	(14)
I.8.4.4) Les structures de contrôle.....	(14)
I.9) Généralité sur les réseaux WIFI.....	(16)
I.9.1) Définition des réseaux sans fils.....	(16)
I.9.2) Techniques de transmission dans les réseaux sans fils.....	(16)
I.9.3) Les technologies sans fils.....	(17)
I.10) Conclusion.....	(17)

### Chapitre II : Généralité sur les capteurs et les capteurs biomédicaux

II.1) Introduction.....	(19)
II.2) Généralité sur les capteurs.....	(19)
II.3) Modes de fonctionnement des capteurs.....	(20)
II.3.1) Les capteurs actifs.....	(20)
II.3.2) Les capteurs passifs.....	(21)
II.4) Choix d'un capteur.....	(21)
II.5) La chaîne de mesure.....	(22)
II.6) Capteurs intelligents.....	(23)
II.7) Les capteurs biomédicaux.....	(24)
II.8) Technologie de mesure biomédicale.....	(25)
II.8.1) Le capteur de la saturation pulsée en oxygène(SpO2).....	(26)

II.8.2) Le capteur d'électrocardiogramme ECG.....	(27)
II.8.3) Le capteur de température.....	(28)
II.8.4) Le capteur de pression artérielle.....	(29)
II.8.5) Le capteur de position et les chutes.....	(29)
II.9) Conclusion.....	(31)

### **Chapitre III : La réalisation pratique de l'application de télémédecine**

III.1) Introduction.....	(33)
III.2) Architecture général du système.....	(33)
III.2.1) Partie client (patient).....	(34)
A) Description du matériel.....	(34)
A.1) La carte Arduino UNO.....	(35)
A.1.1) La nécessité d'utiliser Arduino UNO.....	(35)
A.1.2) La structure général de la carte Arduino UNO.....	(37)
A.1.2.1) Microcontrôleur ATMEL ATMEGA 328.....	(37)
A.1.2.2) L'alimentation de la carte.....	(39)
A.1.2.3) Les entrées et sorties.....	(41)
A.2) Le module ESP-8266.....	(42)
A.3) Le capteur DHT11.....	(45)
A.3.1) Caractéristiques détaillées.....	(46)
A.3.1.1) Alimentation et broche.....	(46)
A.3.1.2) Processus de communication « Interface série ».....	(46)
A.3.1.3) Processus de communication globale.....	(47)
B) Description du programme.....	(47)
B.1) Programme partie client.....	(47)
B.1.1) Partie déclarative.....	(49)
B.1.2) Partie vérification du module.....	(50)
B.1.3) Recherche et connexion a WIFI.....	(51)
B.1.4) Vérification du capteur.....	(51)
B.1.5) Ouverture du port et l'envoi des donnés.....	(51)
B.2) Partie serveur (coté superviseur médecin).....	(52)
B.2.1) Présentation de serveur local XAMPP.....	(52)
B.2.1.1) Définition d'un serveur local.....	(52)
B.2.2) Premier pas avec XAMPP.....	(53)
B.2.3) La programmation de la page de réception.....	(54)
B.2.4) La programmation de la page d'affichage.....	(54)
III.3) La réalisation pratique.....	(55)
III.4) Test d'application.....	(60)
III.5) Conclusion.....	(62)
<b>Conclusion générale</b>	(64)
<b>Bibliographie</b>	

## Liste des figures

<b>Figure</b>	<b>Nom de figure</b>	<b>page</b>
Figure I.1	Les microcontrôleurs	1
Figure I.2	Architecture generale d'un système Arduino	2
Figure I.3	Ensembles des cartes Arduino	3
Figure I.4	Arduino Mini	4
Figure I.5	Arduino Nano	4
Figure I.6	Arduino LilyPad	4
Figure I.7	Arduino Mega	5
Figure I.8	Arduino Uno	5
Figure I.9	Arduino Bluetooth	6
Figure I.10	Le module ESP8266-07	6
Figure I.11	Kit wifi d'Arduino	7
Figure I.12	Kit GSM d'Arduino	7
Figure I.13	carte connecté et alimentée	8
Figure I.14	Interface de programmation Arduino	8
Figure I.15	Présentation des parties principales du logiciel	9
Figure I.16	Présentation des boutons	10
Figure I.17	le programme de test BLINK	11
Figure I.18	La structure d'un programme Arduino	12
Figure II.1	Exemple d'évolution d'un mesurande m et de la réponses correspondante du capteur.	20
Figure II.2	Exemples de constitution de chaînes de mesure	23
Figure II.3	Capteur intelligent	23
Figure II.4	principe de fonctionnement d'un capteur SpO2	26
Figure II.5	Capteur SpO2	27
Figure II.6	Capteur ECG	27
Figure II.7	Positionnement des électrodes	28
Figure II.8	Le capteur de température	28
Figure II.9	Capteur de pression artérielle	29
Figure II.10	les positions de corps	30
Figure III.1	Architecture générale d'une application télémédecine	33
Figure III.2	Schéma globale de notre application	34
Figure III.3	présentation de la carte Arduino UNO	35

## Liste des figures

---

Figure III.4	Schéma simplifié de la carte Arduino UNO	37
Figure III.5	Microcontrôleur ATmega 328	38
Figure III.6	Brochage du microcontrôleur ATMEL ATmega 328	38
Figure III.7	Schéma d'alimentation de la carte Arduino UNO	39
Figure III.8	Les entrées et sortie de la carte Arduino UNO	41
Figure III.9	Bloc diagramme du module ESP-8266	43
Figure III.10	Module ESP8266	43
Figure III.11	Capteur DHT11	46
Figure III.12	Processus de communication globale d'un capteur DHT11	47
Figure III.13	L'organigramme du programme	48
Figure III.14	Logo de XAMP	53
Figure III.15	L'interface de XAMP	54
Figure III.16	Montage prototypage	55
Figure III.17	Schéma électronique du montage	56
Figure III.18	Photo de montage fini	56
Figure III.19	L'envoi du programme vers la carte	57
Figure III.20	La réponse d'exécution de la commande AT+RST	58
Figure III.21	Test des commandes de connexion	59
Figure III.22	Test des commandes d'envoi	60
Figure III.23	Les résultats obtenus dans la page web	61
Figure III.24	Représentation graphique des résultats	61

## Liste de tableaux

---

Tableau	Nom de tableau	page
<b>Tableau III.1</b>	paramètre de module ESP8266	45
<b>Tableau III.2</b>	Le rapport de la programmation de la carte	57

# Introduction Générale

## **Introduction générale.**

### **1. Généralité :**

La télésurveillance médicale (ou télévigilance) des patients atteints de maladies chroniques à domicile est une application majeure de la télémédecine [1]. Elle vise le maintien des personnes souffrant de maladies chroniques ou de personnes dites « fragiles » dans leurs vies quotidiennes. 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 chronique exemple la neurodégénérative (maladie d'Alzheimer par exemple). Ou tous simplement n'importe quel sujet qui nécessitent une surveillance médical.

La télésurveillance médicale présente aussi des enjeux politiques de la santé publique puisqu'elle répond au souhait des patients de réduire les temps d'hospitalisation, tout en promettant des économies d'échelles par la prise en charge extrahospitalière.

En même temps, notre vision de la télévigilance n'est plus de « surveiller » mais de « veiller sur » les personnes prises en charge. Il s'agit donc de mesurer et contrôler à distance des paramètres essentiels pour le diagnostic de la bonne santé physique, éventuellement mentale, du sujet en respectant les règles de confidentialité, de retour d'information vers la personne suivie (accès aux informations mais aussi intelligibilité de cette information), mais aussi de maintien de l'intégrité des informations pendant leur transfert et/ou leur traitement automatisés.

Les principales applications en télémédecine sont :

- La téléconsultation et le télédiagnostic : consultation médicale à distance ;
- La télésurveillance ou surveillance à distance d'un patient ;
- La télé expertise : avis donné à distance par un expert ou un médecin ;
- La téléformation : consultation des informations médicales
- La création de réseau de télémédecine : transmission des dossiers [2].

La télésurveillance médicale est devenue un élément important dans les systèmes médicaux pour les personnes âgées, le postopératoire et les maladies chroniques.

Parmi ses bénéfices, on cite [2] :

- Amélioration de la sécurité des patients ;
- Interventions des médecins plus rapides en cas de besoin ;
- Suivi immédiat et réconfort après un événement ;

---

## Introduction Générale

---

- Diminution du nombre de jours d'hospitalisation ;
- Diminution du cout grâce au suivi qui peut être fait en dehors de l'hôpital ;
- Réponse aux besoins des patients qui habitent les zones rurales isolées .

Au tout début de la chaîne d'information, les capteurs et instruments servent à acquérir les signaux physiologiques « généraux » (poids, tension artérielle, fréquence cardiaque [Fc]) ou spécifiques à une pathologie (taux de glycémie, taux d'oxygène dans le sang, ultrafiltration. . .). Ils permettent aussi de prendre en compte les conditions environnementales (température, hygrométrie, luminosité,...) du sujet puisque le domicile n'est plus « sous contrôle » comme c'est le cas d'une chambre hospitalière. Ils permettent enfin de qualifier, dans son environnement familial, les déplacements et mouvements de la personne, ce qui nous renseigne sur ses aptitudes et dépenses physiques, mais aussi sur l'accomplissement des principales activités de la vie quotidienne (lever, coucher, prises des repas, élimination. . .).[3]

Au milieu de la chaîne d'information, l'électronique programmable plus l'informatique joue un rôle indispensable ce rôle consiste à transmettre les données reçus par les capteurs a travers les réseaux sans fils qui représente un bonne solution pour la transmission des données instantanément et avec une qualité de service souhaitable vers le réseau internet spécifiquement au réseau d'hôpital ou une clinique spécial.

A la fin de la chaîne d'information le domaine de l'informatique doit être présent par un serveur qui héberge des pages web font un rôle de la réception et l'affichage de l'information reçus sous forme soit des courbes comme il est le cas de régime cardiaque et le changement suivait de la température ou la tension artérielle cette représente facilite le travail du superviseur et jette un œil sur la situation de son patient.

### **2. position du problème :**

L'augmentation spectaculaire du taux de mortalité des personnes à cause de maladies chroniques dépasse 63% on 2015 selon les statistiques de l'Organisation mondiale de la santé OMS [4] et avec l'évolution significative de la population mondiale ce taux est susceptible d'augmenter, en outre, beaucoup d'autres facteurs augmentent la probabilité que ce ratio augmentera par exemple, pas exclusivement, le grand temps perdu dans les files d'attente pour le diagnostic.

C'est ce qui nous a fait penser à une application électronique joue un rôle important de réduire le taux de mortalité élevé et le temps perdu dans les fils d'attentes et permettre de

---

## Introduction Générale

---

suivi l'évolution d'état du patient à distance sans intervenir, sauf dans les situations d'urgence, donnant au patient une plus grande liberté de vivre dans sa communauté et sa famille.

### **3. objectif du projet :**

Dans ce projet trois objectifs ont été visés :

- Le premier est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement en plus des informations sur les capteurs biomédicaux
- Le deuxième consiste à chercher sur les serveurs locaux qui ne donne une solution pour appliquer notre application en plus la collecte d'information sur la programmation des pages web .
- Le troisième est d'assurer le fonctionnement de l'application par la confirmation de connexion de différentes parties de l'application et l'envoi des informations à travers le réseau.

#### **Présentation du mémoire :**

Le premier chapitre sera consacré à donner des généralités sur le système Arduino. Puis en exposant la configuration de base des 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 réalisation de notre application, nous commençons par donner une vue globale du circuit, la présentation des différents éléments du système. Par la suite on présentera une explication détaillée de l'algorithme 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 I:

Généralité sur le système Arduino

## I.1) Introduction :

Le développement continu dans le domaine de l'électronique moderne est devenu dépendant à l'électronique programmée dit système embarqué ou informatique embarquée. Considérant que le but principal de l'électronique programmée est de simplifier les schémas électroniques et réduire l'utilisation de composants électroniques et donc fiabiliser le circuit. Et aussi réduire le coût de la réalisation d'un produit. bien que l'électronique câblée est très rapide, puisque il est réalisées par une voie matérielle. Il est peu économique quand l'application est complexe peu de souplesse : durée d'étude prohibitif et circuit difficilement modifiable. [5]

Aujourd'hui l'électronique programmée est devenue accessible à toutes personnes en ayant l'envie. Dans ce chapitre nous parlerons en général sur les systèmes électroniques qui utilisent les microcontrôleurs dans notre cas on a choisis l'environnement Arduino comme une solution pour développer notre application de télémédecine. Puisque les cartes Arduino donnent vraiment un potentiel de création quasi infini pourvu de disposer du matériel approprié. Il est possible de fabriquer des robots, de gérer des caméras, de commander des moteurs etc.

## I.2) Généralité sur les microcontrôleurs

Un microcontrôleur (MCU en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interface entrées-sorties.

Les microcontrôleurs sont utilisés dans les systèmes embarqués car ils permettent de diminuer la taille, la consommation électrique et le coût des produits. [6]

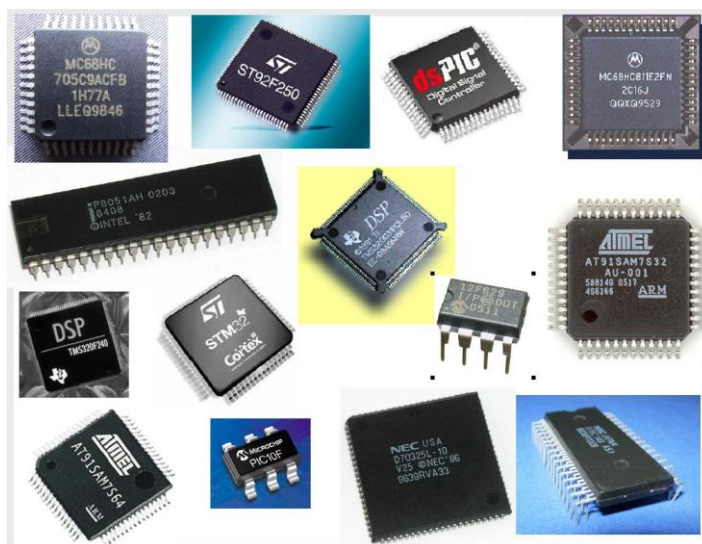


Figure I.1 Les microcontrôleurs

### I.3) Définition du système Arduino :

Le système Arduino est une plateforme open source basée sur une carte programmée à microcontrôleur et un environnement de développement permettant d'écrire, compiler et tester un programme. Les cartes et modules Arduino proposent une interface d'entrées-sorties simples qui peuvent recevoir des signaux de capteurs ou interrupteurs et peuvent commander des moteurs, éclairages, etc. Les projets Arduino peuvent être autonomes ou communiquer avec un PC. Le logiciel de développement open-source peut être téléchargé gratuitement sur [www.arduino.org](http://www.arduino.org) et qui permet de programmer les cartes Arduino en C++. [7]

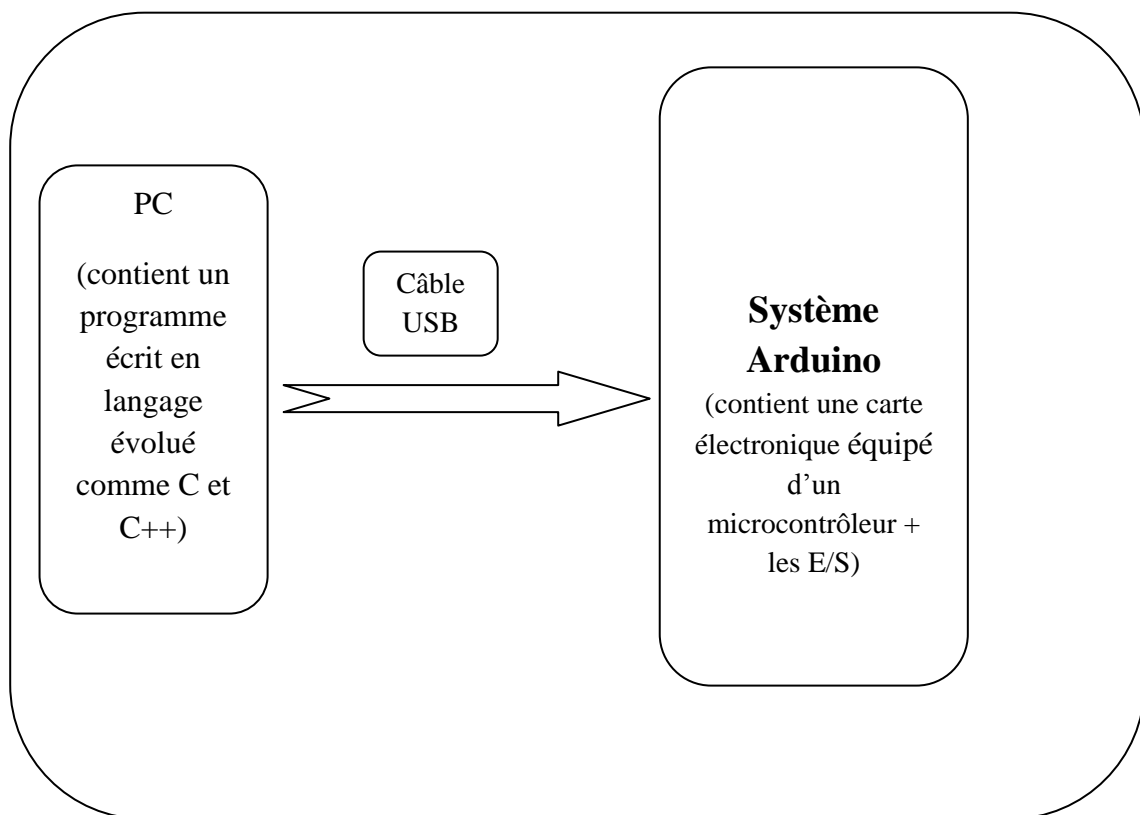


Figure I.2 Architecture generale d'un système Arduino



Figure I.3 Ensembles des cartes Arduino

## I.4) Propriétés de système Arduino :

- **Open source** : La désignation open source, ou « code source ouvert », s'applique aux logiciels (et s'étend maintenant aux œuvres de l'esprit) dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. Mis à la disposition du grand public, ce code source est généralement le résultat d'une collaboration entre programmeurs. [8]
- **Open hardware** : L'Open Hardware est un terme qui désigne les matériels dont la conception peut être reproduite, car toutes les informations nécessaires à leur construction sont en libre accès. Une ouverture qui est finalement l'équivalent de l'Open Source, mais non plus appliquées aux lignes de code d'un logiciel, mais aux plans de conception des objets. [9]

## I.5) Configuration de base d'une carte Arduino :

Un module Arduino est généralement construit de :

- Un microcontrôleur ATMEL AVR (ATmega328, ATmega168 ou ATmega8).
- Des composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits.
- Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz.

