



الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان -

Université Aboubakr Belkaïd - Tlemcen -

Faculté de TECHNOLOGIE

## MEMOIRE

Présenté pour l'obtention du diplôme de : **MASTER**

En : Télécommunications

Spécialité : Réseaux et télécommunications

**Sujet**

**Conception et réalisation d'un système de télémétrie avec ARDUINO et Android**

Par:

Mr. MAZOUNI Mohammed Sofiane Mr. EMBOUAZZA Imad-Eddine

Soutenu publiquement, le 26/06/2018 , devant le jury composé de :

Mr BOUACHA Abdelhafid MCA Université de Tlemcen Président

Mr.MERZOUGUI Rachid MCA Université de Tlemcen Examineur

Mr SLIMANI Hicham MCB Université de Tlemcen Encadrant

Année universitaire : 2017/2018

## REMERCIEMENTS

Tout d'abord, nous remercions ALLAH, notre créateur de nous avoir donné la forces, la volonté et le courage afin d'accomplir ce travail modeste.

Nous adressons le grand remerciement à notre encadreur Mr SLIMANI Hicham, pour ses conseils et ses dirigés du début à la fin de ce travail.

Nous sommes conscients de l'honneur que nous a fait Mr BOUACHA Abdelhafid En étant président du jury et Mr MERZOUGUI Rachid d'avoir accepté d'examiner ce travail. Pour l'intérêt que vous avez porté à ce travail et pour vos précieux conseils et remarques.

A toute personne qui a contribué de loin ou de près à la réalisation et la réussite de ce travail.

Enfin, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues.

Ainsi que l'ensemble des enseignants qui ont contribué à notre formation.

# DÉDICACES

Je dédie ce modeste travail à:

Mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études ;

❖ Mes chères sœurs, pour leurs encouragements permanents, et leur soutien moral, chacune à son nom .

❖ Mon cher frère, Adlen pour son appui et son encouragement.

❖ Toute ma famille pour leur soutien tout au long de mon parcours universitaire.

❖ Mon ami et mon binôme Imad-eddine Embouazza.

❖ Mes chers amis Ayoub, Amine et tous mes collègues de l'université AbuBakr Belkaid .

❖ Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible.

Merci d'être toujours là pour moi.

Sofiane

# DÉDICACES

Je dédie ce modeste travail à:

Mes très chers parents qui n'ont pas cessé de m'encourager durant toutes mes études et qui se sont sacrifiés jour et nuit pour mon bonheur et ma réussite.

❖ Mon frère, Amin.

❖ Mes sœurs, Wissem et Afrah, Rania.

❖ Ma tante, Rachida.

❖ Mon ami et mon binôme Mohammed Sofiane Mazouni.

❖ Mes amis et toute la promotion réseaux et télécommunications, ainsi qu'à toutes les personnes qui nous ont aidés à la réalisation de notre travail. Et pour n'oublier personne, j'utiliserai la formule :

❖ « Merci à ... » .

Imad-Eddine

# Table des matières

---

## **Table des matières :**

<b>Remerciement.....</b>	<b>II</b>
<b>Dédicaces.....</b>	<b>IV</b>
<b>Table des matières.....</b>	<b>VI</b>
<b>Liste des figures.....</b>	<b>X</b>
<b>Liste des tableaux.....</b>	<b>XI</b>
1. Introduction générale .....	1
<b>CHAPITRE I : Introduction à la télémétrie</b>	
1. Introduction .....	3
2. Définition de la télémétrie .....	3
2.1 Définition d'un système de télémétrie .....	3
3. Les applications actuelles de télémétrie.....	3
3.1 La domotique .....	3
3.2 La robotique .....	4
3.3 La médecine .....	5
4. Conclusion .....	5
<b>CHAPITRE II : Les Microcontrôleurs</b>	
1. Introduction .....	7
2. Définition d'un microcontrôleur.....	7
3. La gamme des cartes ARDUINO .....	10
4. L'ARDUINO mega2560 .....	11
4.1 Caractéristiques de la carte ARDUINO méga.....	12
4.2 Alimentation .....	12
4.3 Les broches d'alimentation sont les suivantes .....	13
4.4 Mémoire.....	13
4.5 Entrées et sorties numérique .....	13
4.6 Broches analogiques .....	14
4.7 Autres broches .....	14
4.8 Communications .....	14
4.9 Programmation de la carte ARDUINO méga .....	15
5. Le logiciel ARDUINO : espace de développement intégré (EDI).....	15
6. Les modules Wifi.....	17
6.1 La famille ESP8266 .....	17
6.2 L'ESP01 .....	17
6.3 Utilisation de l'ESP8266 comme module wifi .....	17
6.4 Les commandes AT .....	19
7. Capteurs et actionneurs.....	20

# Table des matières

---

7.1	Actionneurs .....	20
7.2	Capteurs .....	21
8.	Conclusion .....	21
<b>CHAPITRE III : Les technologies Web &amp; Android</b>		
1.	Introduction .....	23
2.	Les technologies web .....	23
2.1	Serveur web .....	23
2.1.1	Définition.....	23
2.2	Les logiciels de gestion de serveurs web courants (gratuits) .....	24
2.2.1	Serveur http Apache .....	24
2.2.2	Apache Tom cat.....	24
2.2.3	NGINX .....	24
2.2.4	Microsoft IIS .....	24
2.2.5	Serveur web lite speed.....	24
3.	Les types de serveurs web .....	25
3.1	Serveur web statique .....	25
3.2	Serveur web dynamique.....	25
4.	Les langages de programmation web.....	25
4.1	PHP .....	25
4.1.1	Définition.....	25
4.1.2	Fonctionnement .....	25
4.2	HTML .....	26
4.3	CSS .....	26
4.4	Bootstrap.....	26
4.5	JavaScript.....	26
4.6	JQuery .....	26
4.7	AJAX.....	26
4.8	JSON.....	27
4.8.1	Présentation de JSON .....	27
4.8.2	Représentation du format JSON.....	27
5.	Les principaux protocoles de communication .....	28
5.1	Le protocole TCP/IP .....	28
5.1.1	Définition.....	28
5.1.2	TCP/IP et modèle OSI.....	28
5.1.3	Le protocole IP .....	28
5.2	Les protocoles TCP/UDP.....	29
5.2.1	Système client/serveur.....	29
5.2.2	Le protocole TCP .....	29
5.2.3	Le protocole UDP.....	29

# Table des matières

---

5.3	Le protocole HTTP .....	29
5.3.1	Requête HTTP .....	29
5.3.2	Les commandes http .....	30
5.3.3	Réponse HTTP .....	31
6.	Bases de données .....	32
6.1	Définition .....	32
6.2	Utilité d'une base de données .....	33
6.3	La gestion des bases de données .....	33
6.4	Les principaux SGBD .....	34
6.5	Définition de MySQL .....	34
6.6	Définition De phpMyAdmin .....	34
6.7	Tables de données, lignes et colonnes .....	34
7.	L'univers ANDROID .....	35
7.1	Historique .....	35
7.2	Définition .....	35
7.3	Versions d'Android .....	35
7.3.1	Android 1.0 et 1.1 .....	35
7.3.2	Android 1.5 .....	36
7.3.3	Android 1.6 .....	36
7.3.4	Android 2.0 et 2.1 .....	37
7.3.5	Android 2.2 et 2.3 .....	37
7.3.6	Android 3.0 .....	38
7.3.7	Android 4.0 .....	38
7.3.8	Android 4.1 et 4.2 .....	39
7.3.9	Android 4.4 .....	39
7.3.10	Android 5.0 .....	40
7.3.11	Android 6.0 .....	40
7.3.12	Android 7.0 .....	41
7.4	Android et les langages de programmation .....	42
7.4.1	Android et java .....	42
7.4.2	Le XML .....	42
7.4.3	Android et C/C++ .....	43
7.5	Définition d'une application mobile .....	43
7.6	Objectifs, et perspectives des applications mobiles .....	43
7.7	Types d'applications .....	43
7.7.1	Les applications natives .....	44
7.7.2	Application sous Android .....	44
7.7.3	Application sous Ios .....	44
7.7.4	Les applications sous Windows Phone .....	44

# Table des matières

---

7.7.5	Les applications multiplateformes .....	44
7.8	Fonctionnement d'une application mobile .....	44
7.8.1	Architecture .....	44
7.8.2	Communication client-serveur .....	45
7.9	Les requêtes HTTP sur Android .....	45
7.9.1	Requête HTTP Sans librairie .....	45
7.9.2	Requête HTTP avec librairie .....	46
8.	Conclusion .....	47
<b>CHAPITRE IV : Réalisation du système de télémétrie</b>		
1.	Introduction .....	49
2.	Présentation de notre système de télémétrie .....	49
2.1	Les étapes du travail .....	50
2.2	Les outils et logiciels utilisés durant ce travail .....	50
1.	Hébergement de notre serveur web de Télémétrie .....	50
2.	Création de la base de données .....	51
2.1	Création des tables SQL .....	51
3.	Scripts web PHP et SQL dans le serveur .....	53
1.	L'environnement de travail .....	53
2.	Récupération des données depuis les capteurs .....	53
3.	Le transfert de ces données vers une base de données .....	56
4.	Connexion entre l'ARDUINO méga et l'ESP8266 .....	56
5.	Câblage de la carte ARDUINO avec les deux capteurs .....	57
5.1	ARDUINO en tant que client web en utilisant l'ESP8266 .....	58
5.2	L'envoi des données dans une boîte de dialogue HTTP .....	58
5.3	Effectuation des requêtes sur le serveur en utilisant les commandes AT .....	58
1.	Introduction .....	62
2.	L'environnement de travail .....	62
2.1	Android studio .....	62
2.2	La Communication entre l'application Android et la base de données MySQL .....	63
2.3	Organigramme d'activité de l'application Android .....	64
2.4	Présentation des interfaces de l'application .....	65
2.4.1	La Page d'authentification .....	65
2.4.2	La Page Principale .....	65
2.5	Récupération des données .....	66
1.	Introduction et objectif .....	67
2.	Programmation web .....	67
3.	Architecture du site .....	67
3.1	Page d'authentification .....	67
3.2	Page d'accueil de site web .....	68

# Table des matières

---

3.3	Page d'affichage de toutes les mesures .....	69
3.4	Page de recherche par date .....	70
3.5	Page de recherche une période par date .....	71
3.6	Page de recherche une période par heure .....	72
4.	Conclusion .....	72
<b>Conclusions et perspectives</b> .....		<b>73</b>
<b>Liste des abréviations</b> .....		<b>74</b>
<b>Bibliographies et Webographie</b> .....		<b>75</b>

# Liste des figures

## **Table des figures :**

Figure I. 1: Architecture de la télémétrie domotique .....	4
Figure I. 2 : Exemple d'un robot d'inspection militaire .....	4
Figure I. 3 : Télémétrie en médecine .....	5
Figure II. 1 : les entrées et sorties d'un microcontrôleur .....	8
Figure II. 2: Logo ARDUINO .....	9
Figure II. 3: Carte ARDUINO mini .....	10
Figure II. 4: Carte ARDUINO nano .....	10
Figure II. 5: Carte ARDUINO uno .....	10
Figure II. 6 : Vue de la carte ARDUINO Méga 2560.....	11
Figure II. 7: Interface de logiciel ARDUINO IDE .....	15
Figure II. 8: Interface du terminal Série .....	16
Figure II. 9: la famille ESP des modules wifi .....	17
Figure II. 10: l'ESP01 .....	17
Figure II. 11: les broches du module ESP01 .....	17
Figure II. 12: ESP-01 avec un microcontrôleur fonctionnant en 3,3V .....	18
Figure II. 13: ESP-01 avec un microcontrôleur fonctionnant en 5V .....	18
Figure III. 1: Schéma d'une requête http .....	23
Figure III. 2: Fonctionnement du PHP .....	25
Figure III. 3: description des deux structures .....	27
Figure III. 4: Exemple d'un fichier JSON .....	27
Figure III. 5: Architecture BD-serveur-clients .....	32
Figure III. 6: Architecture du SGDB .....	33
Figure III. 7: Interface de la Version 1.0 .....	35
Figure III. 8: Interface de la version 1.5 .....	36
Figure III. 9: Interface de La Version 1.6 .....	36
Figure III. 10: Interface des versions 2.0 et 2.1 .....	37
Figure III. 11: Interface des versions 2.2 et 2.3 .....	37
Figure III. 12: Interface de la version 3.0 .....	38
Figure III. 13: Interface de la version 4.0 .....	38
Figure III. 14: Interface des versions 4.1 et 4.2 .....	39
Figure III. 15: Interface de la version 4.4 .....	39
Figure III. 16: Interface de la version 5.0 .....	40
Figure III. 17: Interface de la version 6.0 .....	40
Figure III. 18: Interface de la version 7.0 .....	41
Figure III. 19: L'évolution d'Android .....	41
Figure III. 20: Logo JAVA .....	42
Figure III. 21: Logo XML .....	42
Figure III. 22: Logo C/C+ .....	43
Figure III. 23: Le schéma de base de la méthode Volley .....	47
Figure IV. 1: schéma de notre système de télémétrie .....	49
Figure IV. 2: Interface D'accueil du 000webhost .....	50
Figure IV. 3: interface phpMyAdmin .....	51
Figure IV. 4: création de table <mesure de température> .....	52
Figure IV. 5: création de table <mesure des Gaz> .....	52
Figure IV. 6: Vue sur le programme téléversé dans la carte ARDUINO Méga .....	53
Figure IV. 7: Câblage de l'ARDUINO avec ESP8266 .....	56
Figure IV. 8: Branchement du capteur de température .....	57
Figure IV. 9: Branchement du capteur de gaz .....	57
Figure IV. 10: Envoie d'une requete http .....	60

# Liste des figures

---

Figure IV. 11: Réponse Http « nouvelle valeur ajoutée » dans la base de données.....	60
Figure IV. 12: les mesures de températures après l'envoi .....	61
Figure IV. 13: les mesures du gaz, fumée et CO après l'envoi.....	61
Figure IV. 14: Vue sur l'interface de logiciel Android Studio .....	62
Figure IV. 15: Echange de requêtes entre l'application et la base de données.....	63
Figure IV. 16: Page d'authentification .....	65
Figure IV. 17: Vue sur l'interface principale.....	65
Figure IV. 18: Visualisation de la mesure reçue.....	66
Figure IV. 19: Page d'authentification .....	68
Figure IV. 20: Page d'accueil.....	68
Figure IV. 21: page d'affichage de toutes les mesures de température .....	69
Figure IV. 22: page d'affichage de toutes les mesures de gaz .....	69
Figure IV. 23: filtrage de données de température par date.....	70
Figure IV. 24: filtrage de données de gaz par date .....	70
Figure IV. 25: filtrage de données de température (période par jour) .....	71
Figure IV. 26: filtrage de données de gaz (période par jour).....	71
Figure IV. 27: filtrage de données de température (période d'heure par jour).....	72
Figure IV. 28: filtrage de données de gaz (période d'heure par jour) .....	72

## **Liste des tableaux :**

Tableau II. 1: Caractéristiques de la carte ARDUINO méga.....	12
Tableau II. 2: Liste des commandes AT .....	19
Tableau II. 3: Exemples D'actionneurs .....	20
Tableau II. 4: Exemples De Capteurs .....	21
Tableau III. 1: Types de commandes http.....	31
Tableau IV. 1: Branchement de la carte méga et l'ESP8266.....	56

# Introduction générale

---

## 1. Introduction générale

A l'heure actuelle le domaine des technologies de communication est toujours en état d'évolution et l'existence des communications à tout moment est indispensable. L'évolution de ces technologies nous a facilités beaucoup de tâches dans la vie quotidienne et professionnelle. Cette évolution permet également de gagner du temps et de réduire l'effort des humains.

Aujourd'hui, la portée des communications s'étend de quelques micromètres pour des communications de micro puce, aux milliers de kilomètres pour des communications satellitaires. Dans cette perspective, les systèmes de communication modernes ont réduit radicalement le temps de communication entre les entités distantes en rendant la communication et l'échange de données presque instantanée. Dans ce contexte de communication à distance (où télécommunication), nous allons évoquer dans ce projet de fin d'études un sujet très important et qui fait partie de ces nombreuses technologies de communications qui est « La télémétrie »

La télémétrie est définie comme la technologie qui permet la mesure à distance des grandeurs physiques et chimiques, ainsi que l'envoi des informations générées dans la mesure à l'opérateur du système de télémétrie, pour les rapports, les contrôles et la prise de décision. En raison de sa grande utilité, elle est devenue une technologie indispensable dans plusieurs domaines, tels que l'exploration scientifique avec des engins spatiaux sans pilote (satellites, avions et sous-marins), et aussi dans le milieu de l'industrie comme surveillance des processus et la consommation d'énergie et même dans le domaine de la médecine pour le suivi des patients. Un système de télémétrie en temps réel peut être divisé en plusieurs parties :

Tout d'abord, il doit y avoir un dispositif qui collecte les informations des variables physiques et chimiques grâce à l'utilisation des capteurs, il doit y avoir également un dispositif de transmission de données qui permet d'utiliser le réseau internet à haut débit. Ce Dernier va envoyer les données collectées à partir des capteurs à distance vers un serveur via un protocole de communication. Ce serveur communique avec le dispositif de transmission en utilisant le même protocole. Ces informations vont être stockées dans une base de données.

Enfin, nous finissons par la conception d'une application Android dont l'objectif est de récupérer ces informations ainsi qu'un site web pour la visualisation et l'analyse des données.

Pour la réalisation de ce système, nous avons décidé d'utiliser le microcontrôleur ARDUINO qui est une plate-forme électronique ouverte pour la création des prototypes basés sur des logiciels et des matériels flexibles et faciles à utiliser. ARDUINO a l'avantage par rapport aux autres plateformes d'avoir un grand nombre de modules de transmission et d'accessoires à notre disposition, ceci nous donne la permission de mettre en œuvre un large éventail de fonctionnalités pour notre système et cela de manière simple.

Plus précisément, il a été considéré que ARDUINO est la meilleure plate-forme pour ce système de télémétrie du point de vue simplicité et haute performance.

Dans le 1<sup>er</sup> chapitre nous présenterons une introduction générale sur la télémétrie ensuite, le 2<sup>em</sup> chapitre sera dédié pour les microcontrôleurs .par la suite on entame le 3<sup>em</sup> chapitre qui sera consacré pour les technologies web et Android et nous finissons par le chapitre 4 où on va présenter notre système de télémétrie.

# **Chapitre 1 :**

# **Introduction à la télémétrie**

# CHAPITRE 1 : Introduction à la télémétrie

---

## 1. Introduction

Dans ce 1<sup>er</sup> chapitre nous allons essayer de donner une idée générale sur le monde de la télémétrie. Nous allons commencer par définir la télémétrie, ensuite nous répondrons à la question c'est quoi un système de télémétrie ? Et nous finissons par citer les applications actuelles de télémétrie.

## 2. Définition de la télémétrie

Le mot télémétrie est dérivé des racines grecques télé (à distance) et metron (mesure), Bien que le terme télémétrie soit généralement utilisé pour les systèmes distants sans câbles, dans certaines bibliographies, on peut également le trouver pour les systèmes de transmission câblés. La télémétrie comprend un ensemble de procédures pour mesurer les grandeurs physique et chimique d'une position éloignée à l'endroit où les phénomènes que nous voulons analyser se produisent et en plus, couvre l'envoi ultérieur d'informations à l'opérateur du système.

### 2.1 Définition d'un système de télémétrie

Un système de télémétrie est généralement constitué d'un transducteur en tant que périphérique d'entrée, un support de transmission sous la forme de lignes de câble ou d'ondes radio, dispositifs de traitement du signal, et dispositifs d'enregistrement ou affichage des données.

Le périphérique d'entrée peut être distingué comme un ensemble de deux parties Fondamental :

- Le capteur, qui est l'élément sensible primaire qui répond aux variations d'état des grandeurs physiques de l'étude, et le transducteur, qui est responsable de la conversion de la valeur de la température, de la pression ou des vibrations dans le signal électrique correspondant.
- Le support de transmission peut être établi de manière guidée par des moyens tels que réseaux téléphoniques classiques les réseaux informatiques ou les liaisons par fibre optique, ou de manière non guidée, par ondes radio, par Bluetooth ou par communication Wifi ou même par les réseaux de téléphonie mobile.
- Le dispositif de traitement de signal est composé d'un serveur responsable de l'analyse et de la transformation des données, le cas échéant, pour stocker toutes les informations dans une base de données interne de l'ordinateur lui-même.
- Pour la visualisation des données, nous utilisons une application capable d'afficher les valeurs collectées.

## 3. Les applications actuelles de télémétrie

Voici quelques domaines dans lesquels on applique cette technologie à l'heure actuelle, Comme nous pouvons le voir, il est utilisé dans une grande variété de domaines.

### 3.1 La domotique

Au milieu des années 90, les automatismes destinés aux immeubles de bureaux, avec d'autres spécifiques, ils ont commencé à être appliqué à des maisons privées et d'autres types de bâtiments, donnant lieu à la domotique.

La domotique est une maison capable de prendre les meilleures décisions avant différents événements, À cette fin, il intègre une série de mécanismes dans les domaines de

# CHAPITRE 1 : Introduction à la télémétrie

l'électricité, de l'électronique, de la robotique, de la technologie de l'information et des télécommunications. Dans le but de simplifier certaines tâches, d'offrir plus de confort et de sécurité, et d'augmenter les économies d'énergie.

Avec l'intégration de la télémétrie dans le domaine de la domotique, ce n'est pas seulement possible manipuler un équipement de la maison, mais surveillez et contrôlez leur statut à distance en temps réel. Cette technique peut être implémentée via des passerelles résidentielles, à travers lequel vous pouvez accéder au réseau interne de la maison, avoir seulement un serveur web avec une connexion Internet. Cela permet aux propriétaires de contrôler et de gérer les composants de leur maison depuis n'importe où.



Figure I. 1: Architecture de la télémétrie domotique

## 3.2 La robotique

La robotique est une branche de la technologie qui étudie la conception et la construction de machines capables d'effectuer des tâches exécutées par l'être humain ou nécessitant l'utilisation de l'intelligence. L'informatique, l'électronique, la mécanique et l'ingénierie ne sont que quelques-unes des disciplines qui se combinent pour développer un système robotique. Ces systèmes peuvent effectuer des tâches telles que la manipulation de matériaux radioactifs, Déchets toxiques propres, exploitation minière, recherche et sauvetage des personnes et localisation des mines terrestres. Pour être possible, il est nécessaire de recevoir des informations de l'environnement et de les interpréter, pour lesquelles il existe des techniques telles que la télémétrie laser. Cela consiste à lancer un rayon laser qui détermine la distance à laquelle se trouvent les objets qui l'entourent, ce qui lui permet de se déplacer et d'interagir avec lui.



Figure I. 2 : Exemple d'un robot d'inspection militaire

## 3.3 La médecine

En médecine, la télémétrie est couramment utilisée dans nombreuses situations et en utilisant différentes techniques, essentiellement dans le but de surveiller les signes vitaux ou d'autres paramètres des patients de la manière la moins invasive possible, en particulier lorsque la période d'observation s'étend sur plusieurs jours ou même plusieurs semaines. Dans ces cas, les systèmes modernes de télémétrie et d'alarme basés sur les communications mobiles permettent même la surveillance à domicile de l'évolution des patients chroniques ou des patients ayant de longues périodes de convalescence, qui devraient autrement rester en milieu hospitalier ou effectuer des visites avec un contrôle discontinu de son évolution. La télémétrie a l'avantage de fournir un système de surveillance continue, comme dans le cas d'une hospitalisation, mais avec des économies et un confort pour le patient dans son environnement habituel. Parmi d'autres applications, l'une des plus répandues est l'utilisation de la télémétrie pour enregistrer des événements électro cardiographiques à distance. Les émetteurs radio sont connectés au patient à travers 5 électrodes attachées à la peau, Cela permet aux patients la liberté de se déplacer. L'ordinateur central reflète le E.C.G. (Electro Cardio Gramme) des patients connectés et enregistre les événements importants survenus au cours des dernières 24 heures.



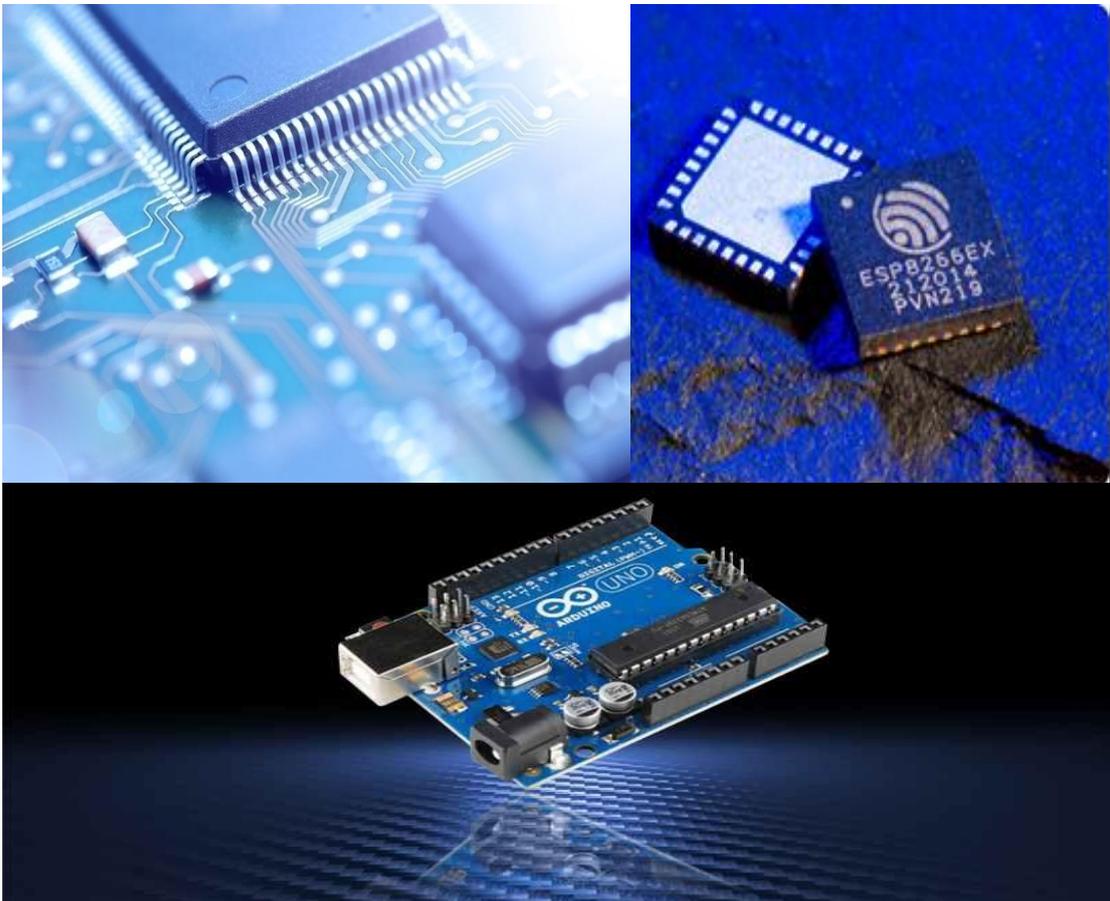
Figure I. 3 : Télémétrie en médecine

## 4. Conclusion

Dans ce 1ère chapitre nous avons parlé de la télémétrie en donnant au début une définition, puis nous avons expliqué qu'est-ce qu'un système de télémétrie et enfin nous avons présenté les différentes applications actuelles de télémétrie, dans le prochain chapitre nous allons évoquer les microcontrôleurs.

# Chapitre 2 :

## Les microcontrôleurs



# CHAPITRE 2 : Les Microcontrôleurs

---

## 1. Introduction

Ce chapitre est une introduction à la plate-forme des microcontrôleurs, nous allons deviser ce chapitre en 3 parties :

-1 ère Partie : nous allons définir les microcontrôleurs en terme général, après nous passons à présenter le microcontrôleur ARDUINO, par la suite, nous examinons la gamme des cartes ARDUINO, en détaillant précisément la carte ARDUINO mega2560 et la raison pour laquelle nous avons choisi d'utiliser cette carte, ses caractéristiques, sa mémoire, Sa programmation ce qui nous conduit à parler de logiciel ARDUINO Ide.

-2 ème Partie : nous parlerons des modules wifi ESP8266, la différence entre ces modules, et nous consacrerons une grande section pour l'ESP01, on évoque son brochage avec la carte ARDUINO et comment le configurer en utilisant les commandes AT.

-3 ème partie : nous présenterons les capteurs et les actionneurs qu'on puisse utiliser dans ce type de projets.

## 2. Définition d'un microcontrôleur

Un microcontrôleur est un circuit intégré rassemblant dans un même boîtier un microprocesseur, plusieurs types de mémoires et des périphériques de communication (Entrées-Sortie). [1']

Le microcontrôleur est composé de quatre parties :

- Un microprocesseur qui va prendre en charge la partie traitement des informations et envoyer des ordres. Il est lui-même composé d'une unité arithmétique et logique(UAL) et d'un bus de données. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur.
- Une mémoire de données (RAM ou EEPROM) dans laquelle seront entreposées les données temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.
- Une mémoire programmable (ROM), qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même. Il existe différents types de mémoires programmables que l'on utilisera selon l'application. Notamment :
  - OTPROM : programmable une seule fois mais ne coute pas très cher.
  - UVPRM : on peut l'effacer plusieurs fois grâce aux ultraviolets.
  - EEPROM : on peut l'effacer plusieurs fois de façon électrique comme les mémoires flash.
- La dernière partie correspond aux ressources auxiliaires. Celles-ci sont généralement :
  - Ports d'entrées / sorties parallèle et série.
  - Des trimmers pour générer ou mesurer des signaux avec une grande précision temporelle.
  - Des convertisseurs A/N pour traiter les signaux analogiques. [2']

# CHAPITRE 2 : Les Microcontrôleurs

Ci-dessous l'aspect d'un microcontrôleur et le schéma des entrées / sorties :

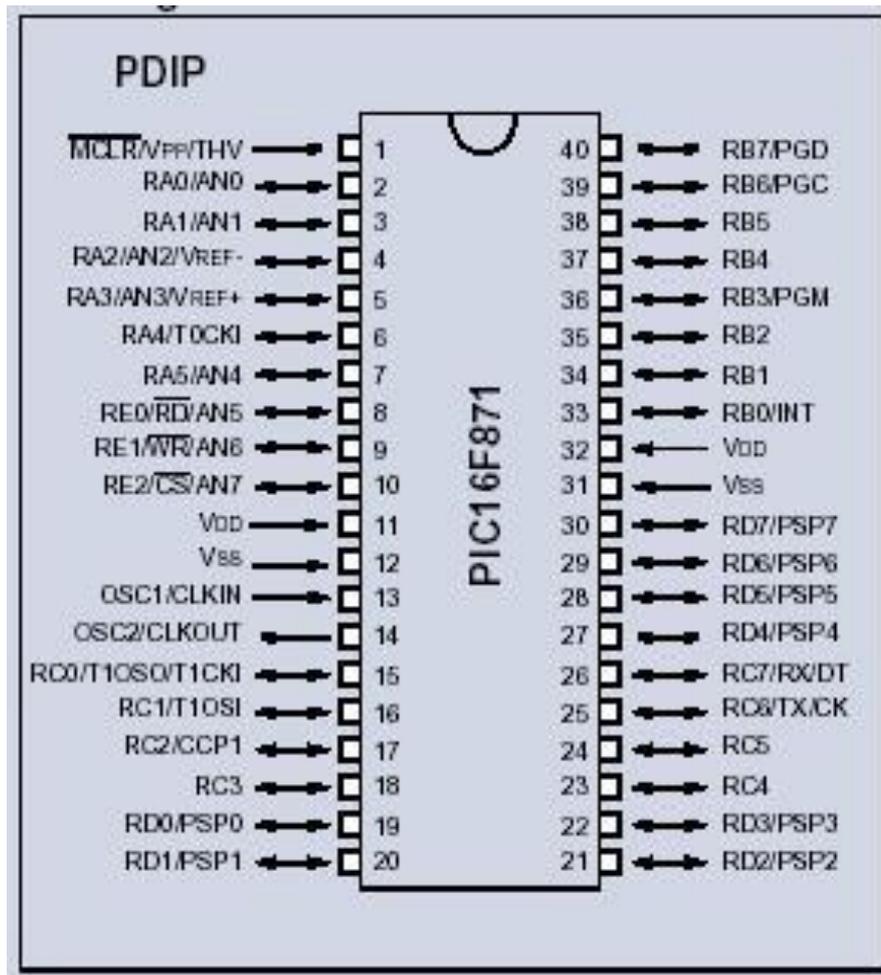


Figure II. 1 : les entrées et sorties d'un microcontrôleur

### ➤ Qu'est-ce qu'un ARDUINO ?

ARDUINO est un outil permettant de construire des dispositifs qui peuvent interagir avec l'environnement qui les entoure. Tu peux t'en servir pour y relier des capteurs détectant du son, de la lumière ou des vibrations, qu'il utilisera alors pour allumer une lumière, changer sa couleur, mettre en route un moteur, et bien d'autres choses. ARDUINO est un système magique, qui se situe au cœur de toutes ces actions. Il collecte des informations à partir de ses capteurs, évaluant ainsi le monde réel qui l'entoure. Il prend ensuite des décisions basées sur les données recueillies et provoque en retour des actions, sous forme de sons, de lumière, ou encore de mouvements. ARDUINO se présente généralement sous la forme d'une carte électronique bleue, qui a à peu près la taille de ta main. Cette carte comporte des inscriptions en blanc qui permettent de repérer ses différents éléments. Tous les composants et les circuits de la carte sont visibles et accessibles. Il existe plusieurs types de cartes ARDUINO, et elles ne sont pas toutes bleues.