### I.6) Les gammes des cartes Arduino :

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique: [10]

- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.



Figure I.4 Arduino Mini

- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).

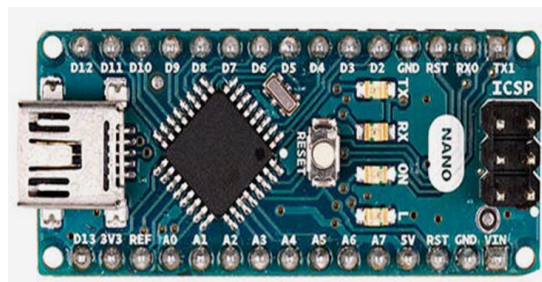


Figure I.5 Arduino Nano

- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.

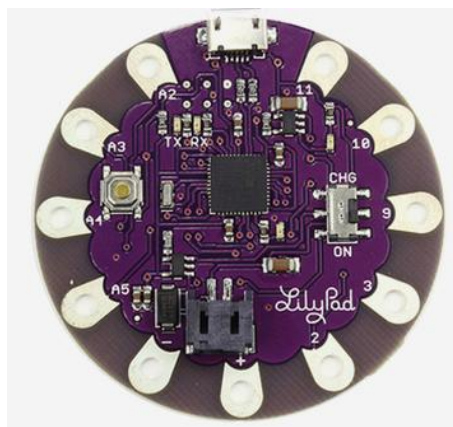


Figure I.6 Arduino LilyPad

- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.



**Figure I.7 Arduino Mega**

- L'Arduino UNO, utilisations microcontrôleur ATmega328.



**Figure I.8 Arduino Uno**

### **I.7) Les modules d'extension de la carte Arduino**

Les cartes Arduino généralement sont associées a des modules externe qui simplifient les réalisations. Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

#### **I.7.1) Le module Arduino Bluetooth :**

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion sérielle 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.[11]



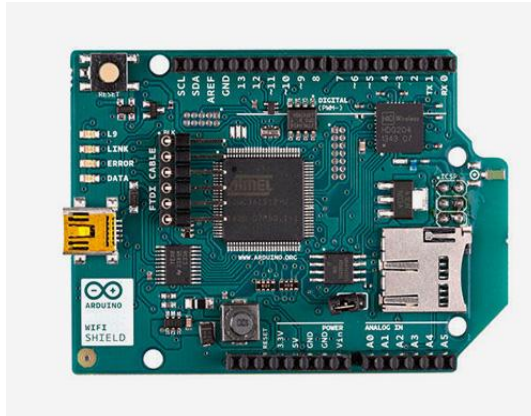


Figure I.11 Kit wifi d'Arduino

### I.7.4) Le kit Arduino 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. [14]



Figure I.12 Kit GSM d'Arduino

### I.8) Partie programme (software) :

La plupart des cartes électroniques programmables qui se basent sur un microcontrôleur doivent avoir une interface de programmation.

L'environnement de programmation open source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux) dans le site officiel <https://www.arduino.cc/en/Main/Software>.

Après l'installation et une suite de clignotement sur les micro-LED de la carte, celle-ci devrait être fonctionnelle; une petite LED verte témoigne de la bonne alimentation de la carte: [15]



Figure I.13 carte connecté et alimentée

La fenêtre du logiciel :

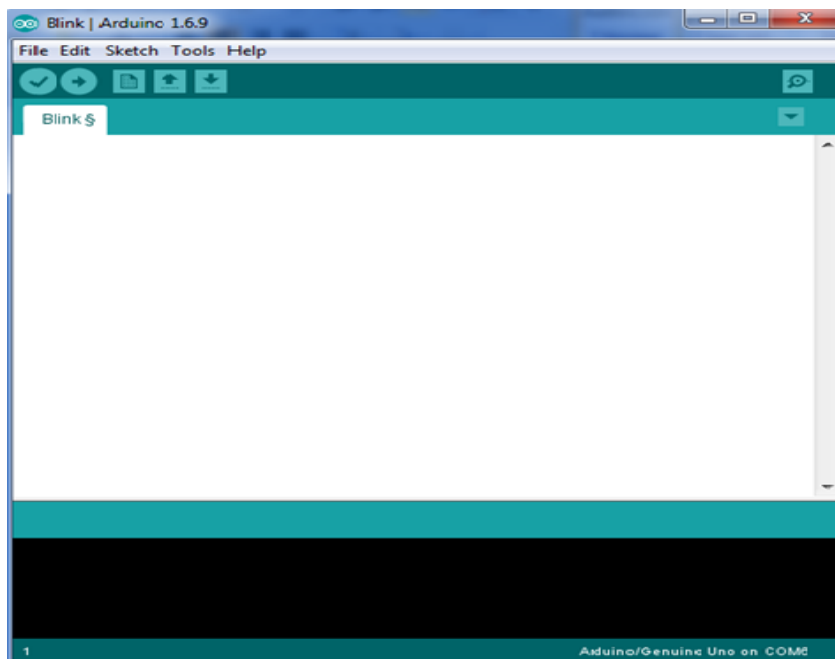


Figure I.14 Interface de programmation Arduino

## I.8.1) Présentation du logiciel

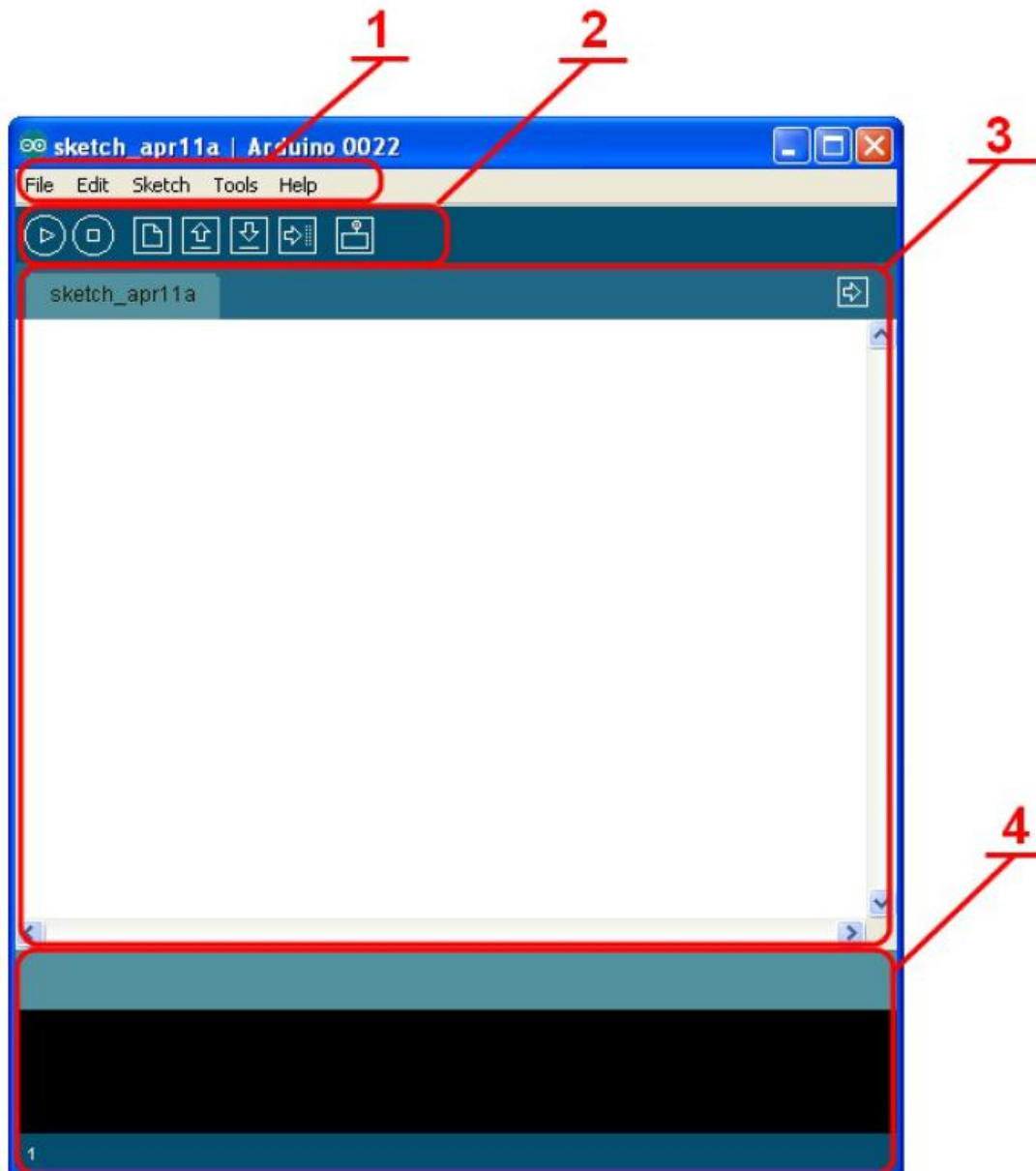


Figure I.15 Présentation des parties principales du logiciel

- **Le cadre numéro 1** : ce sont les options de configuration du logiciel
- **Le cadre numéro 2** : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes
- **Le cadre numéro 3** : ce bloc va contenir le programme que nous allons créer
- **Le cadre numéro 4** : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le **débogueur**.

### I.8.1.1) Les boutons :

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par les chiffres.

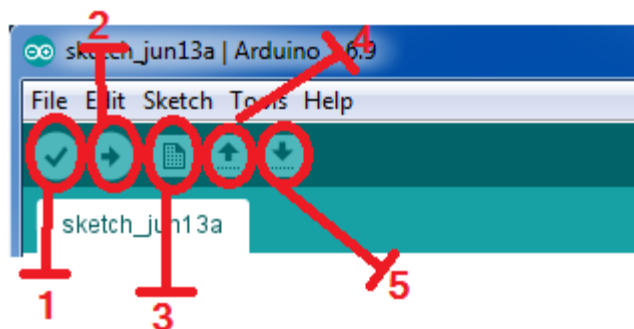
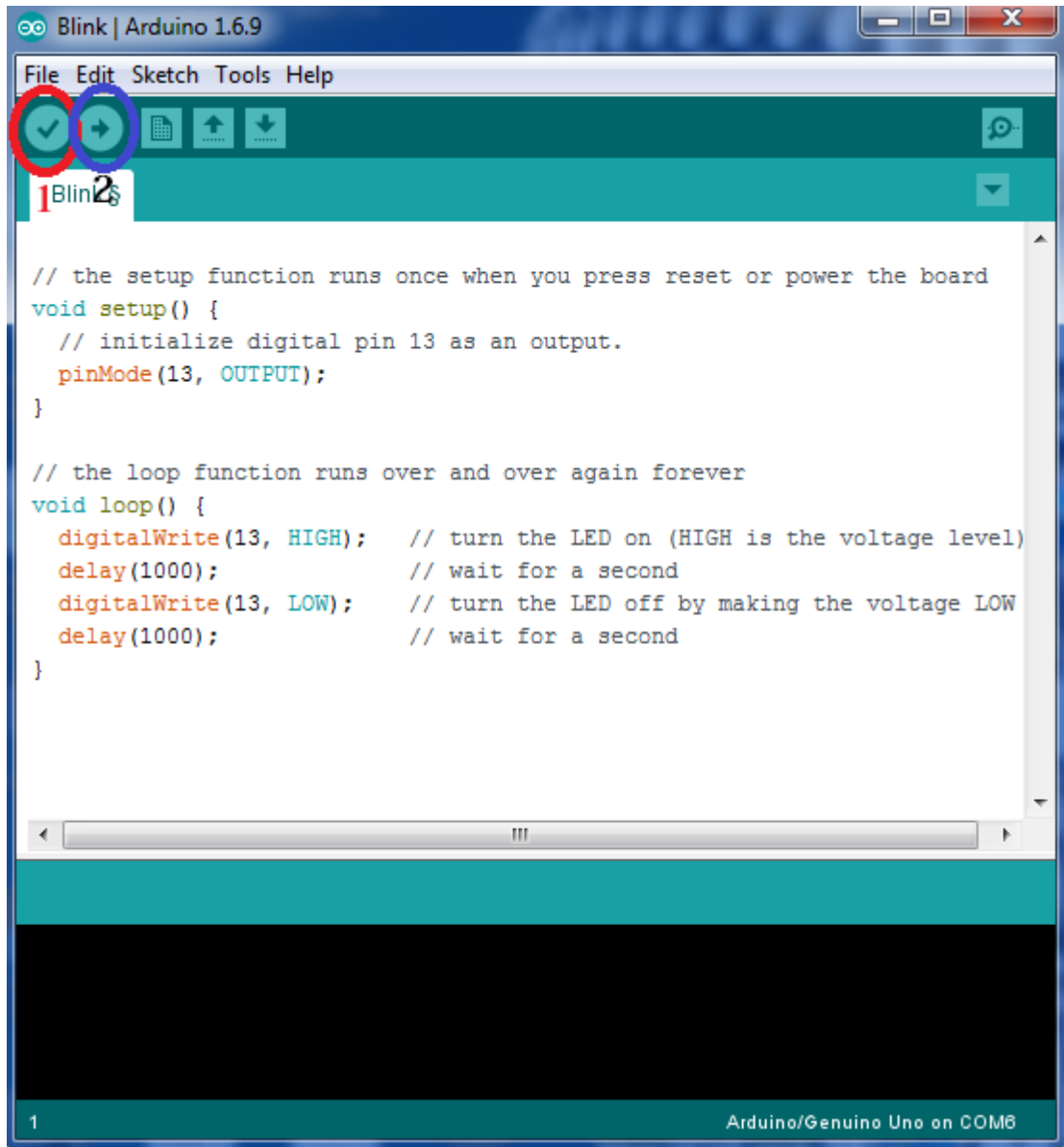


Figure I.16 Présentation des boutons

- **Bouton 1** : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- **Bouton 2** : Ce bouton permet de vérifier et envoyer le programme à la carte.
- **Bouton 3** : Ouvrir un nouveau projet.
- **Bouton 4** : Ouvrir un programme déjà existe.
- **Bouton 5** : Sauvegarder le programme en cours.

La figure I.17 représente un exemple de programme de test **BLINK LED** qui fait clignoter la LED de la carte connectée à la broche 13 ce programme est généralement utiliser a la première fois pour tester le bon fonctionnement de la carte .



**Figure I.17 le programme de test BLINK**

Pour vérifier la syntaxe de programme on click sur le bouton 1(Verify) et pour transférer le programme a la carte Arduino on click sur le bouton 2(UPLOAD).