ARDUINO est un microcontrôleur, autrement dit un ordinateur très simple. Il ne peut pas faire beaucoup de choses en même temps, mais ce qu'on lui dit de faire, il le fait très bien. Sans le savoir, tu utilises déjà beaucoup de microcontrôleurs dans la vie de tous les jours, car ils sont présents dans beaucoup d'appareils, comme les fours à micro-ondes et les machines à laver. Il existe de nombreux types de microcontrôleurs mais ce qui est particulier avec ARDUINO, c'est qu'il est conçu pour être utilisé par les débutants. Donc, si tu es novice en programmation ou en électronique, tout va bien. Mais attention, ne sous-estime pas ARDUINO pour autant : il peut aussi s'adapter à de gros projets !

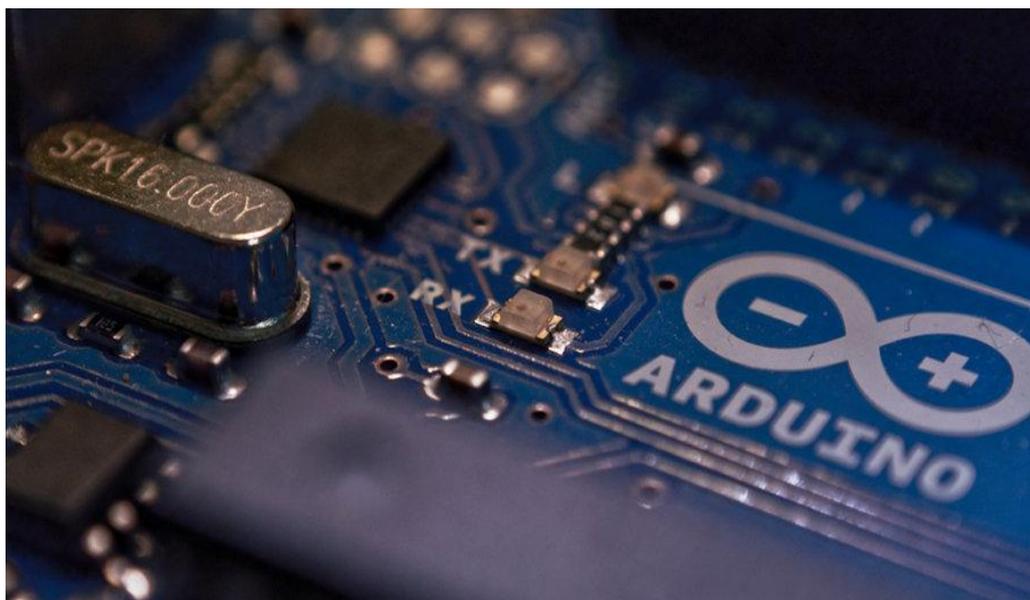


Figure II. 2: Logo ARDUINO

## 3. La gamme des cartes ARDUINO

Actuellement, il existe plus de 20 versions de module ARDUINO, nous citons quelques-uns afin d'éclaircir l'évaluation de ce produit scientifique et académique :

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

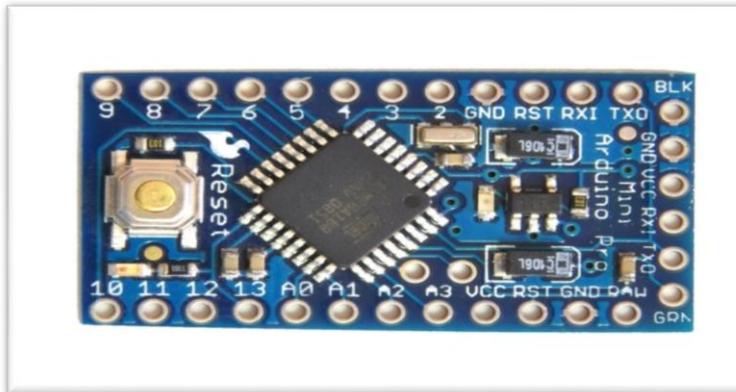


Figure II. 3: Carte ARDUINO mini

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

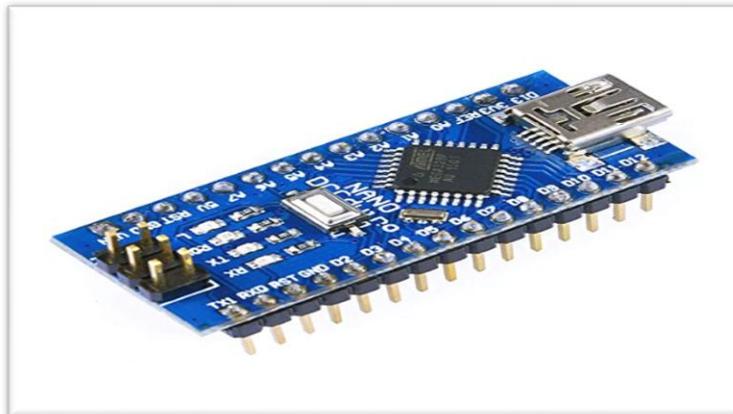


Figure II. 4: Carte ARDUINO nano

- L'ARDUINO UNO, utilisations microcontrôleur ATmega328. La carte ARDUINO UNO diffère de toutes les cartes précédentes car elle n'utilise par le circuit intégré FTDI USB-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série. [1]



Figure II. 5: Carte ARDUINO uno

## ➤ Pourquoi ARDUINO mega2560

Dans notre projet on a choisi d'utiliser la carte ARDUINO mega2560 pour plusieurs raisons mais les plus importantes d'entre elles ce sont : le nombre des entrées et sorties, sa mémoire, son port UART. Nous détaillerons tous ces facteurs dans ce qui suit.

### 4. L'ARDUINO mega2560

La carte ARDUINO Méga 2560 est une carte à microcontrôleur basée sur un ATmega2560.

Cette carte dispose :

- De 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)).
- De 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques).
- De 4 UART (port série matériel).
- D'un quartz 16Mhz.
- D'une connexion USB.
- D'un connecteur d'alimentation jack.
- D'un connecteur ICSP (programmation "in-circuit").
- Et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

La carte ARDUINO Méga 2560 est compatible avec les circuits imprimés prévus pour les cartes ARDUINO UNO, DUEMILANOVE ou DIECIMILA. [3']

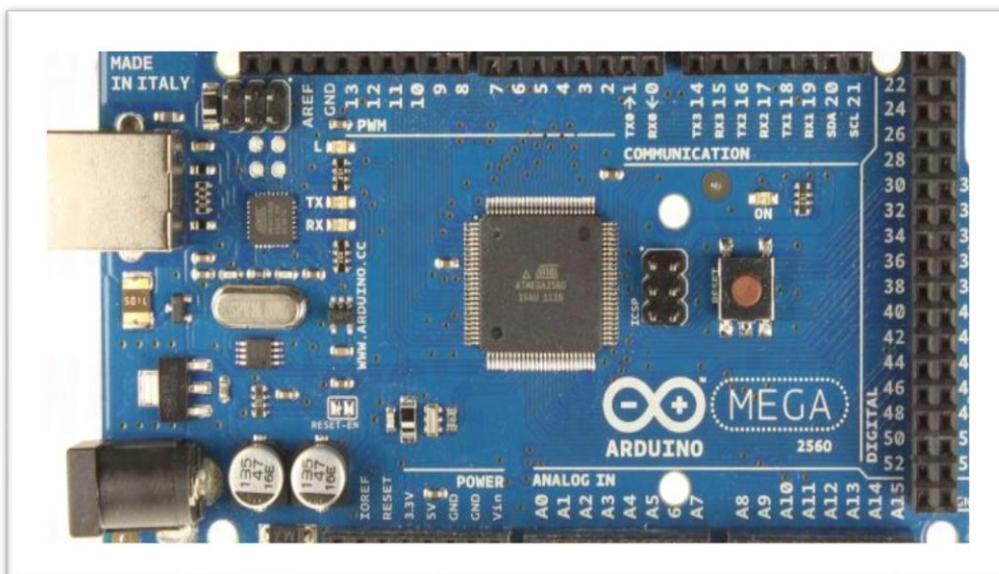


Figure II. 6 : Vue de la carte ARDUINO Méga 2560

## 4.1 Caractéristiques de la carte ARDUINO méga

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	256 KB dont <b>8 KB</b> sont utilisés par le boot loader
Mémoire SRAM (mémoire volatile)	8 KB
Mémoire EEPROM (mémoire non volatile)	4 KB
Vitesse d'horloge	16 Hz

Tableau II. 1: Caractéristiques de la carte ARDUINO méga

## 4.2 Alimentation

La carte ARDUINO Méga 2560 peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe.

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

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte UNO est entre 7V et 12V. La carte ARDUINO mega2560 diffère de toutes les cartes précédentes car elle n'utilise pas le circuit intégré FTDI USB-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

## 4.3 Les broches d'alimentation sont les suivantes

- VIN. La tension d'entrée positive lorsque la carte ARDUINO est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte ARDUINO). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'AT Méga) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.
- GND. Broche de masse (ou 0V). [3']

## 4.4 Mémoire

L'AT méga 2560 a 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le boot loader). L'AT méga 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM). [3']

## 4.5 Entrées et sorties numérique

Chacune des 54 broches numériques de la carte Méga peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage ARDUINO. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH). De plus, certaines broches ont des fonctions spécialisées :

Communication Série : Port Série Serial :

- 0 (RX) et 1 (TX), Port Série Serial 1.
- 19 (RX) et 18 (TX), Port Série Serial 2.
- 17 (RX) and 16 (TX); Port Série Serial 3.

Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

- Interruptions Externes : Broches 2,3,18,19,20 et 21. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.

- Impulsion PWM (largeur d'impulsion modulée) : Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.

- SPI (Interface Série Périphérique): Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.

- I2C : Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils"). Noter que ces broches n'ont pas le même emplacement que sur les cartes UNO, DUEMILANOVE ou DIECIMILA.

- 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. [3']

### 4.6 Broches analogiques

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (c.-à-d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage ARDUINO. Par défaut, ces broches mesurent entre 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.[3']

### 4.7 Autres broches

Il y a deux autres broches disponibles sur la carte :

- AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.

- Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte. [3']

### 4.8 Communications

La carte ARDUINO Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte ARDUINO, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'AT méga 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte uno.

L'ATmega2560 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils")) et SPI :

- Le logiciel ARDUINO inclut la librairie Wire qui simplifie l'utilisation du bus I2C.

- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible. [3']

## 4.9 Programmation de la carte ARDUINO méga

La carte Mega2560 peut être programmée avec le logiciel ARDUINO (téléchargeable). Il suffit de sélectionner "ARDUINO Méga" dans le menu Tools Board.

Le microcontrôleur ATmega2560 présent sur la carte ARDUINO Mega2560 est livré avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le Protocol original STK500.

Vous pouvez bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit" en français) voir ces instructions (en anglais) pour plus de détails.

La source du code pour le circuit intégré ATmega8U2 est disponible. L'ATmega8U2 est chargé avec un bootloader DFU qui peut être activé en connectant le cavalier au dos de la carte (près de la carte de l'Italie) et en réinitialisant le 8U2. Vous pouvez alors utiliser le logiciel FLIP de chez Atmel (Windows) ou le programmeur DFU (Mac OS X et Linux) pour charger le nouveau code. Ou bien vous pouvez utiliser le connecteur ICSP avec un programmeur externe (pour réécrire le bootloader DFU). [3']

## 5. Le logiciel ARDUINO : espace de développement intégré (EDI)

**Le logiciel ARDUINO a pour fonctions principales :**

- De pouvoir écrire et compiler des programmes pour la carte ARDUINO.
- De se connecter avec la carte ARDUINO pour y transférer les programmes.
- De communiquer avec la carte ARDUINO. [4']

### ➤ Description de logiciel ARDUINO

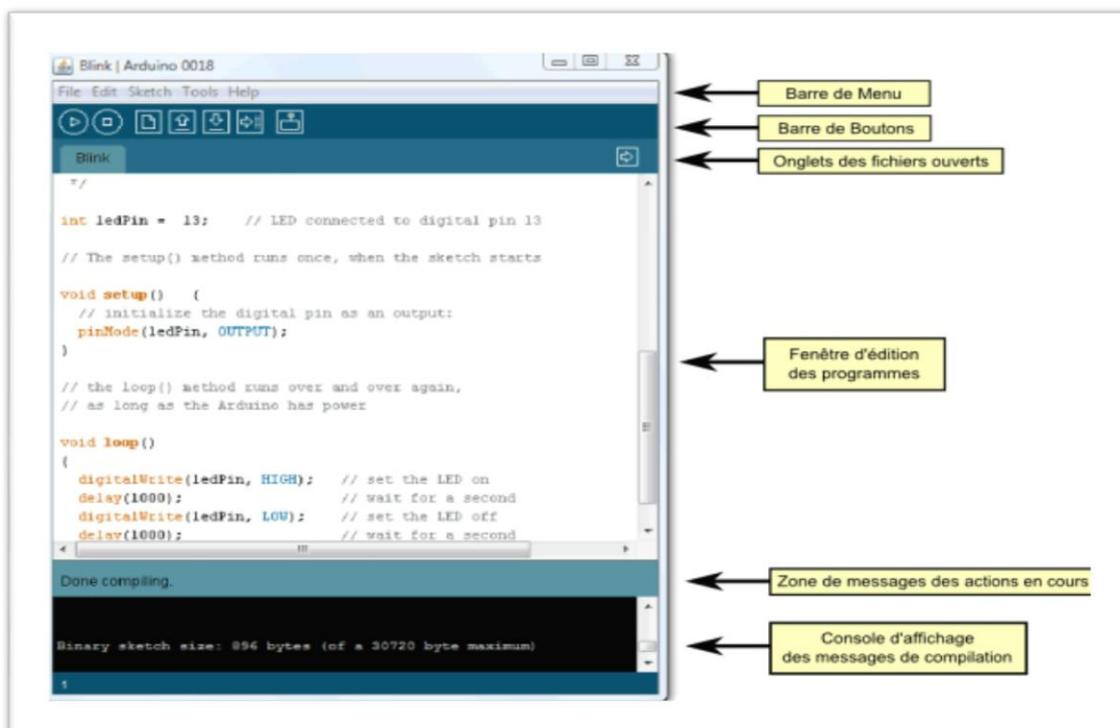


Figure II. 7: Interface de logiciel ARDUINO IDE

# CHAPITRE 2 : Les Microcontrôleurs

## ➤ Principe général d'utilisation

Le code écrit avec le logiciel ARDUINO est appelé un programme (ou une séquence - sketch en anglais) :

- Ces programmes sont écrits dans l'éditeur de texte. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.
- La zone de messages donne l'état de l'opération en cours lors des sauvegardes, des exportations et affiche également les erreurs.
- La console texte affiche les messages produits par le logiciel ARDUINO incluant des messages d'erreur détaillés et autres informations utiles.
- La barre de boutons vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver votre code, et ouvrir le moniteur série.
- La barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel ARDUINO.

Le logiciel ARDUINO intègre également une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte ARDUINO. Les LED RX et TX sur la carte clignote lorsque les données sont transmises. [4']

## ➤ Le terminal série

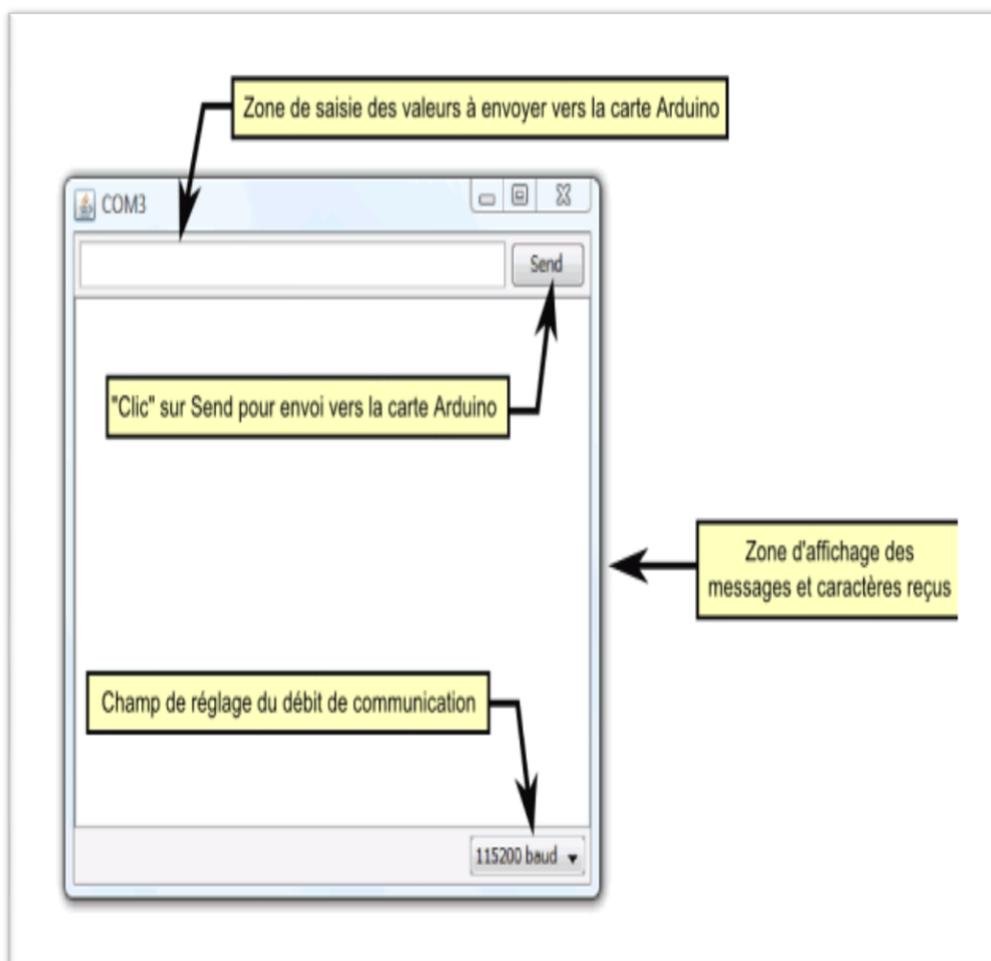


Figure II. 8: Interface du terminal Série

## 6. Les modules Wifi

### 6.1 La famille ESP8266



Figure II. 9: la famille ESP des modules wifi

➤ Les différences entre ces modules portent principalement sur :

- La taille de la mémoire flash externe (de 512 ko à 4 Mo).
- Le nombre de broches accessibles.
- Le type d'antenne WIFI.
- La taille du module.
- Les certifications FCC et / ou CE. [5']

### 6.2 L'ESP01

C'est un petit module permettant de Connecter n'importe quel microcontrôleur (du moment qu'il possède un UART RX/TX) à un réseau Wifi. Il est principalement composé d'un ESP8266 fabriqué par la société Chinoise Expressifs. Il existe d'autres modules à base d'ESP8266 avec plus ou moins de broches. L'ESP-01 est le plus petit. [6']

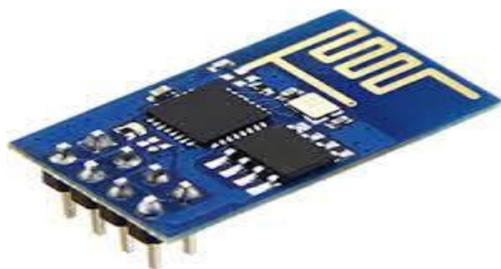


Figure II. 10: L'ESP01

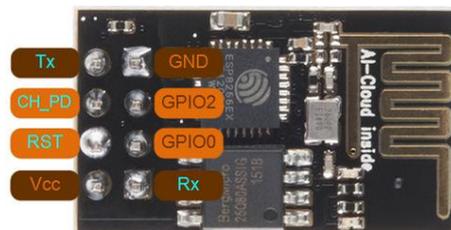


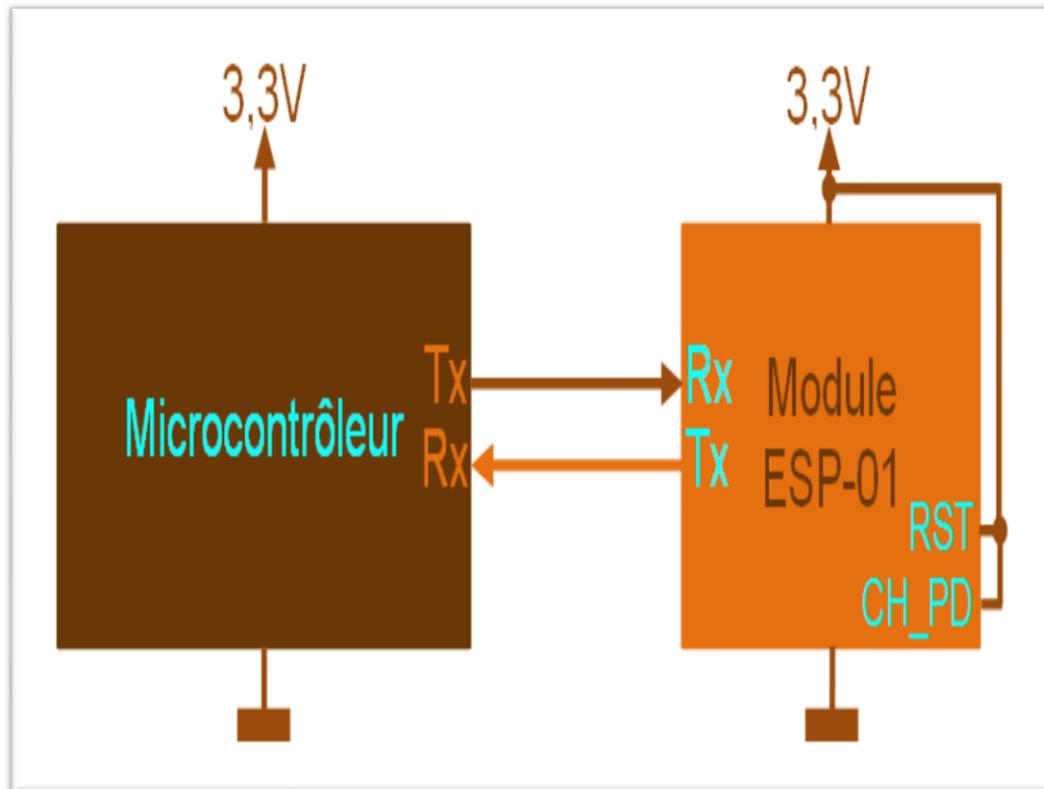
Figure II. 11: les broches du module ESP01

### 6.3 Utilisation de l'ESP8266 comme module wifi

L'utilisation de l'ESP-01 en module Wifi permet d'apporter une connectivité Wifi à un microcontrôleur (ARDUINO, PIC, STM32,...). Dans ce cas-là l'ESP est un périphérique du microcontrôleur.

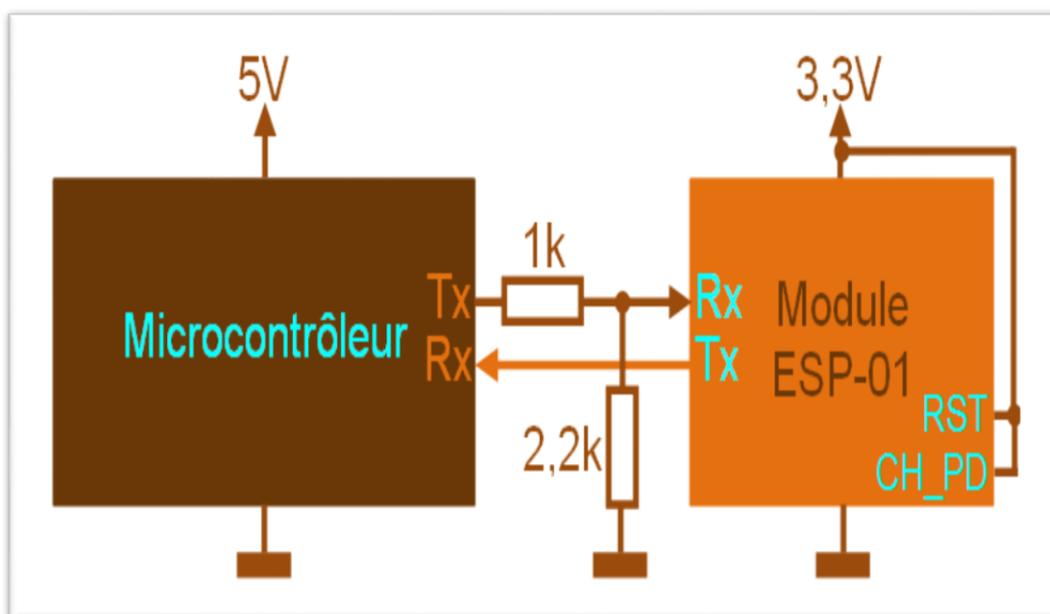
## CHAPITRE 2 : Les Microcontrôleurs

Ce mode de fonctionnement utilise les bornes RX et TX. La connexion avec un microcontrôleur se fait comme indiqué ci-dessous. Les paramètres de configuration par défaut de la liaison série sont : **115200/1/N/N**. [6']



**Figure II. 12: ESP-01 avec un microcontrôleur fonctionnant en 3,3V**

Le TX du microcontrôleur est connecté au RX de l'ESP et inversement, Ceci est logique : Ce qui est envoyé par le microcontrôleur doit être reçu par l'ESP et ce qui est envoyé par l'ESP doit être reçu par le microcontrôleur. [6']



**Figure II. 13: ESP-01 avec un microcontrôleur fonctionnant en 5V**

La commande se fait par l'intermédiaire de commandes AT (langage de commande anciennement utilisé par les modems et également par les modules Bluetooth). [6']

# CHAPITRE 2 : Les Microcontrôleurs

## 6.4 Les commandes AT

Pour pouvoir configurer l'ESP01 nous allons besoin d'ouvrir le terminal série puis nous saisissons les commandes AT. Une commande AT commence toujours par "AT+" suivi de la commande à envoyer. En retour le module renvoie "OK" pour indiquer le bon déroulement de la commande. [7]

Voici quelques commandes AT avec la description qui correspond à chaque commande :

COMMANDE	DESCRIPTION	EXECUTION
<b>AT+RST</b>	Redémarrage du module	AT+RST
<b>AT+GMR</b>	Information sur la version	AT+GMR
<b>AT+GSLP</b>	Règle le temps du sommeil de l'ESP8266 en ms.	AT+GSLP=<Temps>
<b>AT+CWMODE</b>	Le mode WIFI : 1 : Client 2 : Point d'accès 3 : Client et Point d'accès	AT+CWMODE=1,2,3 ?
<b>AT+CWJAP</b>	Connexion au point d'accès	AT+CWJAP= ssid, pwd
<b>AT+CWQAP</b>	Déconnexion	AT+CWQAP
<b>AT+CWSAP</b>	Réglage paramètre point d'accès.	AT+CWSAP= « ssid », «pwd », «ch », «ecn »
<b>AT+CIPSTA</b>	Choisir l'adresse IP en mode client	AT+CIPSTA=« IP »
<b>AT+CIPAP</b>	Choisir l'adresse IP en mode point d'accès	AT+CIPSTA=« IP »
<b>AT+CIPSTATUS</b>	Informations sur la connexion	AT+CIPSTATUS
<b>AT+CIPSEND</b>	Envoyer des données	AT+CIPSEND= «DATA » AT+CIPSEND=id, «DATA »

Tableau II. 2: Liste des commandes AT

## 7. Capteurs et actionneurs

Une carte ARDUINO est capable de fonctionner avec un très grand nombre de capteurs et d'actionneurs du moment qu'ils respectent certaines conditions liées au courant qu'ils acceptent et celui délivré par la carte, mais aussi suivant leurs protocoles de communication nous allons citer quelques exemples de capteurs et d'actionneurs intéressants dans le cadre de l'élaboration d'environnements augmentés.

### 7.1 Actionneurs

Actionneurs	Descriptions	Actionneurs	Descriptions
 <b>LED</b>	S'illumine plus ou moins fort suivant l'intensité de courant envoyée. Existes-en différentes couleurs mais aussi en infrarouge ou en ultraviolet	 <b>Afficheur LCD</b>	Afficheur LCD 2 x 16 se raccorde sur une carte compatible ARDUINO ce module permet d'afficher des informations
 <b>Servo-Moteur</b>	Un servomoteur est un moteur conçu pour générer un mouvement précis selon une commande externe. D'habitude, son mouvement est limité à un arc de 180 degrés.	 <b>Haut-Parleur</b>	Haut-parleur permettant de générer du son.