## I.8.2) La structure d'un programme Arduino :

Un programme Arduino comporte trois parties: [16]



```
Button $  
  
// set pin numbers:  
const int buttonPin = 2;    // the number of the pushbutton pin  
const int ledPin = 13;     // the number of the LED pin  
// variables will change:  
int buttonState = 0;       // variable for reading the pushbutton status  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize the pushbutton pin as an input:  
  pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}
```

Figure I.18 La structure d'un programme Arduino

1. la partie déclaration des variables (optionnelle)
2. la partie initialisation et configuration des entrées/sorties : la fonction **setup ()**
3. la partie principale qui s'exécute en boucle : la fonction **loop ()**

## I.8.3) Coloration syntaxique

Lorsque du code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

En **orange**, apparaissent les mots-clés reconnus par le langage Arduino comme des **fonctions** existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic

---

## Chapitre I : Généralité sur le système Arduino

---

avec le bouton droit de la souris, l'on a la possibilité de choisir « Find in reference » : cette commande ouvre directement la documentation de la fonction sélectionnée. [16]

En **bleu**, apparaissent les mots-clés reconnus par le langage Arduino comme des **constantes**. [16]

En **gris**, apparaissent les **commentaires** qui ne seront **pas exécutés dans le programme**. Il est utile de bien commenter le code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes :

- dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire.
- l'on peut encadrer des commentaires sur plusieurs lignes entre « /\* » et « \*/ ». [17]

### I.8.4) La syntaxe du langage

#### I.8.4.1) Ponctuation

Le code est structuré par une ponctuation stricte : [17]

- **toute ligne** de code se termine par un point-virgule « ; »
- le contenu d'une **fonction** est délimité par des accolades « { » et « } »
- les **paramètres** d'une fonction sont contenus pas des parenthèses « ( » et « ) ».

Une erreur fréquente consiste à oublier un de ces éléments.

#### I.8.4.2) Les variables

Une variable est un espace réservé dans la mémoire de l'ordinateur. C'est comme un compartiment dont la taille n'est adéquate que pour un seul type d'information. Elle est caractérisée par un nom qui permet d'y accéder facilement. [18]

Il existe différents types de variables identifiés par un mot-clé dont les principaux sont :

- nombres entiers (int)
- nombres à virgule flottante (float)
- texte (String)
- valeurs vrai/faux (boolean).

Un nombre à décimales, par exemple *3.14159*, peut se stocker dans une variable de type float. Notez que l'on utilise un point et non une virgule pour les nombres à décimales. Dans Arduino, il est nécessaire de déclarer les variables pour leurs réserver un espace mémoire adéquat.

---

## Chapitre I : Généralité sur le système Arduino

---

On déclare une variable en spécifiant son type, son nom puis en lui assignant une valeur initiale (optionnelle). Exemple :

```
int ma_variable = 45;
// int est le type, ma_variable le nom et = 45 assigne une valeur.
```

### I.8.4.3) Les fonctions

Une fonction (également désignée sous le nom de procédure ou de sous-routine) est un bloc d'instructions que l'on peut appeler à tout endroit du programme.

Le langage Arduino est constitué d'un certain nombre de fonctions, par exemple analogRead(), digitalWrite() ou delay().[18]

Il est possible de déclarer ses propres fonctions par exemple :

```
void clignote(){
    digitalWrite (brocheLED, HIGH) ;
    delay (1000) ;
    digitalWrite (brocheLED, LOW) ;
    delay (1000) ;
}
```

### I.8.4.4) Les structures de contrôle

Les structures de contrôle sont des blocs d'instructions qui s'exécutent en fonction du respect d'un certain nombre de conditions.

Il existe quatre types de structure: [19]

***if...else*** : exécute un code **si** certaines conditions sont remplies et éventuellement exécutera un autre code avec **sinon**.

#### **Exemple :**

```
//si la valeur du capteur dépasse le seuil
if(valeurCapteur > seuil){
    //appel de la fonction clignote
    clignote(); }
```

## Chapitre I : Généralité sur le système Arduino

---

**while** : exécute un code **tant** que certaines conditions sont remplies.

### Exemple :

```
//tant que la valeur du capteur est supérieure à 250
while(valeurCapteur > 250){
    //allume la sortie 5
    digitalWrite(5,HIGH);
    //envoi le message "0" au port serie
    Serial.println(1);
    //en boucle tant que valeurCapteur est supérieure à 250
}
Serial.println(0);
digitalWrite(5,LOW);
```

**For** : exécute un code **pour** un certain nombre de fois.

### Exemple :

```
//pour i de 0 à 255, par pas de 1
for (int i=0; i <= 255; i++){
    analogWrite(PWMPin, i);
    delay(10);
}
```

**switch/case** : fait un choix entre plusieurs codes **parmi une liste de possibilités**

### Exemple :

```
// fait un choix parmi plusieurs messages reçus
switch (message) {
    case 0: //si le message est "0"
        //allume que la sortie 3
        digitalWrite(3,HIGH);
        digitalWrite(4,LOW);
        digitalWrite(5,LOW);
```

```
break;
case 1: //si le message est "1"
    //allume que la sortie 4
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    break;
case 2: //si le message est "2"
    //allume que la sortie 5
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
    break;
}
```

### **I.9) Généralité sur les réseaux WIFI**

#### **I.9.1) Définition des réseaux sans fil**

Un réseau sans fil est un réseau dans lequel au moins deux terminaux sont capables de communiquer entre eux grâce à des signaux radioélectriques. Les réseaux sans fil ne sont pas tout récents, mais avec le développement de l'informatique et des systèmes d'information, la technologie est venue au besoin primaire de l'homme : la mobilité et la facilité. Ces réseaux dits « sans-fil » sont de plusieurs sortes : WiFi (Wireless Fidelity), BlueTooth, BLR (Boucle Locale Radio), UMTS (Universal Mobile Télécommunications System). [20]

#### **I.9.2) Techniques de transmission dans les réseaux sans fil**

Il existe principalement deux méthodes pour la transmission dans les réseaux sans fil: [21]

- **Transmission par les ondes infrarouges**

La transmission par les ondes infrarouges nécessite que les appareils soient en face l'un des autres et aucun obstacle ne sépare l'émetteur du récepteur.(car la transmission est directionnelle).cette technique est utilisée pour créer des petits réseaux de quelques dizaines de mètres. (Télécommande de : télévision, les jouets, voitures...).

- **Transmission par les ondes radios.**

La transmission par les ondes radios est utilisée pour la création des réseaux sans fil qui a plusieurs kilos mètres. Les ondes radios ont l'avantages de ne pas êtres arrêtés par les obstacles car sont émises d'une manière omnidirectionnelle. Le problème de cette technique est perturbations

extérieurs qui peuvent affecter la communication à cause de l'utilisation de la même fréquence par exemple.

### **I.9.3) Les technologies sans fil**

Les technologies dites « sans fil », la norme 802.11 en particulier, facilitent et réduisent le coût de connexion pour les réseaux de grande taille. Avec peu de matériel et un peu d'organisation, de grandes quantités d'informations peuvent maintenant circuler sur plusieurs centaines de mètres, sans avoir recours à une compagnie de téléphone ou de câblage. Ces technologies peuvent être classées en quatre parties: [22]

- Les réseaux personnels sans fil : Wireless Personal Area Network (WPAN)
- Les réseaux locaux sans fil : Wireless Local Area Network (WLAN)
- Les réseaux métropolitains sans fil : Wireless Metropolitan Area Network (WMAN)
- Les larges réseaux sans fil : Wireless Wide Area Network (WWAN)

### **I.10) Conclusion**

Dans ce chapitre, nous avons projeté la lumière sur le système Arduino en donnant ainsi les propriétés pour les quelles on l'a choisie, puis nous avons cité les différents types des cartes de ce système ainsi les kits utiliser pour la communication. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément enfin nous avons donné quelque généralité sur les réseaux wifi.

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

# Chapitre II:

Généralité sur les capteurs  
et les capteurs biomédicaux

### II.1) Introduction :

Dans le domaine médical, Les capteurs jouent un rôle de jours aux jours plus important, ils peuvent être utilisés dans des systèmes pour sauver des vies par l'assurance d'une surveillance permanente des organes vitaux de l'être humain. Ils sont disposés sur la peau ou implantés dans le corps humains pour faciliter le diagnostic des maladies par la mesure continue d'un phénomènes ou plus (exemple le suivit des battements cardiaque, de la tension artérielle, de la fonction respiratoire, etc).

Dans ce chapitre on donne, en premier temps, des généralités sur les capteurs ainsi leur mode de fonctionnement soit actif ou passif ensuite on a cité quelque paramètre qui distinguent le bon choix du capteur, dans un deuxième temps nous avons parlé sur les capteurs biomédicaux leur prospérité et leur classification (physique, chimique ou biologique).

### II.2) Généralités sur les capteurs

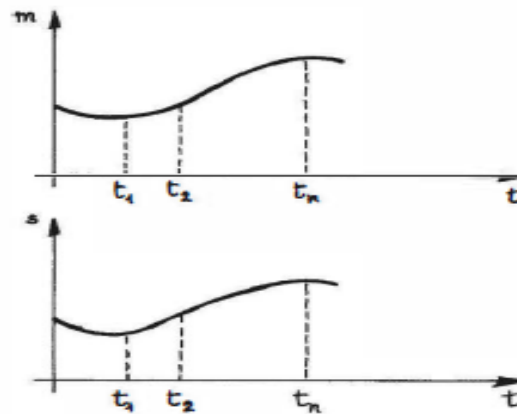
La grandeur physique objet de la mesure : déplacement, température, pression, etc. est désignée comme le mesurande et représentée par  $\mathbf{m}$  ; l'ensemble des opérations expérimentales qui concourent à la connaissance de la valeur numérique du mesurande constitue son mesurage. Lorsque le mesurage utilise des moyens électroniques de traitement du signal, il est nécessaire de produire à partir du mesurande une grandeur électrique qui en soit une représentation aussi exacte que possible : ceci signifie que la grandeur électrique et ses variations apportent toute l'information nécessaire à la connaissance du mesurande. Le capteur est le dispositif qui soumis à l'action d'un mesurande non électrique présente une caractéristique de nature électrique (charge, tension, courant ou impédance) désignée par  $\mathbf{s}$  et qui est fonction du mesurande :

$$\mathbf{S} = \mathbf{F}(\mathbf{m})$$

$\mathbf{S}$  est la grandeur de sortie ou réponse du capteur,  $\mathbf{m}$  est la grandeur d'entrée ou excitation. La mesure de  $\mathbf{S}$  doit permettre de connaître la valeur de  $\mathbf{m}$

La relation  $\mathbf{S} = \mathbf{F}(\mathbf{m})$  résulte dans sa forme théorique des lois physiques qui régissent.





**Figure II.1 Exemple d'évolution d'un mesurande m et de la réponses correspondante du capteur.**

Pour des raisons de facilité d'exploitation on s'efforce de réaliser le capteur, ou du moins de l'utiliser, en sorte qu'il établisse une relation linéaire entre les variations  $\Delta S$  de la grandeur de sortie et celles  $\Delta m$  de la grandeur d'entrée : [23]

$$\Delta S = S \cdot \Delta m$$

$S$  est la sensibilité du capteur.

Un des problèmes importants dans la conception et l'utilisation d'un capteur est la constance de sa sensibilité  $S$  qui doit dépendre aussi peu que possible :

- de la valeur de  $m$  (linéarité) et de sa fréquence de variation (bande passante)
- du temps (vieillessement) ;
- de l'action d'autres grandeurs physiques de son environnement qui ne sont pas l'objet de la mesure et que l'on désigne comme grandeurs d'influence.

En tant qu'élément de circuit électrique, le capteur se présente, vu de sa sortie :

- soit comme un générateur,  $S$  étant une charge, une tension ou un courant et il s'agit alors d'un capteur actif;
- soit comme une impédance,  $S$  étant alors une résistance, une inductance ou une capacité : le capteur est alors dit passif.

### **II.3) Modes de fonctionnement des capteurs**

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

#### **II.3.1) Les capteurs actifs**

Un capteur actif est un capteur pour lequel la grandeur de sortie est une tension ou un courant électrique. Cette grandeur est directement utilisable par l'instrumentation de mesure.

L'énergie physique prélevée sur la mesurande est transformée directement en énergie électrique qui constitue le signal de sortie. Ce signal se présente sous la forme d'un courant ou d'une tension. Lors de la phase de traitement, ce signal est filtré (nettoyé), amplifié et converti dans un format compatible et exploitable. Ainsi, le capteur doit non seulement mesurer des propriétés physiques mais doit également effectuer des tâches additionnelles au travers de circuits de traitement et de communication intégrés. [24]

### II.3.2) Les capteurs passifs

Un capteur passif est un capteur qui ne fournit pas d'énergie. Sa grandeur électrique de sortie est généralement une impédance qui varie avec la grandeur à mesurer. Afin de pouvoir obtenir un signal utilisable, ce type de dispositif nécessite la mise en oeuvre d'un conditionneur qui transforme ces variations en une différence de potentiel ou en un courant. Ce type de capteur est utilisé dans des applications spécifiques qui nécessitent des unités de mesure de grande précision. [24]

### II.4) 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: [25]

- **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.5) La chaîne de mesure :**

La chaîne de mesure est constituée de l'ensemble des dispositifs, y compris le capteur, rendant possible, dans les meilleures conditions, la détermination précise de la valeur du mesurande.

À l'entrée de la chaîne, le capteur soumis à l'action du mesurande permet, directement s'il est actif ou par le moyen de son conditionneur s'il est passif, d'injecter dans la chaîne le signal électrique, support de l'information liée au mesurande.

À la sortie de la chaîne, le signal électrique qu'elle a traité est converti sous une forme qui rend possible la lecture directe de la valeur cherchée du mesurande : [23]

- Déviation d'un appareil à cadre mobile ;
- Enregistrement analogique graphique ou oscillographique ;
- Affichage ou impression d'un nombre.

C'est l'étalonnage de la chaîne de mesure dans son ensemble qui permet d'attribuer à chaque indication en sortie la valeur correspondante du mesurande agissant à l'entrée.

Sous sa forme la plus simple la chaîne de mesure peut se réduire au capteur, et à son conditionneur éventuel, associé à un appareil de lecture :

- Thermocouple et voltmètre ;
- Jauge de contrainte placée dans un pont de Wheatstone, avec pour instrument de lecture un galvanomètre ou un voltmètre.

Pendant les conditions pratiques de mesure telles qu'elles sont imposées par l'environnement d'une part et par les performances exigées pour une exploitation satisfaisante du signal d'autre part amènent à introduire dans la chaîne des blocs fonctionnels destinés à optimiser l'acquisition et le traitement du signal :

- Circuit de linéarisation du signal délivré par le capteur ;
- Amplificateur d'instrumentation ou d'isolement destiné à réduire les tensions parasites de mode commun ;
- Multiplexeur, amplificateur d'instrumentation programmable, échantillonneur
- Bloqueur, convertisseur analogique - numérique lorsque l'information doit être traitée par ordinateur ;
- convertisseur tension-courant ou tension-fréquence lorsque le signal doit être transmis à distance par câble
- modulateur de fréquence dans le cas de télémesure par voie hertzienne.

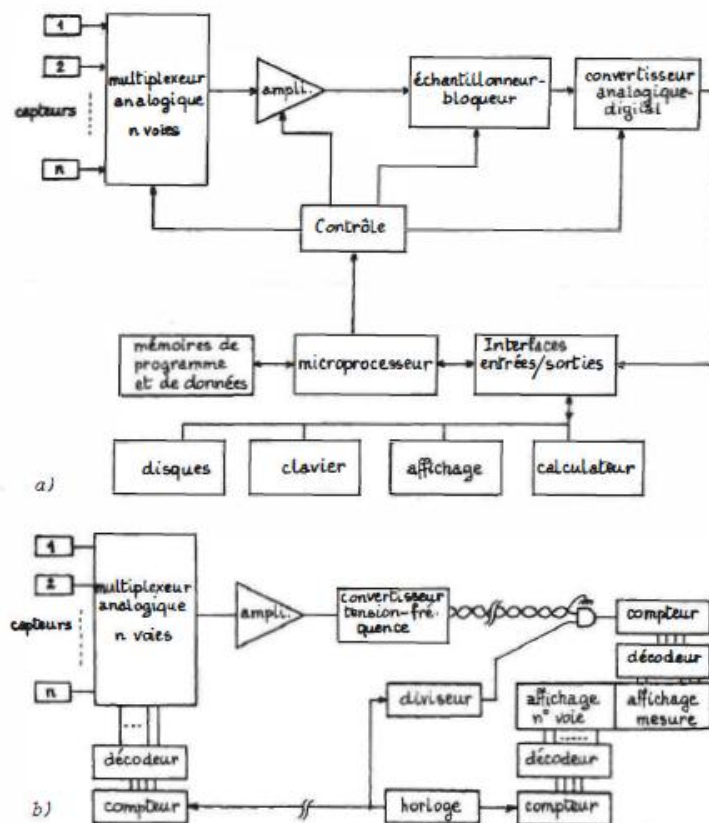


Figure II.2 Exemples de constitution de chaînes de mesure :  
 a) chaîne contrôlée par microprocesseur, b) chaîne avec conversion tension-fréquence des signaux permettant leur transmission bifilaire.

## II.6) Capteurs intelligents

On désigne par capteur intelligent l'ensemble de mesure d'une grandeur physique constitué de deux parties : [26]

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

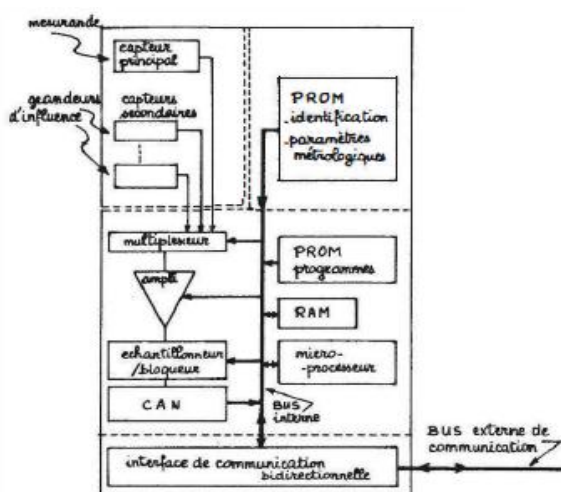


Figure II.3 Capteur intelligent

La chaîne de mesure comporte :

- Le capteur principal spécifique du mesurande étudié, et identifiable par un code stocké en PROM (Programmable Read On/y 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.7) Les capteurs biomédicaux**

Les capteurs biomédicaux sont des capteurs destinés à recueillir les signaux physiologiques générés par le patient. Ils créent un dialogue entre le patient et la machine. Ils doivent obéir aux critères suivant: [27]

- 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.

Les capteurs biomédicaux peuvent être classés dans les catégories suivantes en fonction de leurs quantités de détection. Classés par principe de travail, les capteurs comprennent des capteurs physiques, des capteurs chimiques et des capteurs biologiques.[28]

- 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.

- 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.
- 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.8) 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 liée 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.[28]

Par ailleurs, on rencontre deux types de capteurs ; les capteurs invasifs et les capteurs non invasifs. 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. A l'inverse, les capteurs non 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.[29]

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

- Le capteur de La saturation pulsée en oxygène (SpO<sub>2</sub>)
- Le capteur d'électrocardiogramme ECG
- Capteur de température
- Capteur de pression artérielle
- Capteur de position et les chutes

### II.8.1) Le capteur de La saturation pulsée en oxygène (SpO<sub>2</sub>) :

C'est un capteur qui utilise une méthode de mesure non invasive de la saturation en oxygène de l'hémoglobine au niveau des capillaires sanguins.

L'oxymètre (saturomètre) émet deux types de lumière ; une lumière rouge et infrarouge et va mesurer l'absorption lumineuse de l'hémoglobine. Il faut savoir que l'oxyhémoglobine absorbe l'infrarouge et laisse passer la lumière rouge alors que la désoxyhémoglobine absorbe la lumière rouge et laisse passer l'infrarouge. En fonction de ça, la partie lumineuse non absorbée est recueillie par le capteur photoélectrique et analysée. Exemple, une SpO<sub>2</sub> de 98% = chaque globule rouge est chargé à 98% d'oxyhémoglobine et de 2% de désoxyhémoglobine.

L'oxymétrie de pouls permet donc de mesurer la proportion d'hémoglobine totale combinée à l'oxygène. [30]

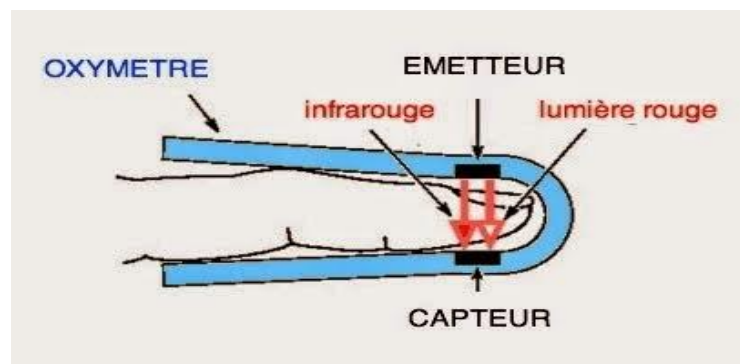


Figure II.4 principe de fonctionnement d'un capteur SpO<sub>2</sub>



**Figure II.5 Capteur SpO<sub>2</sub>**

### **II.8.2) Le capteur d'électrocardiogramme ECG :**

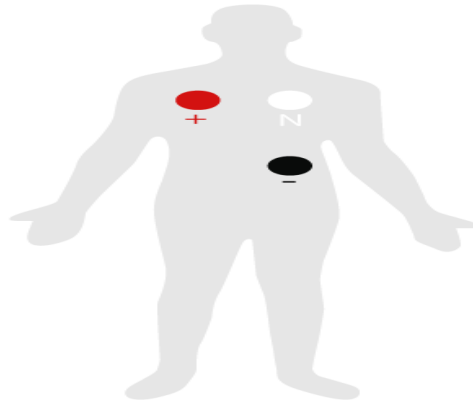
L'électrocardiogramme (ECG) est un outil de diagnostic couramment utilisé pour évaluer les fonctions électriques et musculaires du cœur.

Le capteur d'électrocardiogramme (ECG) est devenu l'un des tests médicaux les plus couramment utilisés en médecine moderne. Son utilité dans le diagnostic d'une myriade de pathologies cardiaques allant de l'ischémie myocardique et l'infarctus à la syncope et aux palpitations a été inestimable pour des cliniciens pendant des décennies.

La précision de l'ECG dépend de la condition testée. Un problème cardiaque peut ne pas toujours apparaître sur l'ECG. Certaines conditions cardiaques ne produisent aucun changement d'ECG spécifique. Les fils d'ECG sont attachés au corps tandis que le patient se trouve à plat sur un lit ou une table. [31]



**Figure II.6 Capteur ECG**



**Figure II.7 Positionnement des électrodes**

### **II.8.3) 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  $37^{\circ}\text{C}$  ( $98.6^{\circ}\text{F}$ ). 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. [32]



**Figure II.8 Le capteur de température**

### II.8.4) 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 la 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. [33]



**Figure II.9 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

### II.8.5) Capteur de position et les chutes

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

## Chapitre II : Généralité sur les capteurs et les capteurs biomédicaux

---

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.[33]

### Caractéristiques:

1,95 v à 3,6 v tension d'alimentation

1,6 v à 3,6 v tension d'interface

$\pm 2 \text{ g} / \pm 4 \text{ g} / \pm 8 \text{ g}$  dynamiquement sélectionnable à pleine échelle

Cet accéléromètre est emballé avec des fonctions intégrées avec des options programmables d'utilisateur flexibles, configurable à deux goupilles d'interruption. L'accéléromètre a des échelles sélectionnables par l'utilisateur de  $\pm 2 \text{ g} / \pm 4 \text{ g} / \pm 8 \text{ g}$  avec des données filtrées passe-haut et des données non filtrées disponibles en temps réel.

### Les positions de corps

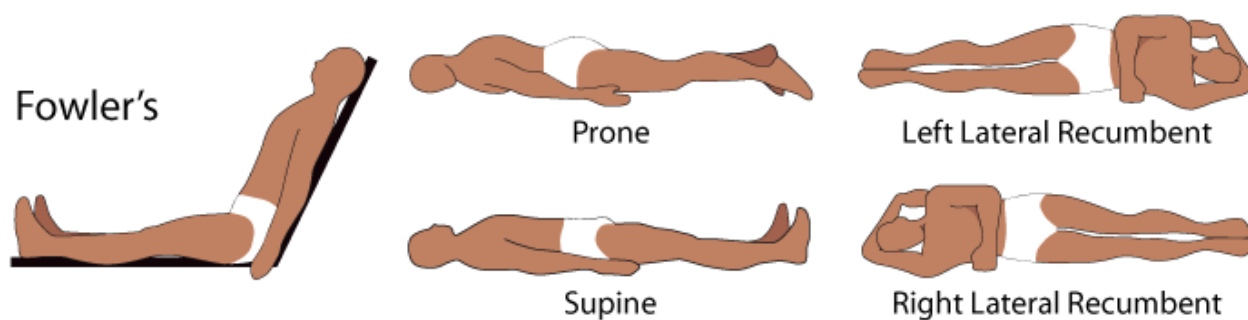


Figure II.10 les positions de corps

### **II.9) 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 III:

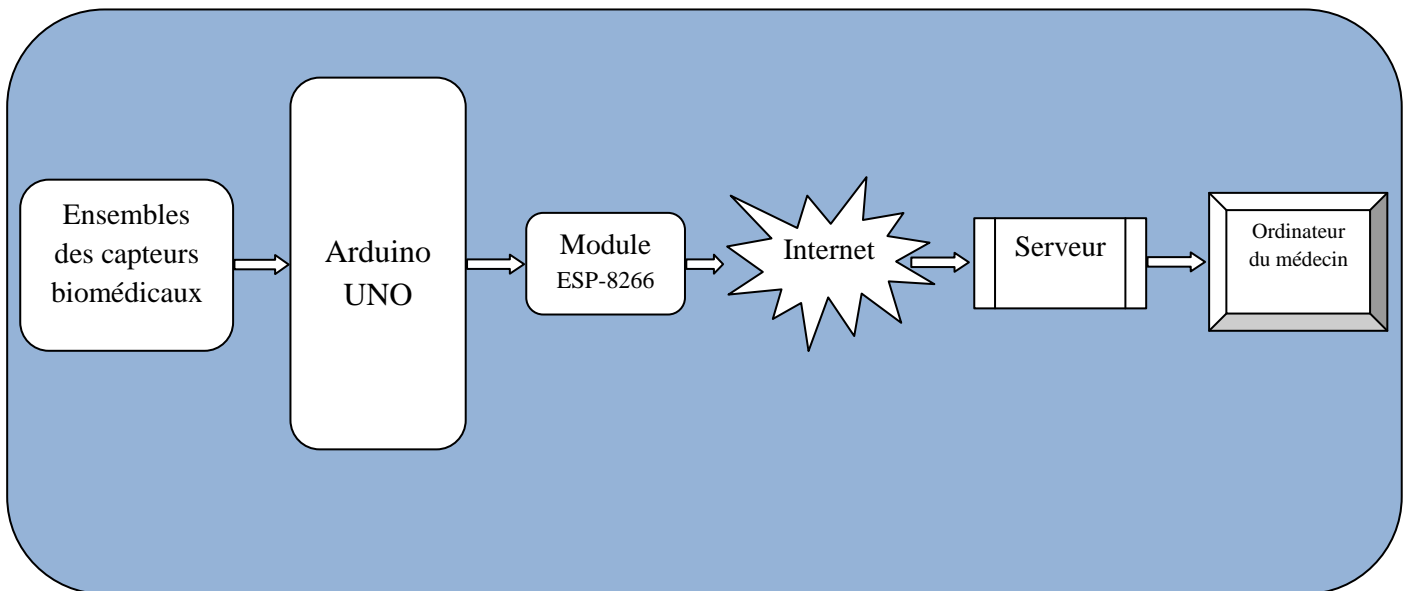
La réalisation pratique de l'application  
de télé-médecine

### III.1) Introduction :

Dans ce chapitre, nous abordons la méthodologie de la réalisation de notre application qui a comme objectif la télésurveillance des patients à distance, on utilisant les technologies de l'information et de la communication et de l'électronique embarquée. Pour atteindre nos objectifs une carte Arduino doit être disponible en plus d'un ensemble de capteurs biomédicaux compatibles avec cette carte plus d'un module wifi qui permet la connexion a un réseau d'internet sans fil afin de transmettre des données vers l'ordinateur distant du médecin.

### III.2) Architecture globale du système :

L'architecture générale de notre application est présentée sur la figure (III.1).



**Figure III.1 Architecture globale du système**

La solution proposée permet d'expérimenter la surveillance à domicile, à l'aide d'un ensemble de capteurs 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 principaux parties :

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

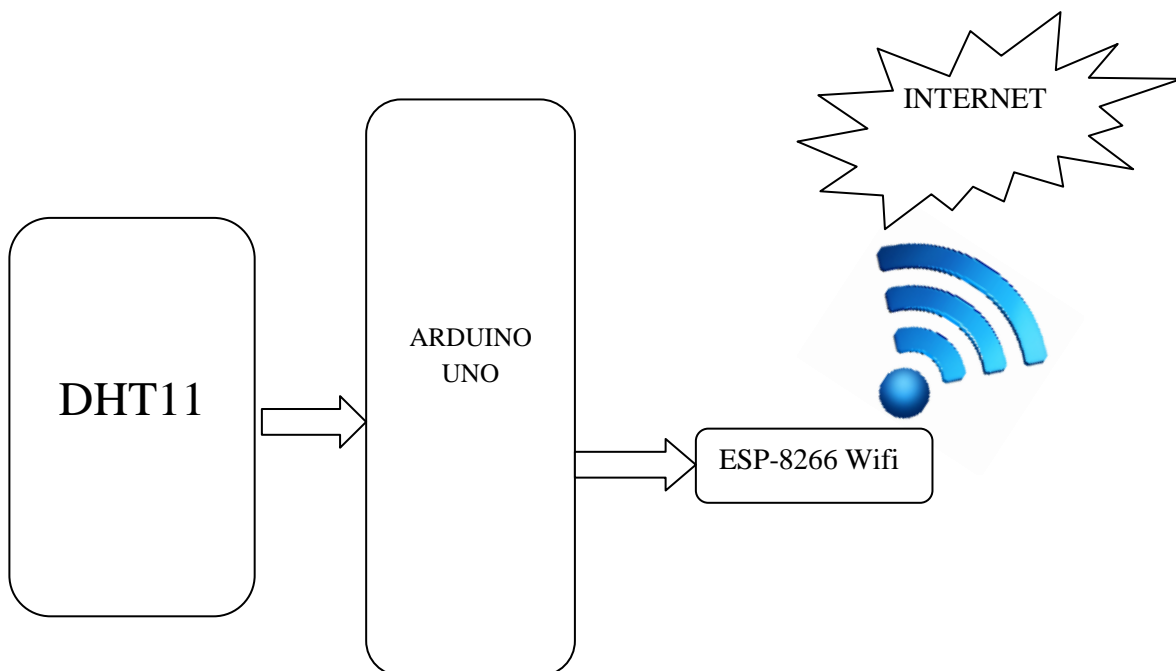
**Partie server (côté médecin) :** dans cette partie nous trouvons un ordinateur reliée a internet qui représente le server local (on a utilisé XAMP) il héberge un système d'information capable d'acquérir les données, les traiter et enfin les présenter sur un terminal au médecin superviseur.

### **III.2.1) Partie client (coté patient)**

#### **A) Description du HARDWARE**

Cette partie repose au tour d'une carte Arduino de type UNO, un ensemble des capteurs biomédicaux qui doivent être relié à cette carte. Ainsi qu'un module WIFI ESP-8266 pour assurer la connexion a un réseau internet sans fil WIFI.

Dans notre cas, et avec l'absence des moyens concernant les capteurs biomédicaux. Nous nous somme limité à l'utilisation d'un seul capteur DHT11 suite a sa disponibilité sur le marché local. Ce capteur dite d'humidité et de température est constitué d'un capteur de température à base de NTC et d'un capteur d'humidité résistif, un microcontrôleur s'occupe de faire les mesures, les convertir et de les transmettre. Notre réalisation est celle de la figure III.2



**Figure III.2 schéma globale de notre application**

### A.1) La carte Arduino UNO :

Le cerveau de notre application est la carte UNO de la société ARDUINO. C'est une carte électronique basée sur un microcontrôleur ATMEL de référence ATmega328, ce microcontrôleur est de la famille AVR a 8bits.