Tableau II. 3: Exemples D'actionneurs

# CHAPITRE 2 : Les Microcontrôleurs

## 7.2 Capteurs

Capteur	Description	Capteur	Description
 <p><b>Capteur de Force</b></p>	<p>Un capteur de force FSR02 (Force Sensing Resistor) permet de détecter une pression physique.</p>	 <p><b>Capteur PIR</b></p>	<p>Le capteur PIR est un capteur infrarouge permet de détecter un mouvement dans son champ de vision en se basant sur l'infrarouge.</p>
 <p><b>Capteur MQ5</b></p>	<p>Le capteur de gaz (MQ5) est utile pour détecter les fuites de gaz. Il peut détecter le GPL, le gaz naturel, le gaz de ville</p> <p>Avec un temps de réponse rapide</p>	 <p><b>Capteur PING</b></p>	<p>ce capteur envoie un signal ultrason (ping) et écoute pour détecter le retour de ce signal sur un objet, suivant le temps de la réponse on peut détecter la distance de l'objet par rapport au capteur</p>
 <p><b>Capteur LM35</b></p>	<p>Le capteur LM35 est un capteur de température où la tension de sortie V out augmente de 10 mV chaque fois que la température augmente de 1°C. sa plage va de 0°C à 100°C</p>	 <p><b>Capteur de lumière</b></p>	<p>La LDR, photorésistance ou capteur de lumière est un composant électronique dont la résistance varie en fonction de la lumière perçue. La valeur de la luminosité est convertit en résistance qu'il est facile d'acquérir avec l'aide d'une carte ARDUINO</p>

Tableau II. 4: Exemples De Capteurs

## 8. Conclusion

Dans ce 2 ème chapitre, nous avons commencé par donner une idée générale sur les microcontrôleurs, ensuite nous avons passé a présenté la plateforme ARDUINO où nous avons donné plus de détails sur la carte utilisé dans ce projet et Nous Avons fini par citer quelques capteurs que nous pouvons utiliser dans ce genre de projets, Dans le chapitre suivant nous parlerons des technologies web et androïde.

# Chapitre 3 :

## Les Technologies web et Android



# CHAPITRE 3 : Les technologies web et Android

## 1. Introduction

Dans ce chapitre nous allons parler dans la première partie :

- Les technologies web (Serveurs Web) et les langages de programmations web.
- Les Bases de données.
- Les différents protocoles de communication.

Dans la deuxième partie nous allons voir tout ce qui concerne le système d'exploitation Android :

- Ses versions, langages de programmation.
- Les applications mobiles et les requêtes http.

## 2. Les technologies web

### 2.1 Serveur web

#### 2.1.1 Définition

Un « serveur web » peut faire référence à des composants logiciels (software) ou à des composants matériels (hardware) ou à des composants logiciels et matériels qui fonctionnent ensemble.

- Au niveau des composants matériels, un serveur web est un ordinateur qui stocke les fichiers qui composent un site web (par exemple les documents HTML, les images, les feuilles de style CSS, les fichiers JavaScript) et qui les envoie à l'appareil de l'utilisateur qui visite le site. Cet ordinateur est connecté à Internet et est généralement accessible via un nom de domaine tel que <Google.com>.
- Au niveau des composants logiciels, un serveur web contient différents fragments qui contrôlent la façon dont les utilisateurs peuvent accéder aux fichiers hébergés. On trouvera au minimum un serveur HTTP. Un serveur HTTP est un logiciel qui comprend les URL et le protocole HTTP (le protocole utilisé par le navigateur pour afficher les pages web).
- Au niveau le plus simple, à chaque fois qu'un navigateur a besoin d'un fichier hébergé sur un serveur web, le navigateur demande (on dit qu'il envoie une requête) le fichier via HTTP. Quand la requête atteint le bon serveur web (matériel), le serveur HTTP (logiciel) renvoie le document demandé, également grâce à HTTP. [8']

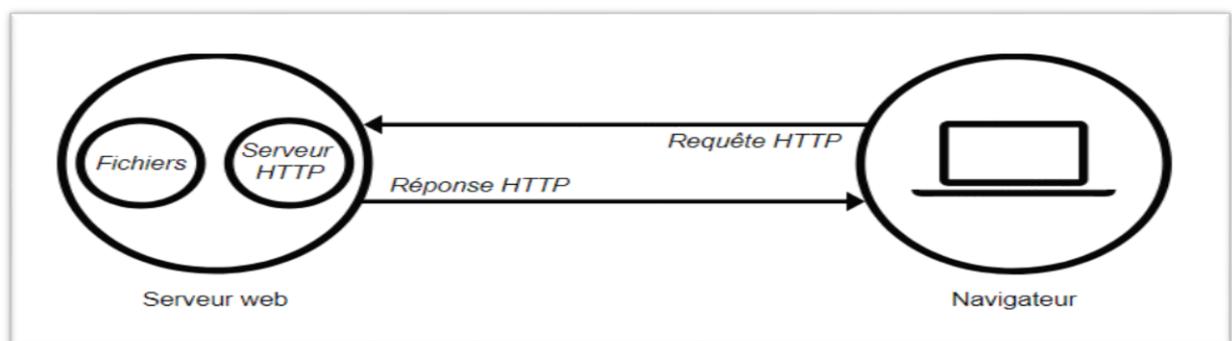


Figure III. 1: Schéma d'une requête http

## 2.2 Les logiciels de gestion de serveurs web courants (gratuits)

Il existe de nombreux programmes permettant d'exploiter un serveur Web. La plupart sont disponibles gratuitement en téléchargement sur Internet. Le serveur Web utilisé doit évidemment être compatible avec le système d'exploitation de l'hébergeur. La plupart des serveurs Web reposent sur UNIX ou sur Linux (Apache, Nginx, Tomcat, etc.) mais souvent aussi sous Windows. Microsoft IIS représente une exception, car il ne fonctionne que sous Windows. Voici une brève présentation des programmes d'administration de serveurs Web les plus utilisés :

### 2.2.1 Serveur http Apache

Le serveur http Apache (ou plus simplement serveur Apache) est continuellement développé et amélioré grâce à sa base open source. Ce programme modulaire fonctionne aussi bien avec tous les systèmes d'exploitation, c'est le serveur Web le plus utilisé. Il est très adapté aux contenus Web dynamiques : on peut en effet intégrer du langage script PHP sans problème. Grâce à l'interface CGI, Apache peut exécuter d'autres programmes.

Il est possible de contourner l'installation d'un serveur Web Apache, qui est assez compliquée, en utilisant un ensemble de logiciels XAMPP ou LAMP. De tels ensembles de programmes simplifient en effet fortement l'installation. Ils comportent, en dehors d'Apache, des logiciels de gestion de base de données et de langages script.

Dernièrement, le serveur http Apache a perdu en rapidité en comparaison avec la concurrence. En effet, de nombreux concurrents se démarquent par des temps de réponse et des transferts de données plus courts.

### 2.2.2 Apache Tom cat

Il s'agit d'un programme de serveur open source basé sur un environnement Java. Si vous souhaitez produire des contenus Web avec Java, ce logiciel est très adapté dans la mesure où il propose un conteneur Web. De plus, il peut être incorporé à d'autres serveurs Web (serveur Apache HTTP ou Microsoft IIS) via un connecteur AJP.

### 2.2.3 NGINX

Ce logiciel gratuit propose plus de fonctions qu'un serveur Web habituel. NGINX propose entre autres aussi un proxy inverse, ce qui permet de réduire la charge sur le serveur et donc de travailler plus rapidement. Ces dernières années, NGINX a beaucoup gagné en popularité. Vous trouverez de plus amples informations sur ce programme dans notre introduction à NGINX.

### 2.2.4 Microsoft IIS

Le programme Microsoft Internet Information Service (IIS) correspond à un ensemble de logiciels dans lequel on trouve aussi un serveur Web (programme). Auparavant, IIS était un élément intégré système de serveur Windows. Depuis, le programme peut être installé ultérieurement et indépendamment, si besoin, sur quelques systèmes d'exploitation Windows. Le fait que l'on ne puisse utiliser IIS que sur des serveurs basés sous Windows est peu surprenant, mais joue en sa défaveur.

### 2.2.5 Serveur web lite speed

Lite Speed est un logiciel de serveur Web pour Linux et Unix avec lequel vous avez le choix entre une version open source, une version standard gratuite et une variante pour les entreprises. Le serveur Web Lite Speed se caractérise surtout pour sa rapidité de transmission de données. Celle-ci est d'autant plus élevée dans la version professionnelle. [9']

## 3. Les types de serveurs web

Pour publier un site web, nous aurons besoin d'un serveur web statique ou dynamique.

### 3.1 Serveur web statique

Un serveur web statique (aussi appelé une pile) est composé d'un ordinateur (matériel) et d'un serveur HTTP (logiciel). Il est appelé « statique » car le serveur envoie les fichiers hébergés « tels quels » vers le navigateur.

### 3.2 Serveur web dynamique

Un serveur web dynamique possède d'autres composants logiciels, certains qu'on retrouve fréquemment dont un serveur d'applications et une base de données. Il est appelé « dynamique » car le serveur d'applications met à jour les fichiers hébergés avant de les envoyer au navigateur via HTTP. [8']

## 4. Les langages de programmation web

### 4.1 PHP

#### 4.1.1 Définition

PHP est un langage informatique de script Open Source, principalement dédié au développement web (d'où son nom : HyperText Préprocesseur). Il s'agit d'un des langages les plus utilisés au monde pour créer des sites web dynamiques.

#### 4.1.2 Fonctionnement

D'un point de vue exécution, PHP a besoin d'un serveur Web pour fonctionner. Toutes les pages demandées par un client seront construites par le serveur Web, en fonction des paramètres transmis, avant d'être retournées au client.



Figure III. 2: Fonctionnement du PHP

## 4.2 HTML

L'HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactifs, tel des liens, des images ou bien encore des animations. Ces éléments ne sont pas dans le code source d'une page codée en HTML mais "à côté" et la page en HTML ne fait que reprendre ces éléments.

## 4.3 CSS

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

## 4.4 Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur) des sites et des applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

## 4.5 JavaScript

Le JavaScript est un langage informatique utilisé sur les pages web. C'est un langage objet interprété dont les instructions sont stockées dans les pages html, ce langage de programmation est conçu pour traiter des événements provoqués par le lecteur (par exemple lorsque le lecteur fait glisser la souris sur une zone de texte cette dernière change de couleur) le texte contenant le programme est analysé au fur et à mesure par l'interprète. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer.

## 4.6 JQuery

jQuery est une bibliothèque JavaScript rapide, gratuite et multiplateforme. Compatible avec l'ensemble des navigateurs Web (Internet Explorer, Safari, Chrome, Firefox, etc.), elle a été conçue et développée pour faciliter les fonctionnalités communes de JavaScript. La bibliothèque jQuery permet les fonctionnalités suivantes : Manipulation du DOM (HTML ou CSS), La gestion des évènements.,AJAX.,Les effets d'animation.

## 4.7 AJAX

AJAX (Asynchronous JavaScript and XML) n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes, dont : HTML ou XHTML, les feuilles de styles CSS, JavaScript, le modèle objet de document (DOM), XML, XSLT, et l'objet XMLHttpRequest. Lorsque ces technologies sont combinées dans le modèle AJAX, les applications Web sont capables de réaliser des mises à jour rapides de l'interface utilisateur sans devoir recharger la page entière du navigateur. Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisateur. [9']

## 4.8 JSON

### 4.8.1 Présentation de JSON

JSON (JavaScript Object Notation) est un format de données basé sur le langage de programmation JavaScript. Il est très utilisé dans l'échange de données au même titre que XML (eXtensible Markup Language). JSON reprend le formalisme d'écriture d'un objet JavaScript ce qui facilite grandement son utilisation dans une application codée dans ce langage. Il n'est pas pour autant dépendant de JavaScript et il existe un grand nombre d'interpréteurs pour l'utiliser dans plusieurs langages. JSON est de plus en plus utilisé pour l'échange de données car il est plus simple à utiliser et plus léger que XML. Ce format est construit suivant deux structures : un objet contenant une collection de paires nom/valeur désordonnée et un tableau contenant une liste de valeurs ordonnées. Une valeur peut être de différents types : string, number, object, array, boolean et nulle.

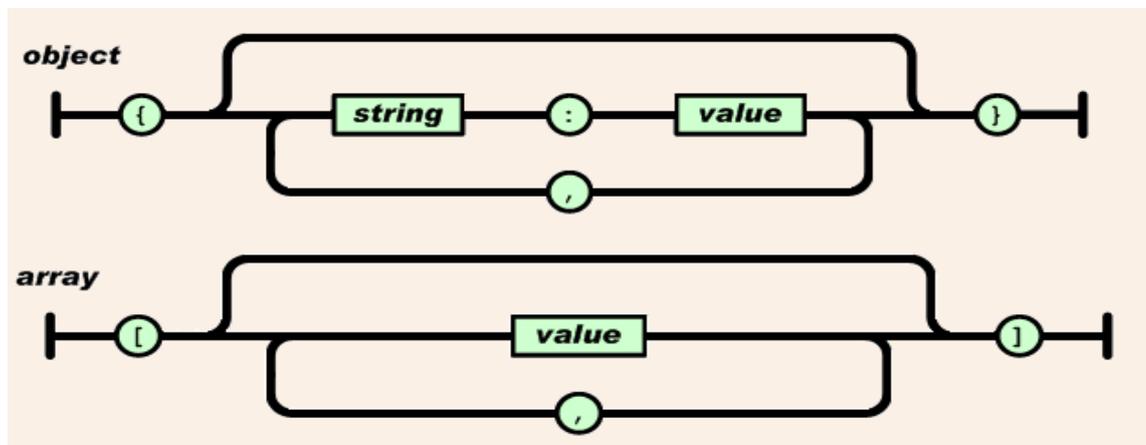


Figure III. 3: description des deux structures

### 4.8.2 Représentation du format JSON

L'API utilise le formalisme JSON pour l'échange de données. Nous l'appliquons à nos représentations. La figure suivante décrit un exemple de représentation d'une ressource utilisateur au format JSON.

```
"metadata": {
  "audio_properties": {
    "analysis_sample_rate": 44100,
    "bit_rate": 256000,
    "equal_loudness": 0,
    "length": 247.719177246,
    "lossless": 0,
    "replay_gain": -9.5851020813,
    "sample_rate": 44100,
    "codec": "mp3",
    "downmix": "mix",
    "md5_encoded": "35287c30b3b0bd514b85022efd09c6bf"
  },

```

Figure III. 4: Exemple d'un fichier JSON

## 5. Les principaux protocoles de communication

### 5.1 Le protocole TCP/IP

#### 5.1.1 Définition

Le protocole TCP/IP développé par le ministère de la défense américaine en 1981 propose l'évolution des concepts déjà utilisés en partie pour le réseau historique ARP Anet 1972 et est employé en très forte proportion sur le réseau internet au-delà de son aspect historique le protocole TCP/IP doit aussi son Indépendance vis-à-vis de tout constructeur informatique.

En réalité TCP/IP définit une suite de divers protocoles probabilistes appelé aussi modèle DOD (département of defense) pour la communication sur le réseau informatique notamment le protocole TCP et le protocole IP qui sont parmi les principaux protocoles de ce modèle. [10']

#### 5.1.2 TCP/IP et modèle OSI

##### 5.1.2.1 Découpage en couche

Le protocole TCP/IP étant antérieur au modèle OSI, il ne respecte pas réellement celui-ci cependant on peut faire grossièrement correspondre les différents services utilisés et proposés par TCP/IP avec le modèle OSI et obtenir aussi un modèle en 4 couches.

OSI	TCP/IP
7. application	Application
6. présentation	
5. session	
4. transport	Transport
3. réseau	Internet
2. liaison	
1. physique	Hôte-réseau

Les services des couches 1 et 2 (physique et liaison) du modèle OSI sont intégrés dans une seule couche (hôte-réseau) les couches 5 et 6 (session et présentation) n'existent pas réellement dans le modèle TCP/IP et leurs services sont réalisés par la couche application si besoin. [10']

#### 5.1.3 Le protocole IP

Le protocole IP (internet Protocol) assure le service attendu de la couche réseau du modèle TCP/IP son rôle est donc de gérer l'acheminement des paquets (issus de la couche transport) entre les nœuds de manière totalement indépendante même dans le cas où les paquets ont mêmes nœuds source et destination. Le protocole IP offre un fonctionnement non fiable et sans connexion, à base d'envoi/réception :

- Non fiable : absence de garantie que les datagrammes arrivent à destination, les datagrammes peuvent être perdus sans que la source ou la destination le sachent.
- Sans connexion : (mode non connecté) chaque datagramme est traité et donc acheminé de manière totalement indépendante des autres, le rôle d'IP est donc de déterminer le chemin entre les nœuds source et destination. [10']

## 5.2 Les protocoles TCP/UDP

### 5.2.1 Système client/serveur

La suite de protocole TCP/IP est généralement la mise en œuvre dans un système de communication où chaque machine tient un rôle précis, pour établir la communication l'une des machines doit débiter l'émission vers la seconde laquelle doit être en mesure de répondre à sa demande et donc être en attente d'une réception. Dans le cadre de la communication ponctuelle on distingue donc :

- La machine SERVEUR : en attente d'une réception provenant de n'importe quelle machine on doit être à l'écoute.
- La machine CLIENTE : réalise une émission en direction d'un serveur précis.

C'est toujours le client qui sera à l'initiation de la communication en s'adressant à un serveur apte à répondre à sa demande. [10']

### 5.2.2 Le protocole TCP

Le protocole TCP (transmission control Protocol) assure les services attendus de la couche transport du modèle TCP/IP son rôle est donc de gérer le fractionnement et le réassemblage en paquets des segments de données qui transitent via le protocole IP afin de fiabiliser la communication TCP doit donc aussi réordonner les paquets avant de les rassembler et doit aussi gérer les paquets erronés ou perdus, pour cela le TCP fonctionne en mode connecté en usant de deux mécanismes mettant en œuvre un principe de synchronisation /question/réponse /confirmation. [10']

### 5.2.3 Le protocole UDP

Le protocole UDP (user datagramme Protocol) assure les services attendus de la couche transport du modèle TCP/IP tout comme TCP son rôle est de gérer le fractionnement et le réassemblage en paquets des segments de données qui transitent via le protocole IP cependant UDP n'assure aucun autre service supplémentaire, pas de réordonnement pas de suivi de la communication à l'aide de l'accusé de réception pas de contrôle de flux. UDP fonctionne en mode non connecté c'est-à-dire qu'il ne fait que transporter les paquets de manière indépendante sans assurer la moindre cohérence entre eux. [10']

## 5.3 Le protocole HTTP

Le HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web. Il est utilisé pour échanger toute sorte de données entre client HTTP et serveur HTTP. C'est-à-dire que HTTP est une espèce de langage qui va permettre au client (vous, par le biais de votre navigateur par exemple) de communiquer avec un serveur connecté au réseau (le serveur HTTP installé sur le serveur d'un site, par exemple Apache). [11']

### 5.3.1 Requête HTTP

La structure de la requête http est décrite dans ce qui suit :

- **Une ligne de requête**

C'est une ligne précisant le type de document demandé, la méthode qui doit être appliquée, et la version du protocole utilisée. La ligne comprend trois éléments devant être séparés par un espace : La méthode, l'URL et la version du protocole utilisé par le client (généralement http 1.0). [12']

# CHAPITRE 3 : Les technologies web et Android

---

## ➤ Les champs d'en-tête de la requête

Il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...).

Chacune de ces lignes est composée d'un nom qualifiant le type d'en tête, suivi de deux points (:) et de la valeur de l'en-tête. [12']

## ➤ Le corps de la requête

C'est un ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire. Une requête HTTP a donc la syntaxe suivante (<crLf> signifie retour chariot ou saut de ligne) :

```
METHODE URL VERSION<crLf>
EN-TETE : Valeur<crLf>
.
.
.
EN-TETE : Valeur<crLf>
Ligne vide<crLf>
CORPS DE LA REQUETE
```

Voici donc un exemple de requête HTTP :

```
GET http://www.google.com HTTP/1.0
Accept : text/html
If-Modified-Since : Saturday, 15-January-2009 14:37:11 GMT
User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 7)
```

### 5.3.2 Les commandes http

Dans le protocole HTTP, une méthode est une Commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

# CHAPITRE 3 : Les technologies web et Android

Il existe de nombreuses méthodes, les plus courantes étant GET, HEAD et POST :

Méthode	Explication
<b>GET</b>	Cette méthode est la plus courante, il s'agit normalement d'une simple requête de téléchargement d'un document. Deux requêtes GET portant sur le même document devraient retourner des réponses sémantiquement identiques (certains en-têtes pouvant influencer sur le comportement du serveur, les réponses peuvent ne pas être totalement identiques). Aucune donnée à traiter ne peut être envoyée au serveur par cette méthode. Il est par contre possible d'ajouter les paramètres d'URL (aussi nommés paramètres GET). Le corps de la requête DOIT être vide, et le document spécifié dans la requête (la Page) est celui qui doit être retourné.
<b>HEAD</b>	Cette méthode ne demande que des informations sur la ressource, sans demander la ressource elle-même.
<b>POST</b>	Cette méthode est utilisée pour transmettre des données en vue d'un traitement à une ressource (le plus souvent depuis un formulaire HTML). L'URI fournie est l'URI d'une ressource à laquelle s'appliqueront les données envoyées. Le résultat peut être la création de nouvelles ressources ou la modification de ressources existantes. À cause de la mauvaise implémentation des méthodes HTTP (pour Ajax) par certains navigateurs (et la norme HTML qui ne supporte que les méthodes GET et POST pour les formulaires), cette méthode est souvent utilisée en remplacement de la requête PUT, qui devrait être utilisée pour la mise à jour de ressources.
<b>OPTIONS</b>	Cette méthode permet d'obtenir les options de communication d'une ressource ou du serveur en général.
<b>CONNECT</b>	Cette méthode permet d'utiliser un proxy comme un tunnel de communication.
<b>TRACE</b>	Cette méthode demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion.
<b>PUT</b>	Cette méthode permet de remplacer ou d'ajouter une ressource sur le serveur. L'URI fourni est celui de la ressource en question.
<b>PATCH</b>	Cette méthode permet, contrairement à PUT, de faire une modification partielle d'une ressource.
<b>DELETE</b>	Cette méthode permet de supprimer une ressource du serveur.

**Tableau III. 1: Types de commandes http**

## 5.3.3 Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

### ➤ Une ligne de statut

C'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace : La version du protocole utilisé, Le code de statut, La signification du code

### ➤ Les champs d'en-tête de la réponse

Il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en tête, suivi de deux points (:) et de la valeur de l'en-tête [12']

## ➤ Le corps de la réponse

Il contient le document demandé. Une réponse HTTP a donc la syntaxe suivante (<crLf> signifie retour chariot ou saut de ligne) :

```
VERSION-HTTP CODE EXPLICATION<crLf>
EN-TETE : Valeur<crLf>
.
.
.
EN-TETE : Valeur<crLf>
Ligne vide<crLf>
CORPS DE LA REPONSE
```

Voici donc un exemple de réponse HTTP :

```
HTTP/1.0 200 OK
Date : Sat, 15 Jan 2000 14:37:12 GMT Server : Microsoft-IIS/2.0
Content-Type : text/HTML
Content-Length : 1245
Last-Modified : Fri, 14 Jan 2000 08:25:13 GM
```

## 6. Bases de données

### 6.1 Définition

Une base de données (son abréviation est BD, en anglais DB, DataBase) est une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données. [13']

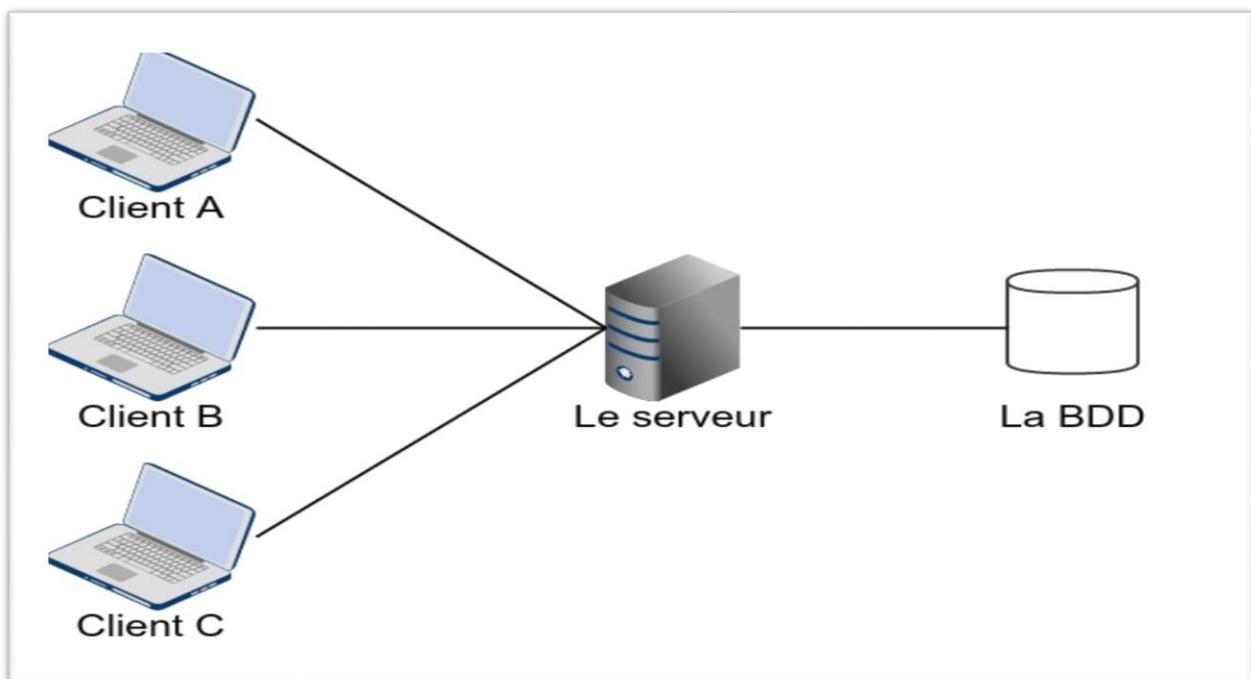


Figure III. 5: Architecture BD-serveur-clients

## 6.2 Utilité d'une base de données

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en plus nombreuses. Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien répartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau. L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément. [13']

## 6.3 La gestion des bases de données

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion de la base de données se fait grâce à un système appelé SGBD (système de gestion de bases de données) ou en anglais DBMS (Data Base management system). Le SGBD est un ensemble de services (applications logicielles) permettant de gérer les bases de données, c'est-à-dire :

- Permettre l'accès aux données de façon simple.
- Autoriser un accès aux informations à de multiples utilisateurs.
- Manipuler les données présentes dans la base de données (insertion, suppression, modification).

Le SGBD peut se décomposer en trois sous-systèmes :

- Le système de gestion de fichiers : il permet le stockage des informations sur un support physique.
- Le SGBD interne : il gère l'ordonnancement des informations.
- Le SGBD externe : il représente l'interface avec l'utilisateur.

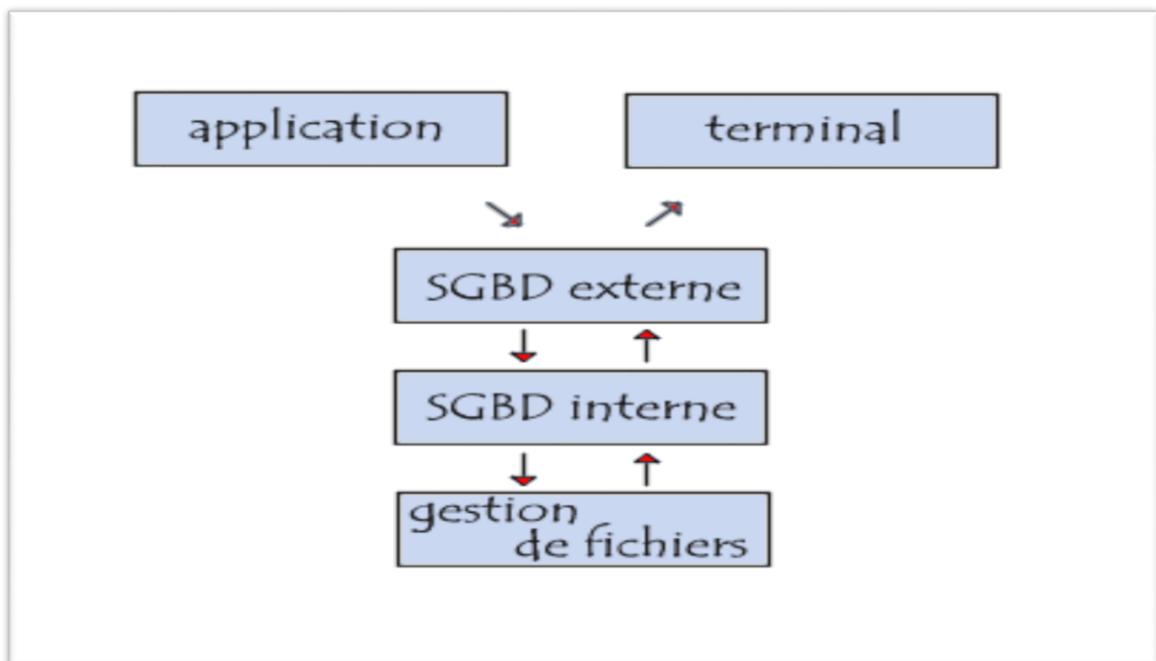


Figure III. 6: Architecture du SGBD

## 6.4 Les principaux SGBD

Les principaux systèmes de gestion de bases de données et Les plus connus sont les suivants :

- MySQL : libre et gratuit, c'est probablement le SGBD le plus connu. Nous l'utiliserons dans notre projet.
- PostgreSQL : libre et gratuit comme MySQL, avec plus de fonctionnalités mais un peu moins connu.
- SQLite : libre et gratuit, très léger mais très limité en fonctionnalités.
- Oracle : utilisé par les très grosses entreprises, sans aucun doute un des SGBD les plus complets, mais il n'est pas libre et on le paie le plus souvent très cher.
- Microsoft SQL Server : le SGBD de Microsoft. [13']

## 6.5 Définition de MySQL

MySQL est un système de gestion de bases de données relationnelles. Le SQL dans "MySQL" signifie "STRUCTURED QUERY LANGUAGE" : le langage standard pour les traitements de bases de données. MySQL est Open Source. Open Source (Standard Ouvert) signifie qu'il est possible à chacun d'utiliser et de modifier le logiciel. Tout le monde peut le télécharger sur Internet et l'utiliser sans payer aucun droit. Toute personne en ayant la volonté peut étudier et modifier le code source pour l'adapter à ses besoins propres. Toutefois, si vous devez intégrer MySQL dans une application commerciale, vous devez vous procurer une licence auprès de MySQL AB. Ce qui rend MySQL très intéressant pour les Webmasters est le nombre d'API (application program interface) dont il dispose. Vous pouvez en effet l'intégrer dans des applications écrites en : C, C++, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl.

## 6.6 Définition De phpMyAdmin

Le phpMyAdmin est un outil logiciel gratuit écrit en PHP, conçu pour gérer l'administration de MySQL dans le Web. Le phpMyAdmin soutient un large éventail d'opérations dans MySQL et MariaDB. Le phpMyAdmin permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export.

Ce système permet de sauvegarder facilement une base de données sous forme de fichier (.sql) et d'y transférer ses données, même sans connaître SQL. Les opérations fréquemment utilisées (bases de données, tables, colonnes, relations, index, utilisateurs, autorisations, etc. de gestion) peuvent être effectuées par l'intermédiaire de l'interface utilisateurs.

## 6.7 Tables de données, lignes et colonnes

Les données d'une base de données sont organisées sous la forme d'un ou plusieurs tables. Une table contient une collection de lignes stockées sur un support externe, généralement un disque, Une ligne est en elle-même une suite de valeurs, chacune d'un type déterminé, d'une manière générale une ligne regroupe des informations concernant un objet, un individu, un événement...etc. C'est-à-dire un concept du monde réel (externe à l'informatique) que nous appellerons parfois une entité ou un fait, l'ensemble des valeurs de même type correspondant à une même propriété des entités décrites s'appelle une colonne de la table. Chaque table est l'implémentation physique d'une relation entre les différentes colonnes. Chaque correspondance est définie par une ligne de la table.