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

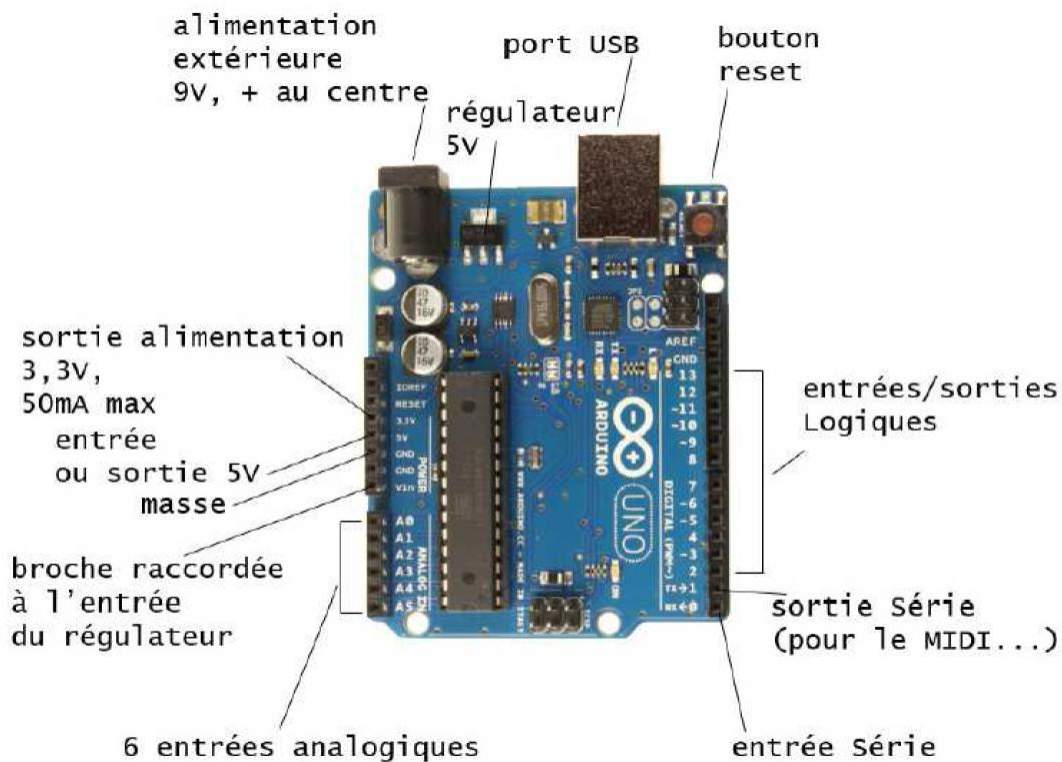


Figure III.3 présentation de la carte Arduino UNO

#### A.1.1) La nécessité d'utiliser Arduino UNO :

Le choix de la carte électronique approprié à notre travail dépend de plusieurs paramètres malgré qu'il existe plusieurs cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs ces derniers 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 aux personnes intéressés on peut étayer notre choix de la carte Arduino UNO par: [34]

- **Faible coût :** Les cartes Arduino sont relativement peu coûteuses par rapport aux autres plates-formes de microcontrôleurs. La version la moins chère du module Arduino peut être assemblée à la main, et même les modules Arduino pré-assemblés coûtent moins de 3000 DA.

- **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 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 (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série.

- **Matériel Open source et extensible :** les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et 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.

### A.1.2) la structure général de la carte Arduino UNO

Généralement le module Arduino est basé sur un microcontrôleur ATMEL AVR ATmega328 et des composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V ou 3.3V et un oscillateur à quartz 16 MHz .Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire. [35]

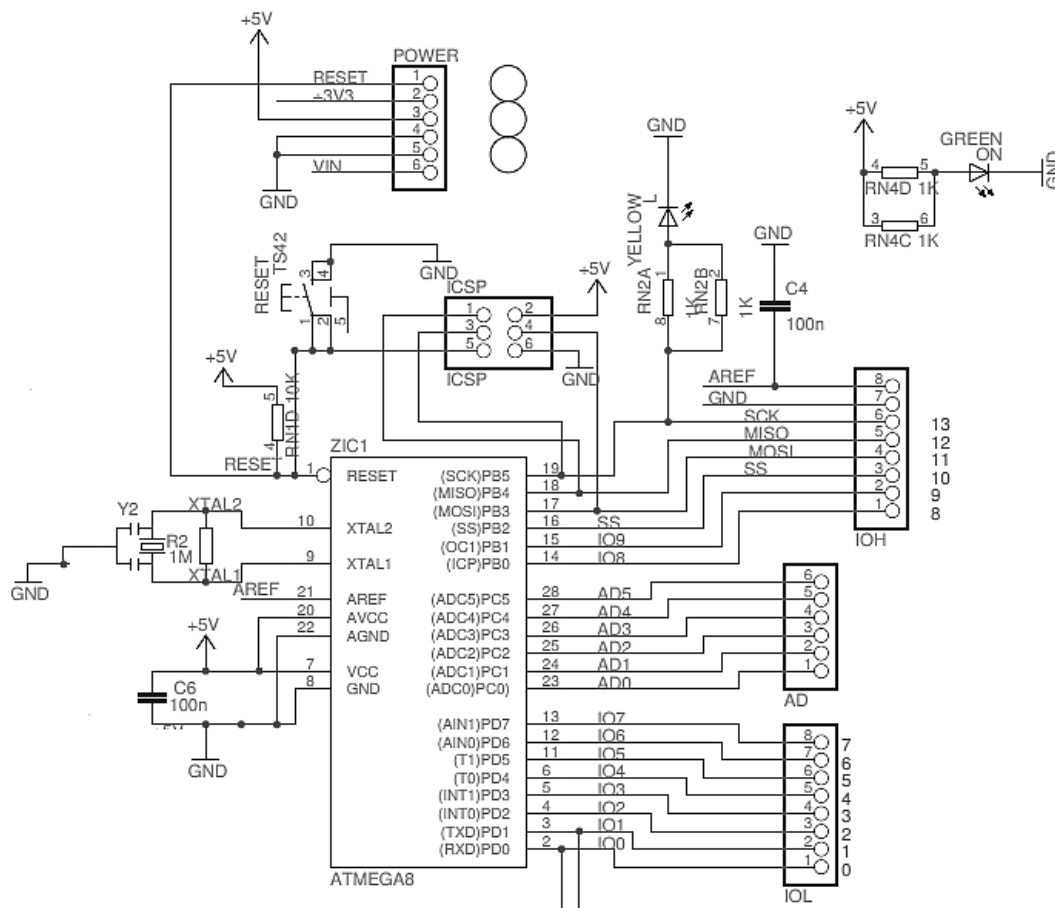


Figure III.4 schéma simplifié de la carte Arduino UNO

#### A.1.2.1) Microcontrôleur ATMEL ATMEGA328 :

Le modèle du microcontrôleur basé sur la carte Arduino UNO est ATMEGA328 de la famille AVR 8 bits [36].

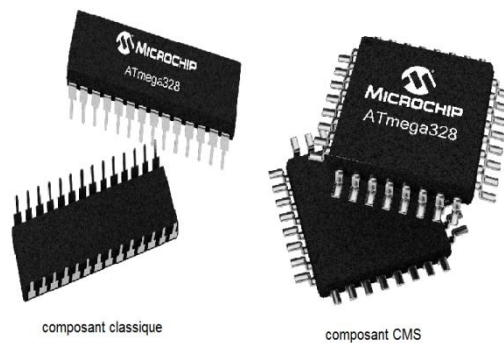


Figure III.5 microcontrôleur ATmega 328

Les principales caractéristiques sont :

**FLASH** = mémoire programme de 32Ko

**SRAM** = données (volatiles) 2Ko

**EEPROM** = données (non volatiles) 1Ko

**Digital I/O (entrées-sorties Tout Ou Rien) =**

3 ports Port B, Port C, Port D

(soit 23 broches en tout I/O)

**Timers/Counters** : Timer0 et Timer2

(Comptage 8 bits), Timer1 (comptage 16bits)

(PCINT14/RESET) PC8	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT8/XTAL1/TOSC1) PB8	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD8	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ( $\overline{SS}$ /OC1B/PCINT2)
(PCINT0/CLKO/CP1) PB0	14	15	PB1 (OC1A/PCINT1)

Figure III.6 Brochage du microcontrôleur ATMEL ATmega 328

Chaque timer peut être utilisé pour générer deux signaux PWM.

(6 broches OCxA/OCxB) broches peuvent avoir plusieurs fonctions différentes choisies par programmation

**PWM** = 6 broches **OC0A(PD6)**, **OC0B(PD5)**, **OC1A(PB1)**, **OC1B(PB3)**, **OC2A(PB3)**, **OC2B(PD3)**

**Analog to Digital Converter** (resolution 10bits) = 6 entrées multiplexées **ADC0 (PC0)** a **ADC5 (PC5)**

**Gestion bus I2C** (TWI Two Wire Interface) = le bus est exploite via les broches **SDA(PC5)/SCL(PC4)**.

**Port série (USART)** = émission/réception série via les broches **TXD(PD1)/RXD(PD0)**

**Comparateur Analogique** = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption

**Watchdog Timer programmable.**

### A.1.2.2) L'Alimentation de la carte :

Pour protéger la carte contre les hautes tensions il faut au contraire soigneusement étudier l'alimentation du module Arduino et des périphériques et autres dispositifs associés [37].

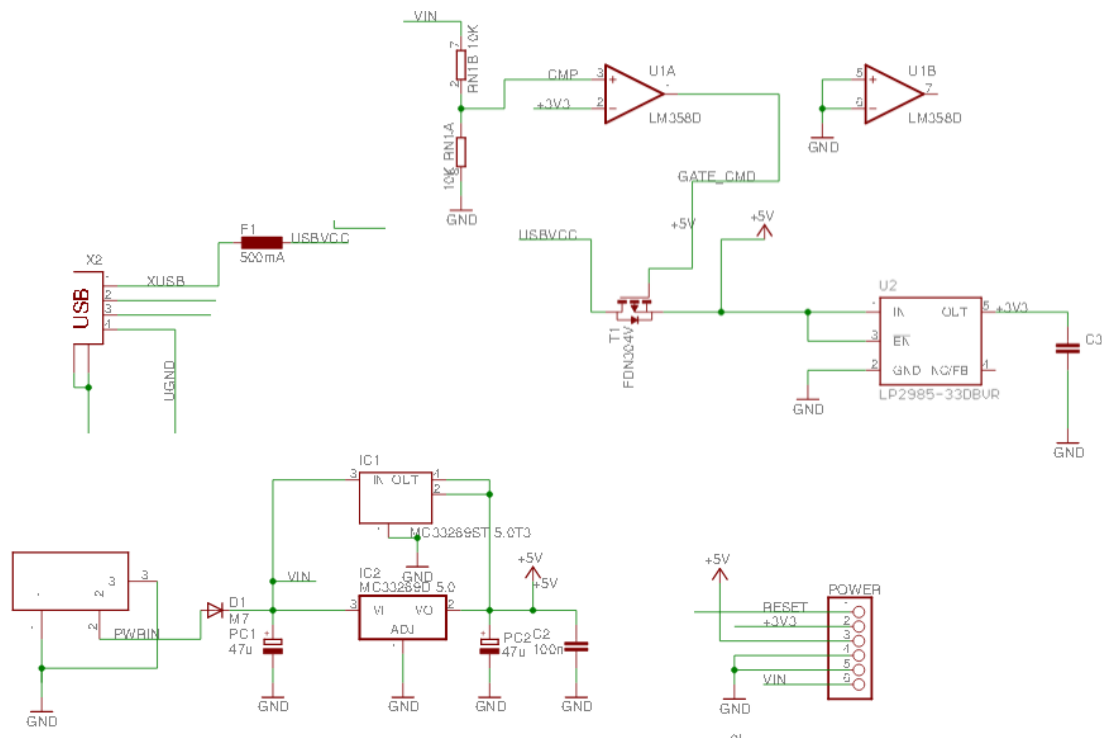


Figure III.7 schéma d'alimentation de la carte Arduino UNO

On a 3 possibilités pour alimenter la carte (qui demande du 5v):

- alimentation directe sur la broche **5v**: dans ce cas, il faut impérativement que le jack ne soit pas alimenté, mais l'USB peut l'être (utilisation de la transmission de données), le circuit constitué du comparateur **LM85 U1A** et du **MOSFET T1** évitera le court-circuit entre la tension **5v** du port USB et la tension mise sur la broche **5V**. Il faudra

protéger l'entrée contre les surtensions, inversions, transitoires (selon la longueur de la ligne d'alimentation, une paire de condensateurs chimiques/céramique, une self, une diode de clamping...). Aucun régulateur ne limite la puissance fournie par cette entrée 5v. Il faudra fournir sur cette broche l'intensité nécessaire:

- à l'alimentation de l'Arduino et aux entrées sorties,
  - aux shields alimentés par lui
  - au régulateur 3.3v
- alimentation sur la broche Vin: dans ce cas la broche 5v est alimentée par la tension Vin abaissée par le régulateur IC1/IC2. Là aussi, l'absence impérative de tension sur le jack permet au circuit comparateur MOSFET d'isoler l'USB. Par contre dans ce cas le régulateur 5V de l'Arduino est actif, et son échauffement limite l'intensité délivrable aux circuits alimentés; en reprenant les valeurs de l'article ci-dessous, il faudra limiter à:

- Alimentation 12 V:  $I = 2 / (12-5) = 2 / 7 = 285\text{mA}$
- Alimentation 9 V:  $I = 2 / (9-5) = 2/4 = 500\text{mA}$
- Alimentation 7 V:  $I = 2 / (7-5) = 2/2 = 1\text{A}$

- alimentation par le jack: ce cas est identique au cas précédent, à une petite différence prêt: une diode M7 (équivalent de 1N4007 en montage de surface) est placée pour protéger contre l'inversion de polarité, la tension nécessaire sera donc un peu plus élevée que pour l'alimentation directe de Vin. Sinon, le 5v sera toujours obtenu par le régulateur et les intensités acceptable seront les mêmes, en tenant compte toutefois de la puissance dissipée par la diode (grosso modo  $1 \times \text{Intensité}$ ).

- alimentation par le port USB: dans ce cas, l'ensemble des composants alimentés ne devront pas consommer plus de 500mA (fusible automatique sur la carte), à vérifier que l'appareil peut les fournir! Le 5v sera fourni directement, en passant par le MOSFET dont la résistance en mode conduction est faible; on se retrouvera donc dans les mêmes conditions que pour l'alimentation en direct sur le 5V, avec la limitation de l'intensité utilisable à 500mA.



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 PWM 8-bits à l'aide de l'instruction **analog Write ( )**.

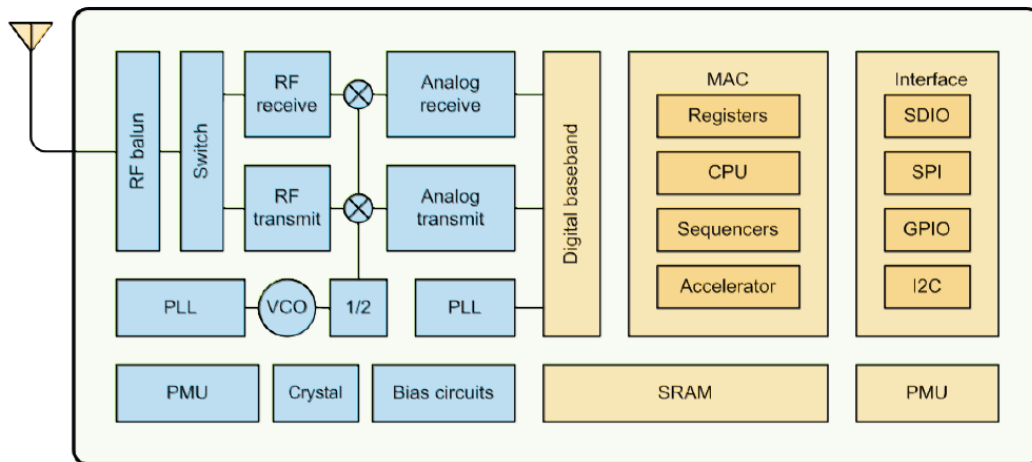
- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la [bibliothèque pour communication SPI](#). Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant [la bibliothèque 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 de 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.

**Note :** les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19

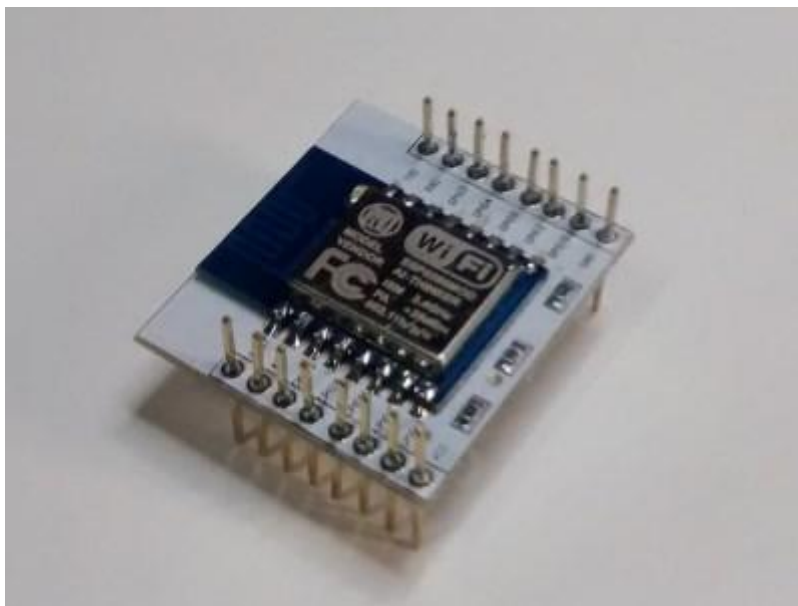
### A.2) Le module ESP-8266

Espressif Systems' Smart Connectivity Platform (ESCP) est un ensemble de hautes performances, sans fil haute intégration, conçues pour les concepteurs de plates-formes mobiles à espace et à puissance. Il offre une capacité inégalée à intégrer les capacités WIFI dans d'autres systèmes, ou à fonctionner comme une application autonome, avec le coût le plus bas, et l'espace minimal requis.



**Figure III.9 Bloc diagramme du module ESP-8266**

ESP8266 offre une solution de réseautage WIFI complète et autonome; Il peut être utilisé pour héberger l'application ou pour décharger les fonctions de réseau WIFI d'un autre processeur d'application. [39]



**Figure III.10 module ESP8266**

Lorsqu'ESP8266 héberge l'application, il démarre directement à partir d'un flash externe. Elle dispose d'un cache intégré pour améliorer les performances du système dans de telles applications.

Alternativement, elle sert comme un adaptateur WiFi, l'accès Internet sans fil peut être ajouté à n'importe quelle conception controllerbased avec une connectivité simple (SPI/SDIO ou interface I2C/UART).

ESP8266 est parmi les puces WiFi les plus intégrées de l'industrie; Il intègre les commutateurs d'antenne, RF Balun, amplificateur de puissance, amplificateur de réception à faible bruit, filtres, modules de gestion de l'alimentation, il nécessite un

## Chapitre III : La réalisation pratique de l'application de télémédecine

---

circuit externe minimal, et la solution entière, y compris le module frontal, est conçu pour occuper une zone de PCB minimale.

ESP8266EX intègre également une version améliorée du processeur L106 Diamond Series 32-bit de Tensilica, avec SRAM sur puce, en plus des fonctionnalités WiFi. ESP8266EX est souvent intégré avec des capteurs externes et d'autres dispositifs spécifiques d'application par son GPIO; les exemples de codes pour ces applications sont fournis dans le kit de développement logiciel (SDK).

On peut se servir du module pour : [39]

- **Créer un serveur Web**, pour visualiser les données d'un capteur sur un téléphone mobile ou un ordinateur via le navigateur Web ou bien pour lui envoyer des commandes comme fermer la porte du garage ou arroser les plantes (pour une utilisation un peu classique)
  - **Communiquer avec un serveur Web**, pour envoyer ou recevoir des données sur le Web, à partir d'un site.
  - **Créer un réseau maillé**, où chaque module envoie et reçoit des informations des autres.
  - **Créer des connexions sans fils**, et donc envoyer et recevoir des informations par le WiFi en passant par d'autres méthodes que le navigateur Web, par communications UDP/TCP/OSC par exemple, qui sont interfaçables avec beaucoup de langages de programmations et logiciels.

**Paramètre :**

<b>Catégories</b>	<b>Articles</b>	<b>Valeurs</b>	
<b>Parametre de wifi</b>	<b>Certificats</b>	FCC/Ce/Telec/CSRR	
	<b>WiFi protocles</b>	802,11 b/g/n	
	<b>Bande passante</b>	2.4 g-2.5 g	
	<b>Puissance de TX</b>		802,11 b: + 20 dBm
			802,11 g: + 17 dBm
			802,11 n: + 14 dBm

	<b>Sensibilité RX</b>	802,11 b:-91 dBm (11 Mbps)
		802,11 g:-75 dBm (54 Mbps)
		802,11 n:-72 dBm (MCS7)
	<b>Types d'antennes</b>	PCB trace, externe, connecteur IPEX, Copeaux de céramique
<b>Paramètre du HARD</b>	<b>Bus périphérique</b>	UART/SDIO/SPI/I2C/I2S/IR télécommande
		GPIO/PWM
	<b>Tension de fonctionnement</b>	3.0 ~ 3,6 v
	<b>Courant de fonctionnement</b>	80 mA
	<b>température de fonctionnement</b>	40 ° ~ 125 °
<b>Paramètre du SOFT</b>	<b>Mode WiFi</b>	station/softAP/SoftAP+station
	<b>Sécurité</b>	WPA/WPA2
	<b>Cryptage</b>	WEP/TKIP/AES
	<b>Protocoles réseau</b>	IPv4, TCP/UDP/http/FTP

**Tableau III.1 paramètre de module ESP8266**

Les principaux domaines d'applications ESP8266EX sont les suivants:

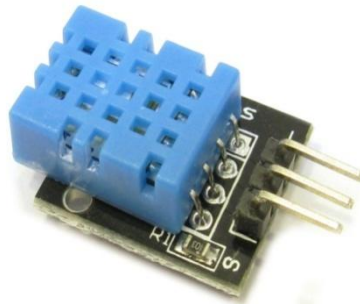
- Domotique
- Réseau maillé
- Contrôle sans fil industriel
- Moniteurs pour bébés
- Caméras IP
- Réseaux de capteurs
- Électronique portable

### **A.3) Le Capteur DHT11 :**

Ce capteur est dédié à la mesure de température et l'humidité avec une sortie de signal numérique calibrée en utilisant la technique exclusive « numérique-signal-acquisition » et la technique de détection de température & d'humidité, il assure la fiabilité élevée et une excellente stabilité à long terme, ce capteur comprend une

mesure de l'humidité du type résistif et un élément NTC de mesure de température, et se connecte à un microcontrôleur 8 bits haute performance, offrant d'excellente qualité, réponse rapide, capacité anti parasitage et la rentabilité.

Chaque élément DHT11 est strictement calibré dans le laboratoire qui est extrêmement précis sur l'étalonnage de l'humidité. Les coefficients d'étalonnage sont stockés sous forme de programmes dans la mémoire de l'OTP, qui servent de signal interne de la sonde de détection de processus. L'interface série du fil simple rend l'intégration du système facile et rapide. Sa petite taille, de faible consommation d'énergie et de transmission du signal vers le haut à 20 mètres ce qui en fait le meilleur choix pour diverses applications. [40]



**Figure III.11 Capteur DHT11**

### **A.3.1) Caractéristiques détaillées :**

- Alimentation : 5V
- Consommation : 0.5 mA en nominal / 2.5 mA maximum
- Etendue de mesure température : 0°C à 50°C ± 2°C
- Etendue de mesure humidité : 20-90%RH ±5%RH

#### **A.3.1.1) Alimentation et broche :**

La tension d'alimentation de DHT11 est 3-5.5V DC, lorsque le capteur est alimentée, n'envoyez pas n'importe quelle instruction du capteur en moins d'une seconde afin de transmettre le statut instable. Un condensateur d'une valeur 100nF peuvent être ajoutés entre VDD et GND pour le filtrage de l'alimentation.

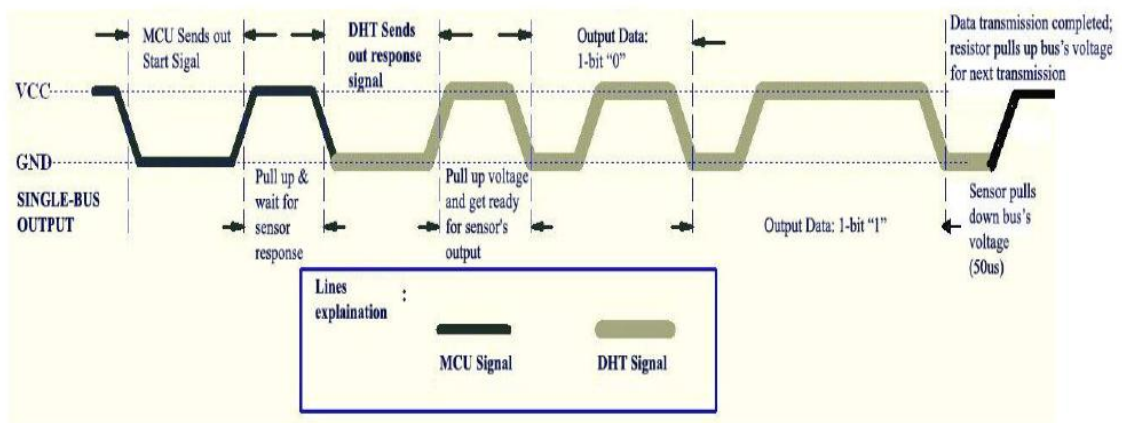
#### **A.3.1.2) Processus de communication : Interface série**

Un seul format bus de données est utilisé pour la communication et la synchronisation entre le capteur DHT11 et le MCU dans ce cas le mode de communication utilisé est HALF-DUPLEX.

Données se composent de parties décimales et intégrantes. Une transmission de données complètes est de 40 bits et le capteur envoie les données qui ont des bits plus élevés tout d'abord.

### A.3.1.3) Processus de Communication globale

Lorsque MCU envoie un signal de départ, DHT11 change son état de fonctionnement, en attendant que le MCU compléte le signal de départ. Une fois terminé, DHT11 envoie un signal de réponse de 40-bits de données qui incluent l'humidité et



la température vers MCU. Sans signal de départ de MCU, DHT11 ne donnera le signal de réponse à MCU

Figure III.12 Processus de communication globale d'un capteur DHT11

## B) Description du programme :

Notre application est une solution basée sur l'électronique programmée. Pour cela nous avons développé un ensemble de programmes qui seront implémentés dans les deux parties de l'application. Ces programmes ont les rôles suivants : la gestion du système, le traitement des données provenant des capteurs et les transmettent via internet et enfin leur traitement au niveau serveur pour qu'ils puissent être exploités par le superviseur qui est éventuellement le médecin traitant. L'ensemble des programmes sont implantés dans la carte ARDUINO qui représente le contrôleur principal de l'application, et sur le serveur.

### B.1) Programme partie client :

Dans cette partie nous avons développé un programme en langage ARDUINO qui est le langage de programmation des cartes ARDUINO comme il a été décrit auparavant. À partir de l'algorithme présenté sur la figure. Ce programme a pour rôle

principale est la gestion de la partie hardware, donc la carte Arduino, le module wifi et les différents capteurs physiologiques qui peuvent être connectés au système. La gestion de la communication entre les différents éléments et enfin l'acquisition des données à partir des capteurs et leur transmission vers le serveur distant via internet.

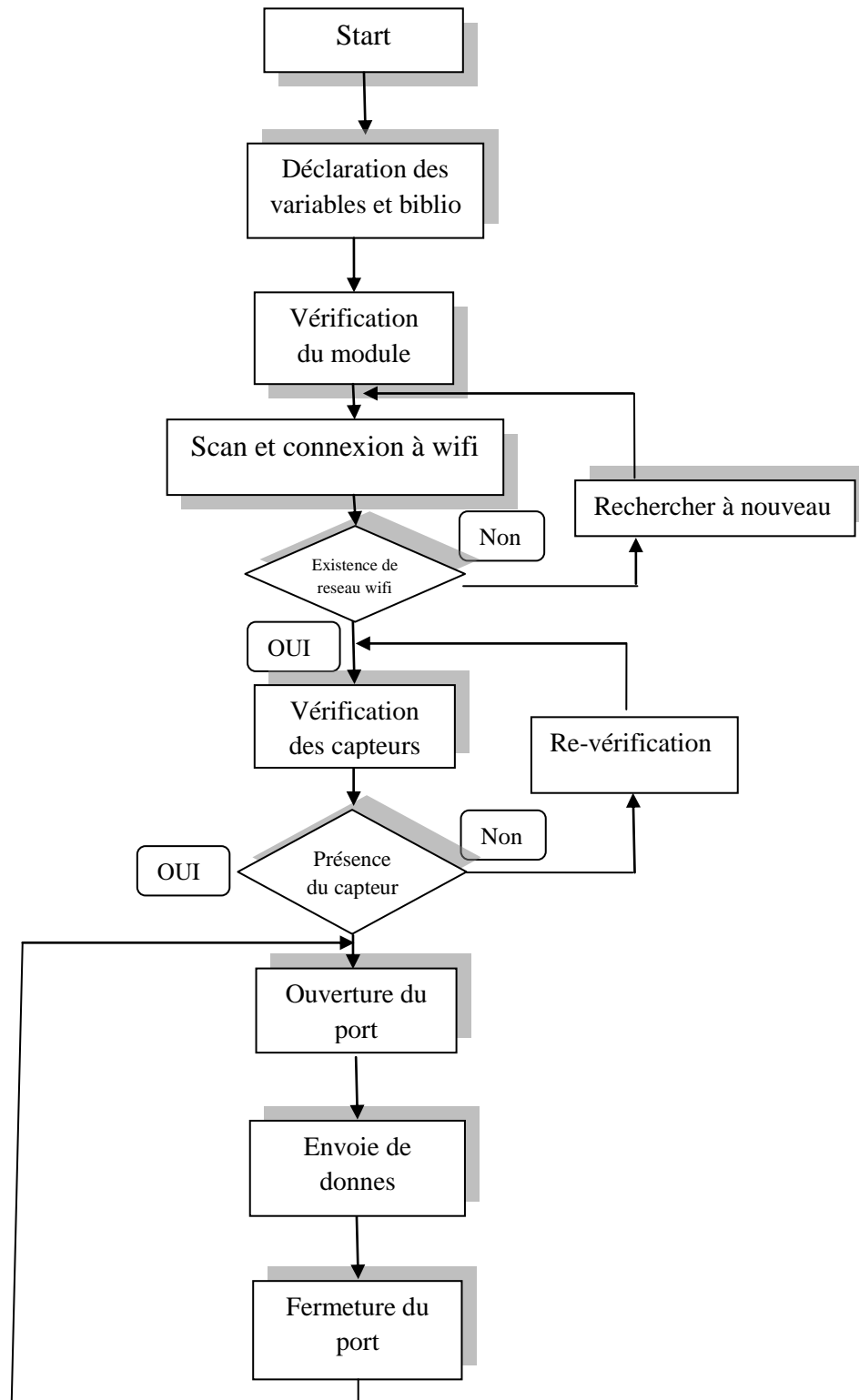


Figure III.13 L'organigramme du programme

### B.1.1) partie déclarative :

Pour commencer le programme, il nous faut tous d'abord introduire la partie déclarative. Dans cette nous faisons appel aux différentes bibliothèques nécessaires à la gestion de l'ensemble des modules de notre application. Comme dans le logiciel Arduino beaucoup de bibliothèques sont déjà présentes et intégrés avec l'IDE. Elles nous permettent de programmer plus vite et plus simplement nos systèmes. Voici en conséquence les bibliothèques que nous avons utilisées et qui doivent être déclarées :

**DHT** : Pour communiquer avec le capteur, on doit utiliser la bibliothèque DHT présentée sur le site Arduino. Il suffit de la télécharger et de l'installer dans le répertoire "..Arduino/libraries".

**stdlib.h** : le langage Arduino est un langage typique c.a.d toutes les porteurs d'informations (variables, constante ....) doivent avoir un type bien déterminée. Et comme dans notre programme nous avons manipulé des données avec des différents types nous avons besoin parfois de faire des conversions de type. La bibliothèque `stdlib.h` déclare des fonctions qui effectuent la conversion de nombres, Certaines fonctions permettent de la conversion de chaînes de caractères en valeurs numériques, la gestion de la mémoire ainsi que d'autres tâches.

**SoftwareSerial.h** : Dans notre application nous avons besoin à des liaisons de transmission de type série, et comme la carte UNO ne dispose que d'une seule liaison série matérielle sur les broches 0 et 1 (qui va aussi à l'ordinateur via la connexion USB). Cette communication série matérielle est réalisée par un module matériel (intégré dans la puce) appelé UART. Ce matériel permet a la puce Atmega de recevoir des communications série même en travaillant sur d'autres tâches, tant qu'il y a de la place dans les 64 octets de buffer de l'UART (le buffer reçoit les données entrantes et les stocke en attendant qu'elles soit lues).

La librairie de communication série présentée ici a été développée pour permettre des communications série sur d'autres broches numériques de la carte Arduino, en utilisant un programme (le code de la librairie) pour répliquer la même fonction de communication série (d'où le nom de Software Serial pour communication série logicielle).

Pour inclure la librairie SerialSoftware dans un programme, on ajoutera au début du programme la ligne suivante : `#include <SoftwareSerial.h>`

En raison du fait qu'elle n'est pas basée sur un module matériel, cette librairie a quelques limitations :

- seules les vitesses jusqu'à 9600 bauds fonctionnent
- l'instruction `Serial.available` ne fonctionne pas
- l'instruction `Serial.read` attendra jusqu'à ce toutes les données arrivent
- Seules les données reçues tant que l'instruction `Serial.read()` est active seront reçues. Les données reçues à d'autres moments ne seront pas prises en compte.