## 7. L'univers ANDROID

### 7.1 Historique

L'histoire d'Android commence en octobre 2003 lorsque Andy Rubin, Riche Miner, Nick Sears et Chris White créent la société Android à Palo Alto (Californie). Google a racheté la société en août 2005. Deux ans plus tard, l'Open Hand Set Alliance est annoncée et Android devient officiellement open Source. La première version du SDK Android 1.0 a sorti en 2008 avec le premier téléphone sous Android (HTC Dream). [2]

### 7.2 Définition

Android est un OS (Operating system ou système d'exploitation en français) open source pensé pour les téléphones mobiles et développé par l'Open Hand Set Alliance (OHA) sous l'autorité de Google. L'OHA est composée d'environ 80 sociétés telles que Samsung, HTC, SFR, Orange, ASUS, QUALCOMM.

L'Android est basé sur un KERNEL Linux et possède une licence Apache Licence 2.0. Il sépare la couche matérielle de la couche logicielle, c'est-à-dire que n'importe quel téléphone Android peut faire fonctionner la même application et ainsi créer un écosystème riche pour les constructeurs, les utilisateurs et les développeurs. Android est aussi un Framework et en étant développeur, vous avez accès au SDK (Software Development Kit de développement) et à tout le code source de la plateforme. Ce qui vous permet de comprendre son fonctionnement, de créer des versions personnalisées de l'OS et bien sûr, de développer des applications. [2]

### 7.3 Versions d'Android

Le système d'exploitation Android évolue très rapidement chaque mois, Google publie des statistiques qui rendent compte de la répartition des différentes versions de son système d'exploitation. Dans ce qui suit on présente l'évolution des versions Android. [14]

#### 7.3.1 Android 1.0 et 1.1

À la date du 22 octobre 2008, le premier Smartphone sous Android débarque chez l'opérateur T-Mobile en Amérique, le HTC Dream (ou HTC G1). La grande innovation pour cette première version d'Android fut la barre de notifications, au travers d'un menu déroulant qui regroupe plusieurs affichages pratiques et rapidement accessibles. [15]

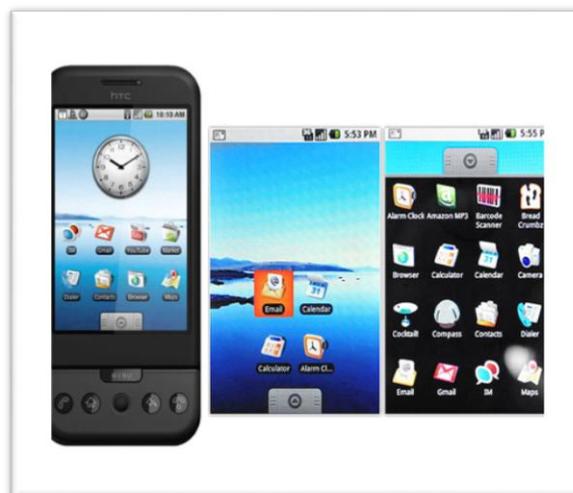


Figure III. 7: Interface de la Version 1.0

## 7.3.2 Android 1.5

La mise à jour 1.5 « **CupCake** » a été publiée le 30 avril 2009, améliorant l'interface graphique d'Android. CupCake était donc la troisième itération de l'O.S., mais surtout la venue du premier clavier tactile, et du presse-papier. Avant cela, il était impossible de faire un copier/coller d'un texte, c'est dire combien cette version fut attendue. C'est à partir de cette version qu'on peut enregistrer des vidéos sur le Smartphone, envoyer celles-ci directement sur You Tube ou Picasa et profiter de quelques ajouts mineurs dans les contacts et le journal d'appels.



Figure III. 8: Interface de la version 1.5

## 7.3.3 Android 1.6

Si la version 1.5 était une mise à jour majeure, celle qui a suivi l'était moins. Android 1.6 « **Donut** » offrait tout de même quelques nouvelles fonctionnalités comme la prise en charge du réseau CDMA, le support d'autres définitions d'écran. Auparavant, il ne fallait compter que sur une définition maximum de 480 x 320 pixels (en mode portrait) et la mise à jour 1.6 aura permis de gérer toutes les définitions allant jusqu'à la HD, avec une définition de 1280 x 720 pixels, un grand pas en avant pour le système. Cette version apportait également la fonction de recherche directement en local sur votre appareil Android, en comptant également sur les recherches internet grâce à Google Search.

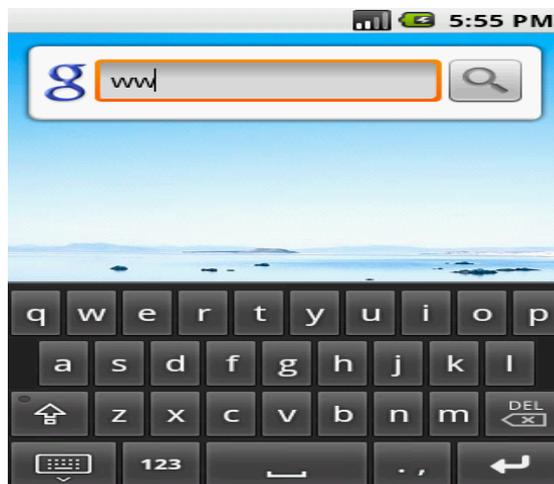


Figure III. 9: Interface de La Version 1.6

## 7.3.4 Android 2.0 et 2.1

Google a frappé fort à la fin du mois d'octobre 2009, en publiant la version Android 2.0 « **Eclair** ». Il aura surtout été possible grâce à cette mise à jour d'associer plusieurs comptes Google au même appareil. Celle-ci aura été majeure et aura apporté beaucoup de fonctionnalités intéressantes, comme l'amélioration du clavier tactile, Fonds d'écrans animés, Prise en charge du Bluetooth 2.1, Arrivée de Google Maps et des nouvelles fonctions pour l'appareil photo (flash, zoom digital, balance des blancs, effets de couleurs et focus macro).



Figure III. 10: Interface des versions 2.0 et 2.1

## 7.3.5 Android 2.2 et 2.3

Le 20 mai 2010 arrivait la version Android 2.2 « **Froyo** », De nouveaux raccourcis ont été ajoutés aux côtés du lanceur d'applications : le téléphone et le navigateur natif, la mise à jour proposait pour la galerie avec une meilleure interface pour regarder vos photos, des effets de transition, ce qui rendait l'O.S. encore plus convivial.

Le 6 décembre de la même année, Google publie la version 2.3 « **Gingerbread** ». Elle n'aura pas apporté grand-chose, mais surtout des améliorations sur la prise en charge et le support de la VoIP.



Figure III. 11: Interface des versions 2.2 et 2.3

## 7.3.6 Android 3.0

Dans toute l'histoire de l'OS, il existe une mise à jour, la 3.0 « **Honeycomb** », qui fut réservée aux tablettes Android.



Figure III. 12: Interface de la version 3.0

## 7.3.7 Android 4.0

C'est le 19 octobre 2011, l'arrivée de la version 4.0 « **Ice Cream Sandwich** », compatible avec les Smartphones et les tablettes, venues avec les nouveautés suivantes :

- Ajout de dictionnaires dans le clavier virtuel.
- Application photo améliorée, rapidité de capture quasiment en mode rafale.
- Capture d'écran native.
- Consultation de son trafic data, pour ne plus avoir de hors forfait à votre prochaine facture.
- Création plus facile de dossiers d'applications.
- Écran de verrouillage repensé, sous forme d'un cadenas qui permet de lancer des applications en raccourci.
- Mise à jour du navigateur (ouvertures de plusieurs pages, navigation privée)
- Prise en charge de l'enregistrement vidéo Full HD 1080p.



Figure III. 13: Interface de la version 4.0

## 7.3.8 Android 4.1 et 4.2

« **Jelly Bean** » est le nom d'Android, annoncé le 27 juin 2012. Le premier appareil équipé de la mise à jour 4.1 fut la NEXUS 7. D'abord, le « Project Butter » a triplé la vitesse de rafraîchissement du système pour atteindre 16 millisecondes, un record. Le système d'Apple avait donc bien du souci à se faire, n'étant plus le seul à être aussi réactif. Le 13 novembre 2012 sortait la version suivante, Android 4.2, toujours une Jelly Bean.

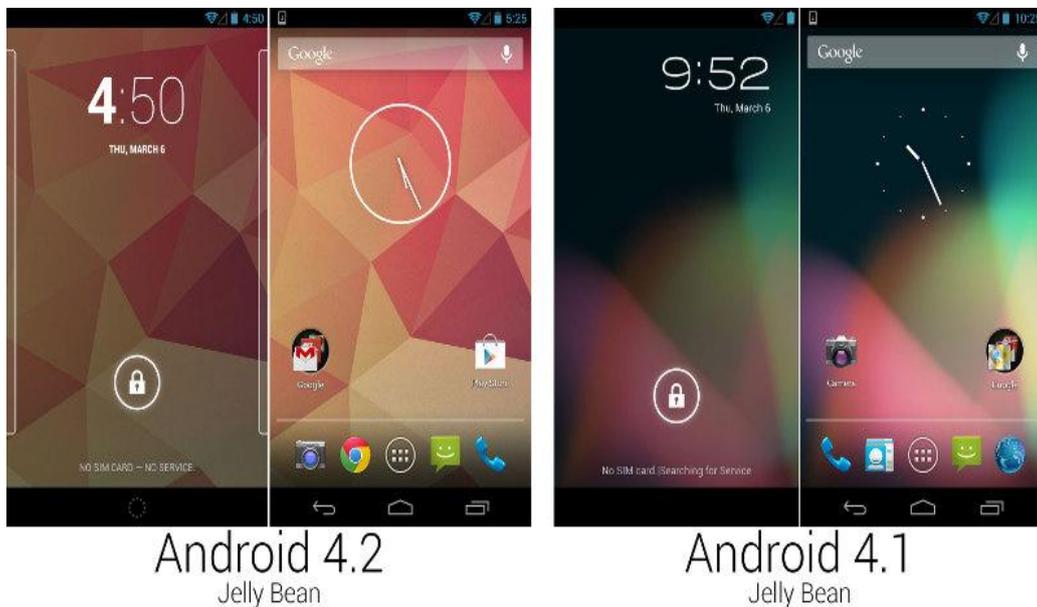


Figure III. 14: Interface des versions 4.1 et 4.2

## 7.3.9 Android 4.4

Elle est sortie le 4 novembre 2013, Le nom d'origine était Key Lime Pie, mais grâce à un accord avec Nestlé, Google l'a renommé « **KitKat** ». Au niveau des nouveautés, on peut clairement voir que KitKat consomme beaucoup moins de RAM que Jelly Bean. D'autres améliorations de design font partie des nouveautés, comme la barre de notification transparente.



Figure III. 15: Interface de la version 4.4

## 7.3.10 Android 5.0

le 15 octobre 2014 Google ont annoncés l'Android « **Lollipop** » la version 5 du système d'exploitation mobile Android, pour une disponibilité publique le 3 novembre 2014, Les changements les plus importants d'Android 5.0 sont sa disponibilité sur les nouvelles plateformes Android TV et Androïde Auto, le renouvellement de l'interface utilisateur nommée Matériel Design, l'amélioration du système de notifications, l'environnement d'exécution ART qui remplace officiellement Dalvik pour améliorer les performances, ainsi que l'amélioration de l'autonomie de la batterie via le projet Volta.[16']



Figure III. 16: Interface de la version 5.0

## 7.3.11 Android 6.0

En octobre 2015 la version 6.X du système d'exploitation mobile Android développé par Google et lancée .Nommée «**Marshmallow**» .Les fonctionnalités mises en lumière dans cette nouvelle version sont l'introduction d'un système d'autorisations pour les applications :accès aux contacts, localisation, les liens profonds entre les applications, le paiement sans contact via NFC , le support natif de la reconnaissance des empreintes digitales ainsi que des options pour améliorer l'autonomie et "App Standby" qui fait hiberner les applications rarement utilisées.[17']



Figure III. 17: Interface de la version 6.0

# CHAPITRE 3 : Les technologies web et Android

## 7.3.12 Android 7.0

En aout 2016, la version 7.0 appelée « Nougat » est Lancée par Google. Parmi les nouvelles fonctionnalités introduites dans cette version, il y a le multi fenêtrage qui permet d'afficher plusieurs applications en même temps, la possibilité de paramétrer l'utilisation du téléchargement de données via le réseau cellulaire et il est possible de poser des actions sur des notifications sans accéder aux applications qui les ont émises.



Figure III. 18: Interface de la version 7.0



Figure III. 19: L'évolution d'Android

## 7.4 Android et les langages de programmation

Le développement Android connaît un véritable essor ces dernières années, Pour développer sous Android et créer une application il faut connaître les langages de programmation adéquats. [18']

### 7.4.1 Android et java

Java est le langage Android le plus utilisé dans le développement mobile. L'un de ses plus grands avantages est que les logiciels créés avec ce langage peuvent être facilement installés et exécutés sur différents systèmes d'exploitation, que ce soit Windows, Mac OS, Linux ou autre. Avec un petit coup de main de Google, qui fournit l'environnement de développement Android Studio et le kit de développement SDK pour les développements d'application en Java vous pourrez créer une application Android bien plus complexe.



Figure III. 20: Logo JAVA

### 7.4.2 Le XML

C'est un langage de balisage utilisé pour gérer l'affichage des contenus sur l'écran. Il n'est pas indispensable pour créer une application Android, mais il facilite le développement en permettant de séparer l'affichage des algorithmes. Avec XML, on gagne du temps et on simplifie le code de l'application, ce qui permet d'éviter des erreurs. [18']



Figure III. 21: Logo XML

## 7.4.3 Android et C/C++

Il est possible développer des applications Android en utilisant le langage C/C++ qui seront exécutées directement par le système d'exploitation Linux embarqué. Androide fournit le kit de développement NDK pour les développements d'application en C/C++ il est Utilisé dans le développement de jeux 2D/3D se basant fortement sur la librairie Open GL.



Figure III. 22: Logo C/C+

## 7.5 Définition d'une application mobile

Une application mobile est un logiciel applicatif (exploitable directement par l'utilisateur) pour divers services, conçue pour être adaptées aux différents environnements techniques des terminaux mobiles (un téléphone, un Smartphone, une tablette...) à des caractéristiques matérielles et logicielles bien définies elles sont souvent téléchargées et installées depuis une boutique (store) de façon gratuite ou payante tel que l'APP store pour des téléphones sous Ios, ou Google Play pour des téléphones sous Android, exécutable à partir du système d'exploitation du terminal mobile.

## 7.6 Objectifs, et prospectives des applications mobiles

On recense 3 principaux objectifs pouvant être atteints grâce à une application mobile :

- Les applications mobiles visent d'abord à améliorer la productivité et à faciliter la récupération d'informations telles que le courrier électronique, calendrier électronique, contacts, marché boursier et informations météorologiques.
- Pour les professionnels ça permet de renforcer la relation et garder le lien avec ses Interlocuteurs, d'atteindre une cible précise, et d'optimiser ses couts.
- Disposer d'une solution métier : une application conçue dans cette perspective permet à l'utilisateur final de gérer ses stocks et commandes, et de piloter ou carrément D'automatiser ses interventions sur le terrain. [3]

## 7.7 Types d'applications

Autre que le classement des applications selon le service proposé (jeux, consultation de courrier électronique, écouter de la musique...etc.), il est aussi possible selon une perspective de développeur de distinguer les applications mobiles par leurs environnements d'exécution et les outils ayant contribués à leurs développements.