Ensuite, nous sommes tenus de définir le type de capteur et le pin qu'il utilise, tout pins peut être utilisé, dans notre cas nous avons utilisé le pin 7 ce pin est configuré comme un port de transmission série et que la carte Arduino a deux port de communication série est sont déjà utiliser par le module Wifi c'est en ajoutant `#define DHTPIN 7` au programme, `#define DHTTYPE DHT11` cette ligne nous définit le type de capteur utilisé.

La suite consiste à définir un ensemble de variable on peut citer : le variable `url` de type `String` ce variable contient l'adresse du page de réception fusionné avec elle le variable `state` qui sauvegarde les valeurs reçus de la carte Arduino et puis il fait apparaitre sur la page d'affichage.

Après on doit déclarer un ensemble de réseaux wifi auquel la système peut se connecté. Enfin on doit définit les pins que le module wifi ESP-8266 va utiliser. Ces pins sont la pin0 et la pin1 qui assure une liaison série et qui sont définit par défaut comme RX et TX

```
#include <stdlib.h>
#include "DHT.h"
#include "SoftwareSerial.h"
#define DHTPIN 7
#define DHTTYPE DHT11
String url = "GET /rec.php?state=";
String ssid ="LIE4G";
String password="147852369aa";
SoftwareSerial esp(1, 0);// RX, TX
DHT dht(DHTPIN,DHTTYPE);
bool done = false;
```

### B.1.2) partie verification du module

Avant de commencer le processus d'envoi des données au serveur on doit s'assurer que le module Wi-Fi est connecté et fonctionne correctement, nous commençant tous d'abord par configurer son vitesse de transmission dans notre cas ont définit la vitesse de transmission par 115200 bauds. Pour vérifier l'envoi des données aux serveur on à déclarer la led branché sur le pin 13 comme indicateur de la bonne transmission.

### B.1.3) Recherche et connexion a wifi :

Pour connecter au réseau Wi-Fi tous d'abord le module WIFI doit chercher les reseaux disponibles ces reseau peut être declarer comme un ensemble de constant dans la partie de déclaration des variable apré on utilise la commande **AT+CWJAP** pour atteindre au reseau a la portée de module avec le SSID et le mot de passe déclarer précédement une fois le système est accédé au reseau on utilise la commande **AT+CIPMUX=1** pour configurer la connexion en multimode. A ce moment le réseaux va attribuer une add ip le module WIFI.

La commande **AT+CIFSR** ne donne l'adresse IP du module cette commande est utilisé juste pour la confirmation de la connexion. Cette comande

```
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(1000);
  done=esp.find("OK");
}
delay(5000);
esp.println("AT+CIFSR");
done = esp.find("OK");
if(!done){
  delay(5000);
  done=esp.find("OK");
}
}
```

### B.1.4) vérification du capteur :

Le capteur utilise le pin 7 comme une liaison série et on sait que le capteur utilise le mode de connexion semi-deplex ce necessite un test de présence de ce dernier par la commande **dht.begin()** OÙ la carte envoie un signal de réinitialisation et après un temps de quelque nano seconde le capteur répond et être prêt à travailler.

### B.1.5) Ouverture du port et l'envoi des donnés

Dans cette partie qui est considéré comme une boucle nous lisons la valeur du température par la commande **dht.readTemperature()**, Ensuite, nous ouvrons un port ce qui nous permet de connecter à l'ordinateur du médecin par la commande **AT+CIPSTART** , la commande **AT+CIPSEND** est utilisé pour l'envoi des donnés avec une taille précise la commande **digitalWrite (13,OUTPUT)** allume la led

branché dans la pin 13 cette led confirme l'envoi des données ,et après la fermeture du port par la commande **AT+CIPCLOSE** la led s'est éteint.

```
void loop() {
  int chk = dht.readTemperature();
  delay (3000);
  esp.println("AT+CIPSTART=4,\"TCP\", \"192.168.8.100\",80");
  delay(8000);
  esp.println("AT+CIPSEND=4,44");
  delay(3000);
  done=esp.find(">");
  delay(3000);
  esp.println(url + chk);
  delay(10000);
  done = esp.find("Recv 44 bytes");
  if(!done){
    delay(5000);
    done=esp.find("OK");
    digitalWrite(13,HIGH);
  }
  esp.println("AT+CIPCLOSE");
  done = esp.find("CLOSED");
  if(!done){
    delay(5000);
    done=esp.find("CLOSED");
    digitalWrite(13,LOW);
  }
  delay(2500);
  esp.println("AT");
  delay(2000);
}
```

### **B.2) Partie Serveur (cote superviseur médecin) :**

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 le traitement et l'affichage des données pour qu'elles puissent être exploités par le superviseur (le médecin). Notre choix été l'utilisation d'un serveur local de type XAMP.

#### **B.2.1) Présentation de server local XAMP :**

**B.2.1.1) Définition d'un server local :** Un serveur local est un programme (et par implication généralement l'ordinateur dans lequel il s'exécute) qui «dessert» les ressources (fichiers, stockage, programmes d'application, imprimantes et autres périphériques) pour un certain nombre de postes de travail attachés.

Dans notre cas nous avons utilisé XAMPP qu'est une distribution d'Apache populaire dans le milieu de la programmation PHP. Sa facilité d'installation et d'utilisation ainsi que sa gratuité permet de se mettre à la programmation PHP en plus contient la plupart des outils nécessaires pour reproduire le comportement d'un serveur web, à savoir :

- **Apache** : il s'agit du serveur web, c'est lui qui va réceptionner les requête HTTP et les étudier afin de présenter au visiteur la page demandée.
- **Mysql** : il s'agit du Systeme de Gestion de Bases de Données (le SGBD), il va permettre de sauvegarder les données de manière organisée sur le serveur.
- **Php** : ce module d'Apache va lui permettre d'interpréter les pages PHP.
- **PhpMyAdmin** : phpmyadmin est une interface entre vous et vos données, il est fait pour simplifier l'administration de mysql grâce à des pages web. [41]

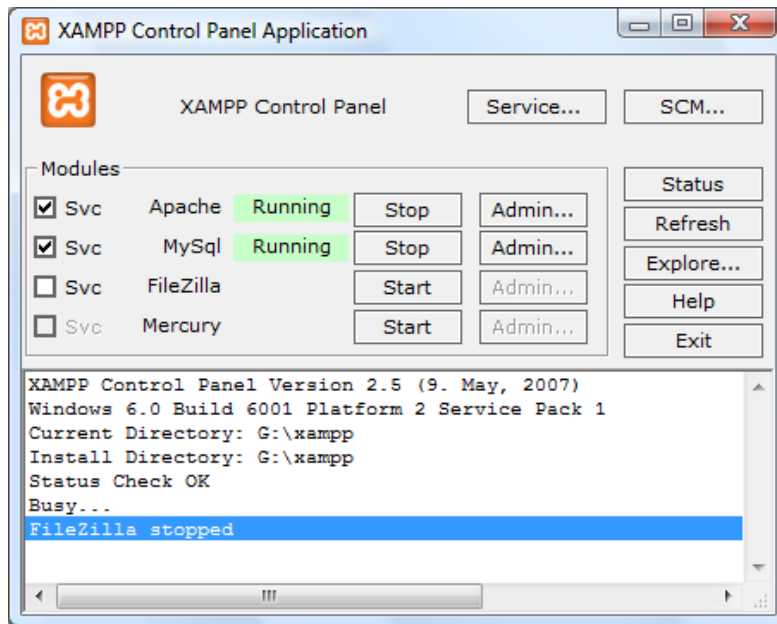


**Figure III.14 Logo de XAMP**

Dans un premier temps il faudra télécharger le programme d'installation disponible à l'adresse suivante : <https://www.apachefriends.org/download.html>

### **B.2.2) Premiers pas avec XAMPP**

Une fois l'installation réalisée, double-clique sur l'icône Xampp-control, Une fenêtre s'ouvrira alors :



**Figure III.15 L'interface de XAMP**

Comme nous avons déjà dit on a programmé deux pages web avec langage PHP et nous l'avons mis dans la piste

**Ordinateur>System C :>XAMP>HTDOCS**

### **B.2.3) La programmation de la page de la réception :**

```
<?php
$status = $_GET["state"];
$fileContent="les donnés recus sont :".$status." degres <br >";
echo "<br >";
$fileStatus = file_put_contents('status.txt',$fileContent,FILE_APPEND);
?>
```

Le but du programme de cette page est de recevoir des données émis par la carte Arduino a partir du réseau internet et les enregistre dans un tampon mémoire puis les stocke en contenu dans un fichier status.txt

Cette page est considéré comme une base de donné, au premier lieu nous avons déclaré un variable **status** qui contient un paramètre **state** ce paramètre prend les valeurs reçus de la carte Arduino.

### **B.2.4) La programmation de la page de l'affichage**

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

Le but principale de cette page est de lire le contenu du fichier status.txt et est d'afficher les résultats à un moment donné. L'affichage des résultats généralement sous forme de courbes graphiques ce qui permet au superviseur (le médecin) d'analyser facilement les résultats obtenus, cette page est hébergée directement dans le Pc du médecin

### III.3) La réalisation pratique

Maintenant que nous avons tout présenté sur le système de télémédecine, 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 en se limite a la température) et d'envoyer cette mesure en degré Celsius vers l'ordinateur via un réseau d'internet WIFI.

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. Puis la broche VCC du module WIFI sur la broche 3.3V et la même chose pour le GND, les broches RX et TX du module WIFI sont ligoté avec leurs pairs dans la carte Arduino suivant le montage de prototypage de la figure III.16:

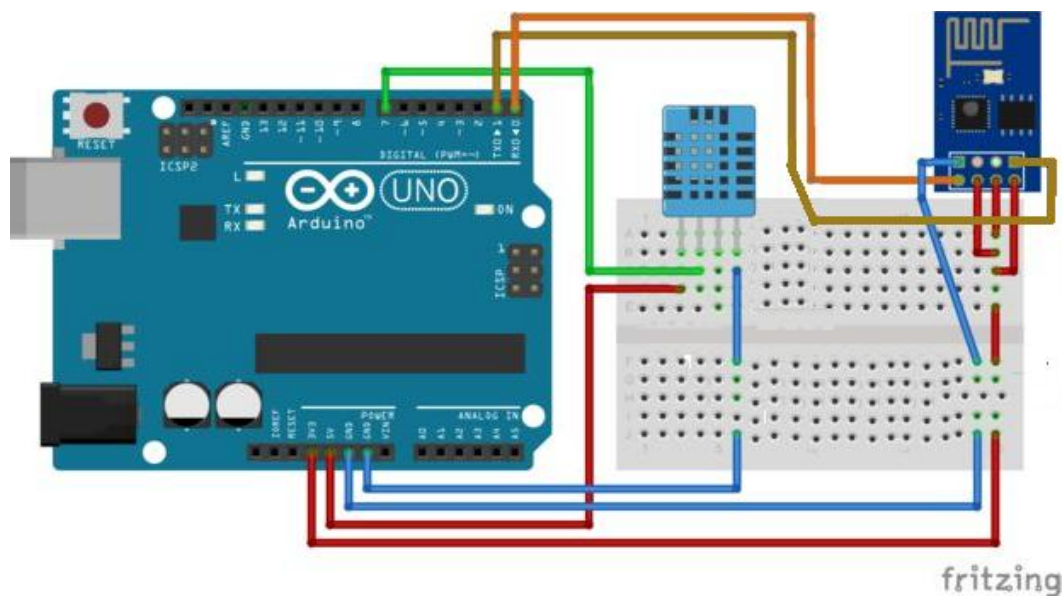
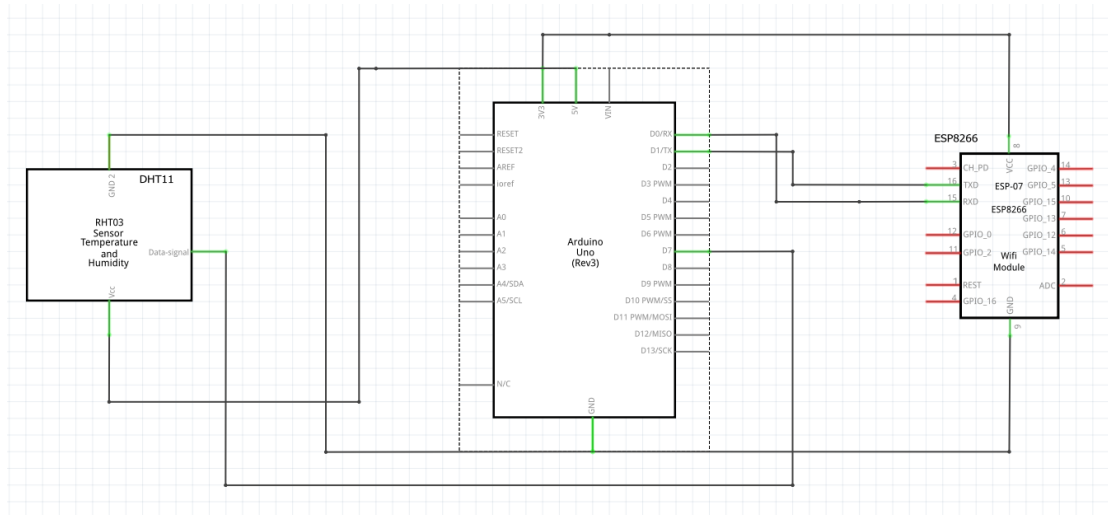
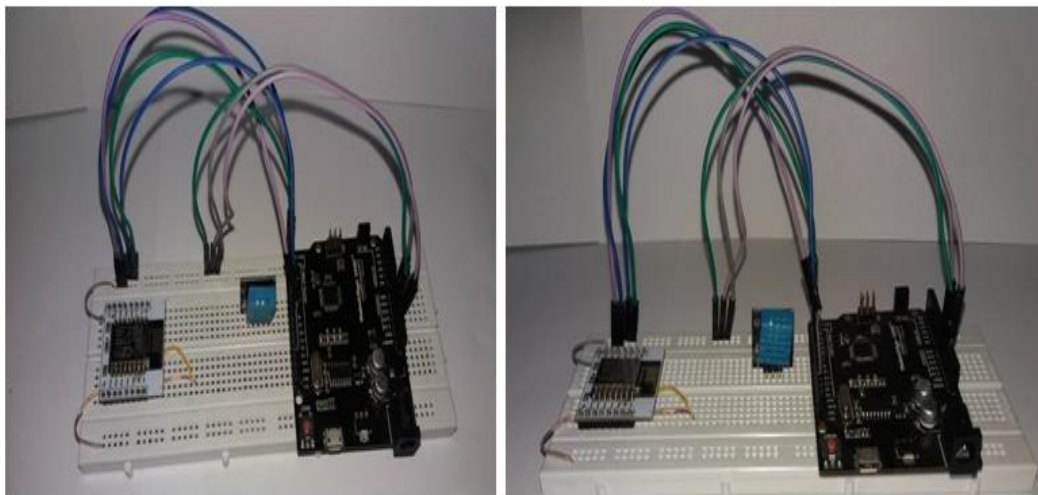


Figure III.16 montage prototypage



**Figure III.17 Schéma électronique du montage**

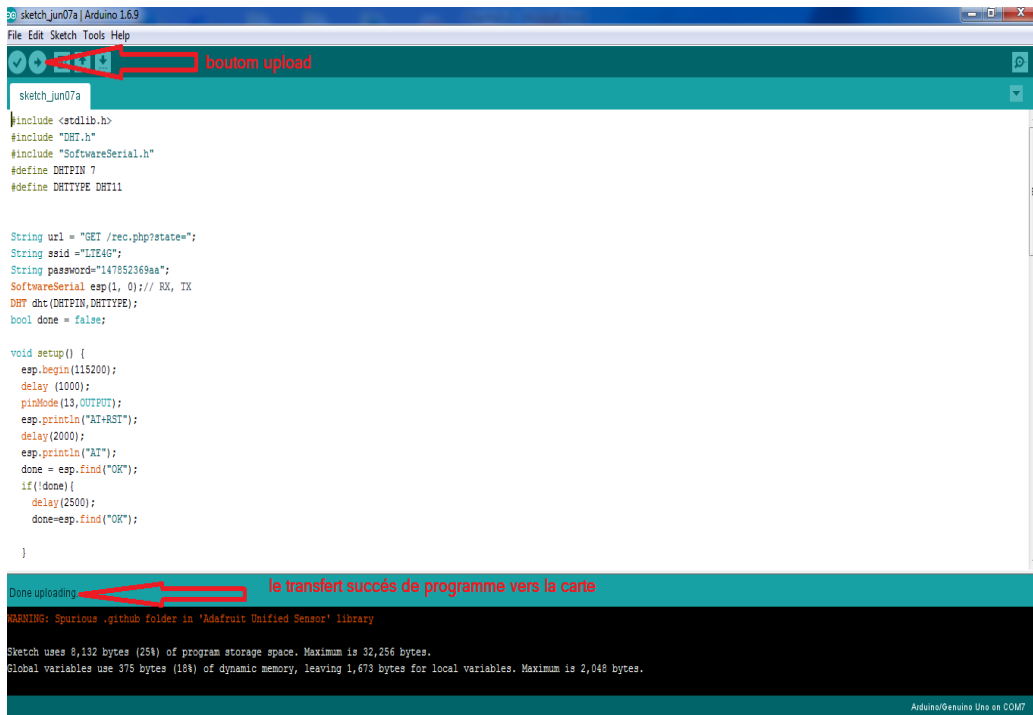
La figure suivante représente l'ensemble des photos du montage fini



**Figure III.18 Photo de montage fini**

Maintenant et après la réalisation de notre montage il est temps de faire le transfert du programme écrit et compilé auparavant vers la carte Arduino, cette étape s'appelle la procédure de téléchargement du programme. La figure III.19 montre une capture d'écran du programme Arduino après le transfert du programme. Comme nous observons sur le rapport le programme a été transféré en succès.

Le tableau III.2 illustre le résultat de la programmation concernant les ressources mémoire utilisées par notre programme. Les résultats obtenus justifient notre choix à cette carte (Arduino UNO).



**Figure III.19 L'envoi du programme vers la carte**

	Total	Utiliser
Mémoire de la carte	32 256 bytes	8132 bytes(25%)
Mémoire dynamique	2048 bytes	375 bytes (18%)

**Tableau III.2 Le rapport de la programmation de la carte**

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 Smartmodem 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.

La première commande à envoyer est la commande AT tous court. Cette commande permet la vérification de la connectivité du module à la carte Arduino. L'ESP doit répondre par « OK » s'il est bien connecté. Cette phase elle est indispensable.

Après on doit initialiser l'ESP en envoyant la commande de reset « AT+RST » cette commande permet son initialisation sans toucher aux paramètres du réseau déjà utilisés. La figure III.20 représente la réponse à l'exécution de cette commande

```
ets Jan  8 2013,rst cause:4, boot mode:(3,7)

wdt reset
load 0x40100000, len 24444, room 16
tail 12
chksum 0xe0
ho 0 tail 12 room 4
load 0x3ffe8000, len 3168, room 12
tail 4
chksum 0x93
load 0x3ffe8c60, len 4956, room 4
tail 8
chksum 0xbd
csum 0xbd

ready
```

**Figure III.20 la réponse d'exécution de la commande AT+RST**

Jusqu'à maintenant l'ESP se souvient toujours des paramètres du dernier réseau auquel il a déjà connecté, donc on doit le forcer à oublier ce réseau. Ce qui nécessite l'utilisation de la commande « AT+CWQAP ». Cette commande permet au module d'oublier tous les réseaux utilisés auparavant. Ce qui rend le module apte à prendre les paramètres d'un nouveau réseau.

Dans l'étape suivante nous devons forcer le module à se connecter au nouveau réseau par l'envoi de la commande « AT+CWJAP=ssid,password », l'exécution de tel commande permet à l'ESP l'obtention d'une adresse ip "WIFI GOT IP", on peut afficher cette adresse ip on envoyions la commande « AT+CIFSR ».

On va configurer le module de se connecter on multimode cela se fait par l'envoi de la commande « AT+CIPMUX=1 ».



```
AT
OK
AT+CWQAP
WIFI DISCONNECT

OK
AT+CWJAP="LTE4G", "147852369aa"
AT+CIPMUX=1
busy p...
WIFI CONNECTED
WIFI GOT IP

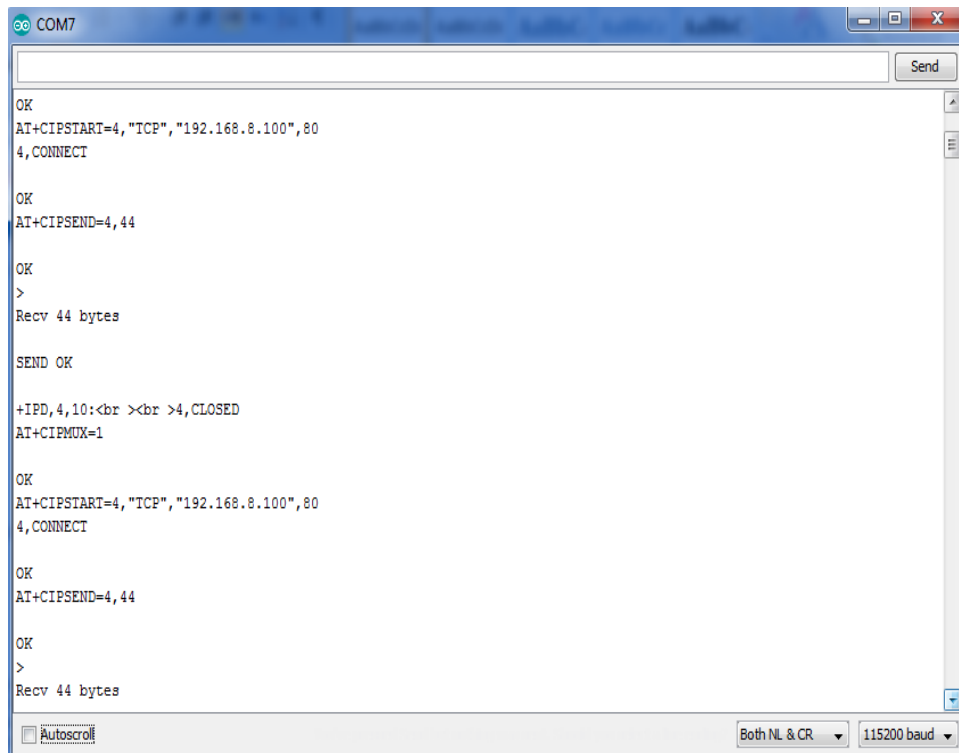
OK
AT+CIFSR
+CIFSR:APIP, "192.168.4.1"
+CIFSR:APMAC, "5e:cf:7f:0b:92:bb"
+CIFSR:STAIP, "192.168.8.101"
+CIFSR:STAMAC, "5c:cf:7f:0b:92:bb"

OK
```

**Figure III.21 Le test des commandes de connexions**

Si l'ensemble des commandes déjà cités sont exécutés avec succès. On peut passer à la transmission des données. La transmission commence tout d'abord par l'ouverture d'un port par la commande « AT+ CIPSTART=id, protocole, adresse ip de server , port de connexion ». Une fois le port est ouvert le système peut transmettre des données on utilise la commande « AT+CIPSEND=id, taille de la donnée » dont on spécifiant le id et la taille de la donnée. la bonne réception est vérifié par l'acquisition des deux réponses "Recv taille\_donnée" et "SEND OK" puis le port doit être fermé par la commande « AT+CIPCLOSE ».

L'exécution correcte de cette tâche assure le bon fonctionnement de notre circuit



```
COM7
Send
OK
AT+CIPSTART=4,"TCP","192.168.8.100",80
4,CONNECT

OK
AT+CIPSEND=4,44

OK
>
Recv 44 bytes

SEND OK

+IPD,4,10:<br ><br >4,CLOSED
AT+CIPMUX=1

OK
AT+CIPSTART=4,"TCP","192.168.8.100",80
4,CONNECT

OK
AT+CIPSEND=4,44

OK
>
Recv 44 bytes

Autoscroll Both NL & CR 115200 baud
```

**Figure III.22** Le test des commandes d'ouverture de porte et d'envoi des données

### III.4) Test de l'application :

Pour tester le bon fonctionnement de notre application nous avons alimenté le circuit par une alimentation externe.

Le test a été effectué dans différents environnements pour tester l'efficacité du capteur Parfois, on le met dans un temps chaud et d'autres fois par temps froid. Les résultats obtenus sont représentés dans la figure III.23 et le graph représenté dans la figure III.24 suivant illustre les changements de température.



Figure III.23 Les résultats obtenus dans la page web

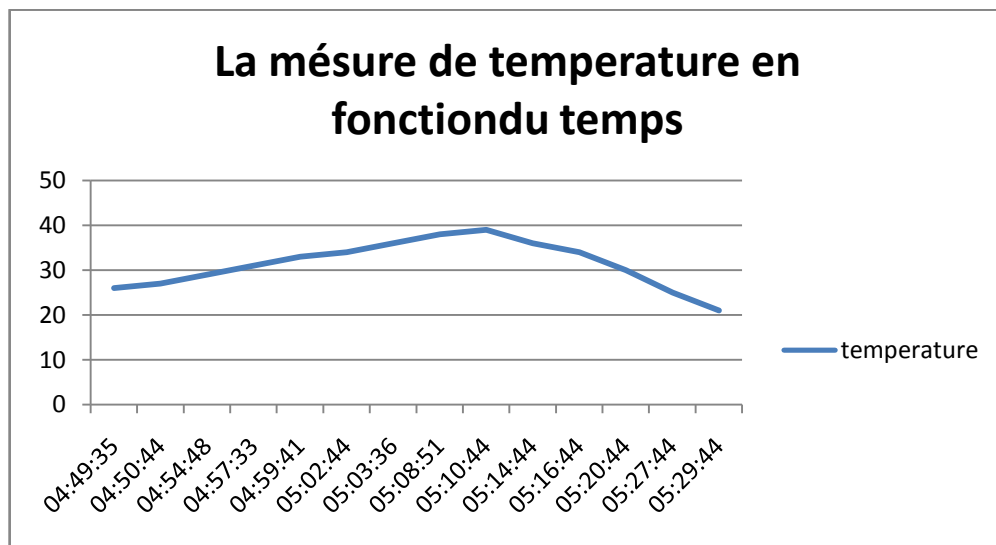


Figure III.24 Représentation graphique des résultats

### **II.5) Conclusion**

Ce travail a pour vocation de proposer une réalisation d'un système de télémédecine pour la surveillance des patients à distance en utilisant un ensemble de capteurs physiologiques. Malheureusement nous n'avons pas pu faire des tests sur un grand nombre de capteurs physiologiques pré-testés, à cause de la non disponibilité de ce type de capteurs au laboratoire de l'université, ainsi que le coût d'acquisition est très important. Nous avons procédé à un prototypage en se limitant à un seul capteur qui est un capteur de température. La réalisation proposée est testée et confirmée.

# Conclusion Générale

### Conclusion Générale

Dans ce mémoire on a traité une problématique actuelle qui est la télésurveillance des malades ; ce qui constitue l'une des questions importantes. En utilisant les technologies de l'information et de la télécommunication, les soins a domicile ou n'importe quelle place font partie de la prise en charge des personnes âgées ou atteinte de maladies chroniques. 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 a 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 a plusieurs problèmes parmi eu :

- La disponibilité de matériel au niveau du laboratoire de l'université.
- La disponibilité des capteurs biomédicaux sur le marché.
- Le cout élevé des différentes composantes surtout les capteurs biomédicaux.
- La réalisation de l'application elle-même.

### 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,

- L'amélioration dans le coté de la communication par exemple ajout d'un module GSM peut nous donner un addendum est de l'utiliser comme une liaison de secours parmi les propriétés que nous bénéficions de ce module est l'appel du médecin, localisation du patient.
- Introduire des traitements des données dans la partie client (coté capteur) afin de ne pas envoyer les données non utiles.

## Conclusion Générale

---

- L'amélioration du système d'information par l'introduction d'un ensemble de procédure de traitement afin d'aider le médecin dans la prise des décisions. Même pour l'envoi des alertes en cas de nécessité.

# Bibliographie

## Bibliographie

---

- [1]. La télésurveillance médicale des patients atteints de maladies chroniques au domicile. European Research in Telemedicine / La Recherche Européenne en Télémedecine Volume 3, Issue 2, June 2014, Pages 85-93
- [2]. Mémoire d'exigence partielle en informatique « Un prototype de système de télésurveillance médicale basé sur les capteurs et les réseaux de capteurs sans fils » présente par Hasna.B en 2014 université de Québec.
- [3]. Capteur pour la télésurveillance médicale. Capteurs, algorithmes et réseaux les auteurs : N.Noury, A.Fleury, R.Nocua, J.Pajoud, C.Gehin, A.Dittmar edition Elsevier Masson
- [4]. <http://www.who.int/mediacentre/factsheets/fs310/fr/>
- [5]. <http://for-ge.blogspot.com/2015/07/microcontrolleur.html>
- [6]. <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
- [7]. <https://www.générationrobots.com/fr/152-arduino>
- [8]. [https://fr.wikipedia.org/wiki/Open\\_source](https://fr.wikipedia.org/wiki/Open_source)
- [9]. <http://www.journaldunet.com/solutions/magazine/open-hardware-et-materiel-libre-definition.shtml>
- [10]. <https://www.arduino.cc/en/main/products>
- [11]. <https://www.arduino.cc/en/Main/ArduinoBoardBT?from=Main.ArduinoBoardBluetooth>
- [12]. [https://en.wikipedia.org/wiki/ESP8266\\_Kit\\_wifi](https://en.wikipedia.org/wiki/ESP8266_Kit_wifi)
- [13]. <https://www.arduino.cc/en/Main/ArduinoWiFiShield>
- [14]. <https://www.arduino.cc/en/Main/ArduinoGSMShield>
- [15]. <http://eskimon.fr/81-arduino-104-le-materiel>
- [16]. <https://fr.flossmanuals.net/arduino/programmer-arduino/>
- [17]. <https://programmingelectronics.com/tutorial-3-arduino-ide-and-sketch-overview/>
- [18]. <https://www.arduino.cc/en/Reference/HomePage>
- [19]. <http://discuss.littlebits.cc/t/introduction-to-arduino-programming-1-the-basics/22237/3>
- [20]. <http://www.commentcamarche.net/faq/3020-wifi-cours-d-introduction>
- [21]. <http://www.commentcamarche.net/contents/1285-transmission-de-donnees-dans-les-reseaux-sans-fils>
- [22]. <http://www.buffalo-technology.fr/fr/technologie/technologie-standard/wireless-80211-technologies/>

## Bibliographie

---

- [23]. Les capteurs en instrumentation industrielle auteur : George.A et Coll DUNOD edition7
- [24]. [http://sti.discip.ac-caen.fr/IMG/pdf/Generalites\\_sur\\_les\\_capteurs.pdf](http://sti.discip.ac-caen.fr/IMG/pdf/Generalites_sur_les_capteurs.pdf)
- [25]. [http://gte.univlittoral.fr/sections/documents-pdagogiques/chapitre-8\\_mesure/downloadFile/file/Les\\_capteurs.pdf?nocache=1289041293.82](http://gte.univlittoral.fr/sections/documents-pdagogiques/chapitre-8_mesure/downloadFile/file/Les_capteurs.pdf?nocache=1289041293.82)
- [26]. [https://asi.insarouen.fr/enseignement/siteUV/capteur/site%20uv%20capteur/CM/pdf/cours6\\_intelligents.pdf](https://asi.insarouen.fr/enseignement/siteUV/capteur/site%20uv%20capteur/CM/pdf/cours6_intelligents.pdf)
- [27]. Thèse de doctorat « traitement numérique du signal physiologique en télémédecine » présenté par RERBAL Souhila soutenue en 2014 université de Tlémcen
- [28]. Biomedical Sensors and Measurement auteur Ping.W and Qingjun.L edition SPRING
- [29]. [http://www-master.ufr-info-p6.jussieu.fr/2005/IMG/pdf/GRETR\\_biocapteurs-haduong-carlier.pdf](http://www-master.ufr-info-p6.jussieu.fr/2005/IMG/pdf/GRETR_biocapteurs-haduong-carlier.pdf)
- [30]. <http://recap-ide.blogspot.com/2014/11/la-saturation-pulsee-en-oxygene-sp02.html>
- [31]. <http://www.biolab.fr/index.php/component/virtuemart/capteurs-de-donnees-exao/capteur-ecg-emg-detail?Itemid=0>
- [32]. <https://www.generationrobots.com/fr/402175-capteur-de-temp%C3%A9rature-corporelle-pour-plateforme-e-health.html>
- [33]. <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>
- [34]. <https://www.arduino.cc/en/guide/introduction>
- [35]. <https://navlab.fr/projets/outils/arduino/alimenter-votre-carte-arduino/>
- [36]. Carte ARDUINO UNO Microcontrôleur ATmega328 B. Cottenceau B311 ISTIA
- [37]. <https://navlab.fr/projets/outils/arduino/alimenter-votre-carte-arduino/>
- [38]. [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.MaterielUno](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielUno)
- [39]. ESP8266EX Datasheet Version 4.3 disponible sur le site : <http://bbs.espressif.com/>
- [40]. [www.micropik.com/PDF/dht11.pdf](http://www.micropik.com/PDF/dht11.pdf)
- [41]. <http://www.supinfo.com/articles/single/1387-presentation-installation-xampp-windows>