## 7.7.1 Les applications natives

Une application native est une application qui est développée spécifiquement pour une plateforme, grâce aux outils spécifiquement conçus pour la plateforme. Aussi, s'il est prévu de développer une application native pour Android et Ios, il convient en réalité de développer deux applications bien distinctes qui ne pourront malheureusement pas partager la moindre ligne de code. [3]

## 7.7.2 Application sous Android

Pour développer une application sur la plateforme de Google, le langage de programmation utilisé est le JAVA. Les applications sont alors développées à l'aide du SDK Android. Suivant le besoin, on pourra également le mixer à l'utilisation du C++ et du NDK. L'emploi d'un IDE est aussi de mise, ça peut aussi bien être Eclipse ou Android studio. [3]

## 7.7.3 Application sous Ios

Pour développer une application sur la plateforme d'Apple, le langage de programmation utilisé historiquement est objective-c. Depuis quelques temps, il est également possible d'utiliser un nouveau langage : le Swift. X-Code est l'IDE par lequel jurent les développeurs d'Ios. [3]

## 7.7.4 Les applications sous Windows Phone

Pour développer une application sur la plateforme de Microsoft, il convient d'utiliser un langage de la famille .NET ce qui offre aux développeurs plusieurs possibilités par exemple le VB.NET, le C# ou le C++. Il est également possible d'utiliser Win JS et ainsi développer une application à l'aide de technologies web. Développer une application Windows Phone ne fait pas exception à la règle de l'IDE. Aussi, il convient généralement d'utiliser Visual Studio. [3]

## 7.7.5 Les applications multiplateformes

Désignées aussi sous le nom d'applications hybrides. Les applications Cross-Platform s'opposent donc aux applications natives. Une application cross -Platform est une application qui une fois développée est capable de s'exécuter sur de nombreuses plates -formes. Les applications cross-Platform sont souvent développées à l'aide de technologies web (HTML, CSS, JavaScript). Les Framework les plus connus et utilisés étant probablement Phone gap, IONIC, XAMARIN, et apache CORDOVA. L'avantage de ce type d'application est qu'un seul et unique code peut être développé sur différentes plateformes, ce qui permettra de réduire les couts de développement et d'homogénéiser l'expérience utilisateur. [3]

## 7.8 Fonctionnement d'une application mobile

### 7.8.1 Architecture

A l'exception des jeux et des applications de bureautique diverses. Dans la plupart des cas les applications mobiles sont basées sur une architecture dite client-serveur. Deux actions sont récurrentes sur les terminaux mobiles ou ce genre d'applications est installé : afficher des informations et en recueillir. Au final une application mobile ne traite que très peu de donnée, elle délègue le travail à ce qu'on appelle un serveur. Le serveur tient le rôle principal dans le cadre d'une application mobile. C'est lui qui alimente les applications en données et qui sauvegarde les données qui auraient besoin d'être persistées comme par exemple le compte d'un utilisateur (identifiant, mot de passe, données personnelles).[3]

## 7.8.2 Communication client-serveur

La plupart du temps, les échanges entre un serveur et une application mobile se font à travers un web service, également appelé API. Bien que l'on distingue deux grandes familles :

- Les web services REST.
- Les web service SOAP.

C'est bien les web services REST qui sont le plus souvent utilisés.

La communication est alors extrêmement simple. L'application mobile se contente de faire une requête HTTP au web service qui va alors s'exécuter, remplir son rôle (par exemple sauvegarder des données ou en récupérer dans une base de données) puis renvoyer une réponse à l'application mobile. Cette réponse, en plus de contenir les informations demandées contient souvent un certain nombre d'informations relatives au déroulement de la demande.

Les données renvoyées à l'application mobile sont formatées d'une manière très précise permettant à l'application de les traiter facilement. Souvent, les données sont formatées à l'aide du JSON car il a l'avantage d'être facile à écrire et léger à échanger à travers le web. Mais il est tout à fait possible d'utiliser une autre technologie come XML, ou encore YAML.

A noter que les web services peuvent être écrits dans n'importe quel langage de Programmation capable de s'exécuter sur un serveur. Parmi les plus utilisés, on peut citer par exemple : PHP, JAVA, Python, Perl, Ruby, JS, ou encore C#. [3]

## 7.9 Les requêtes HTTP sur Android

Très rares sont les applications Android n'ayant pas besoin d'effectuer un appel HTTP à un serveur distant. Cependant la bibliothèque standard d'Android ne fournit que des outils pour travailler à un niveau très bas, exigeant que l'on gère tous les effets de bords nous-mêmes, à un niveau plus précis (et donc couteux en développement) que la plupart des développeurs ont besoin de se soucier. Nous allons présenter quelque manière sans librairies et avec librairies de faire une requête http.

### 7.9.1 Requête HTTP Sans librairie

#### 7.9.1.1 Http URL Connection

La classe `http Url Connection` du package `java.net`. Permet à nos appareils Android d'adopter les caractéristiques d'un client HTTP léger. Son fonctionnement est conditionné à des versions supérieures à Gingerbread. Pour les versions précédentes, vous devez utiliser le client `Apache HttpClient`.

Avec cette classe, nous pouvons recevoir et envoyer des informations via le web, ce qui améliorera nos applications Android.

Les utilisations de cette classe suivent un modèle :

- Obtenez un nouveau `http URL Connection` en appelant `URL.openConnection ()` et en convertissant le résultat en `http URL Connection`.
- Préparer la demande. La propriété principale d'une requête est son URI. Les en-têtes de demande peuvent également inclure des métadonnées telles que les informations d'identification, les types de contenu préférés et les cookies de session.

- Télécharger éventuellement un corps de requête. Les instances doivent être configurées avec `setDoOutput (true)` si elles incluent un corps de requête. Transmettez des données en écrivant dans le flux renvoyé par `getOutputStream ()`.
- Lisez la réponse. Les en-têtes de réponse incluent généralement des métadonnées telles que le type et la longueur du contenu du corps de la réponse, les dates modifiées et les cookies de session. Le corps de la réponse peut être lu à partir du flux renvoyé par `getInputStream ()`. Si la réponse n'a pas de corps, cette méthode renvoie un flux vide.
- Déconnecter. Une fois que le corps de la réponse a été lu, le `HttpURLConnection` doit être fermé en appelant `disconnect ()`. La déconnexion libère les ressources détenues par une connexion afin qu'elles puissent être fermées ou réutilisées [19']

## 7.9.2 Requête HTTP avec librairie

### 7.9.2.1 Méthode OK.HTTP

OK.HTTP est une librairie Java développée par Square afin de faciliter, mais aussi d'accélérer les appels réseaux. Il respecte la norme HTTP, mais aussi HTTP/2 et SPDY, ce qui en fait aujourd'hui un outil plus performant que celui déjà présent sur Android.

```
1 | compile 'com.squareup.okhttp:okhttp:2.4.0'
```

- Importer OKHTTP:

Comme pour toutes les librairies, importez OKHTTP depuis votre fichier `build.gradle` [20']

### 7.9.2.2 Méthode Volley

Présenté par Google lors de la conférence Google I/O 2013 qui s'est déroulé du 15 au 17 mai, Volley est une librairie HTTP qui permet facilement de faire des appels réseaux rapides sur Android. Disponible à partir d'Android 2.2 (API Level 8 – Froyo), elle utilise les méthodes GET / POST / PUT / DELETE du protocole HTTP et est particulièrement efficace pour des téléchargements courts et rapides tels que le téléchargement d'images ou de données utilisateurs. [21']

En effet, l'un des nombreux avantages de Volley est que tous les résultats des requêtes sont automatiquement sauvegardés en cache (peut-être désactivé) afin de pouvoir être réutilisé, si besoin, plus tard. De plus, Volley offre la possibilité de :

- Hiérarchiser des requêtes : les prioriser afin de charger leurs contenus en fonction de leur rang.
- Programmer un groupe de requêtes.
- Implémenter ses propres mécanismes de `retry / backOff`.
- Annuler une ou plusieurs requêtes en cours.
- Support natif du JSON.

# CHAPITRE 3 : Les technologies web et Android

- **Prérequis**

Pour utiliser cette librairie au sein de nos projets, deux solutions s'offrent à nous. On peut directement passer par Gradle, qui est intégré à Android Studio et qui permet de gérer les modules et dépendances de nos projets de manière très simple. Pour cela, il nous suffit d'ouvrir le fichier build .Gradle du module principale de notre application et d'ajouter la ligne suivante dans le nœud dependencies.

```
1 | compile 'com.android.volley:volley:1.0.0'
```

Ce qui devrait nous donner :

```
1 | dependencies {  
2 |   ...  
3 |  
4 |   compile 'com.android.volley:volley:1.0.0'  
5 |  
6 |   ...  
7 | }
```

Sinon, nous pouvons récupérer le répertoire Volley directement sur GitHub et l'importer dans votre projet en tant qu'Android Library module. [21']

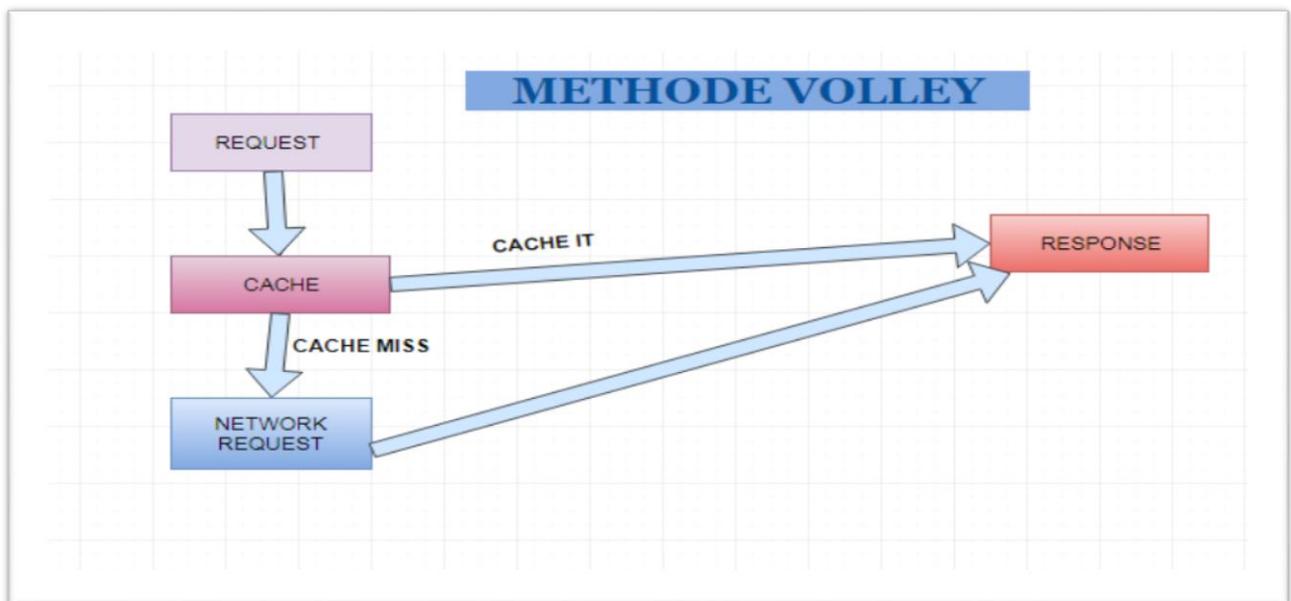


Figure III. 23: Le schéma de base de la méthode Volley

## 8. Conclusion

Dans ce chapitre nous avons détaillé tout ce qui concerne les technologies web et l'Android, commençant par citer les différents serveurs web les langages de programmation web en suite nous avons parlé sur les bases de données et enfin nous avons présenté les versions Android, les requêtes dans l'Android et les langages utilisés pour développer sous Android, dans le dernier chapitre nous allons entamer la partie pratique où on parlera de notre projet étape par étape.

# **Chapitre 4 :**

# **Réalisation du système**

# **de télémétrie**

# CHAPITRE 4 : Réalisation Du Système de télémétrie

## 1. Introduction

Ce chapitre est dédié à la partie conception et réalisation de notre système de télémétrie. Il est composé de 4 parties :

Nous allons parler dans la 1 ère partie de serveur où nous avons hébergé notre base de données. De plus, nous parlons aussi des scripts web (PHP, SQL, Java) que nous avons intégré dans le serveur pour interagir avec le circuit ARDUINO d'une part et l'application Android et le site web d'une seconde part.

Ensuite, nous passons à la partie programmation du dispositif ARDUINO après cela on entame la réalisation de l'application Android et le site web avec l'aperçu de leurs interfaces.

Mais avant cela nous présenterons notre système de télémétrie dans une petite description. Ainsi que les outils que nous avons utilisés dans notre travail.

## 2. Présentation de notre système de télémétrie

Notre système se base sur une architecture client/serveur. Tous les traitements effectués transitent par le serveur, donc il joue le rôle d'un noyau et aussi le centre de notre système.

Des scripts PHP en exécution permanente sur le serveur permettent de récupérer les données des capteurs collectées par l'ARDUINO et les stockées dans une base de données MySQL. Et aussi convertir les données enregistrées dans la base de données en un fichier JSON pour faciliter la récupération de ces données via l'application Android, et la visualisation de toutes les données sur un site web.

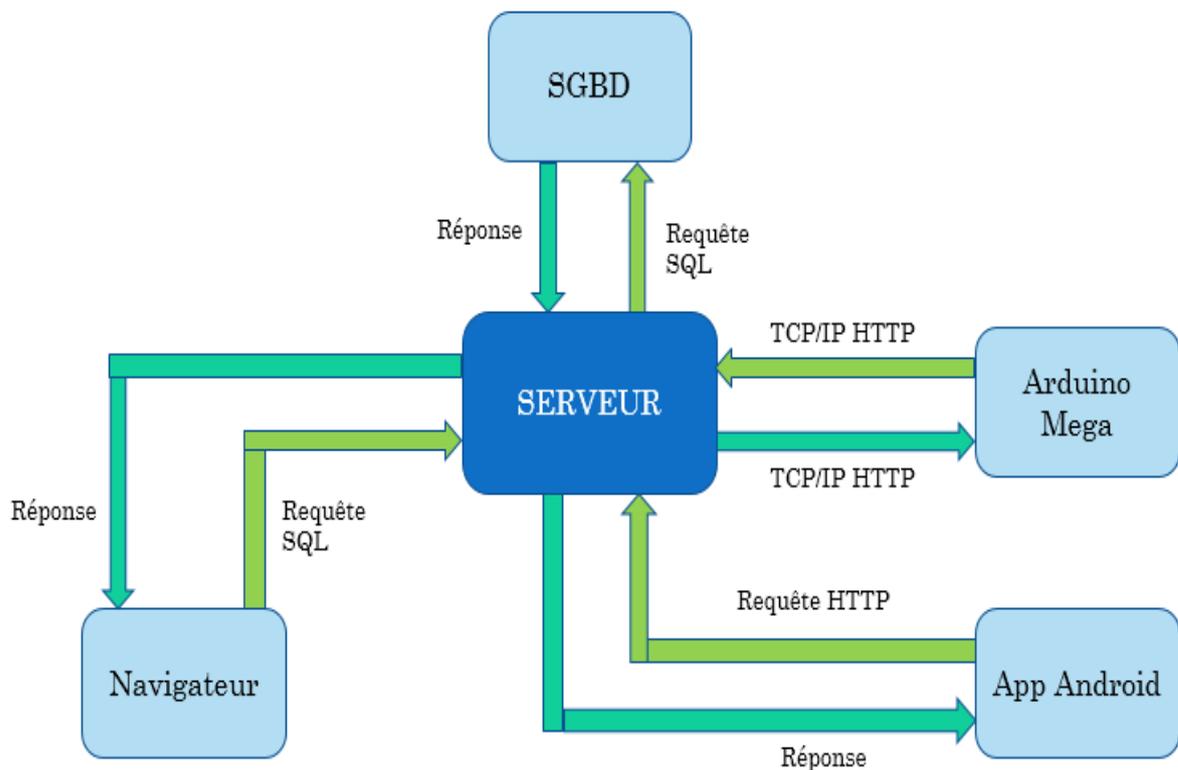


Figure IV. 1: schéma de notre système de télémétrie

## 2.1 Les étapes du travail

Dans notre projet nous avons suivi les 4 étapes suivantes :

- La collecte des données depuis les capteurs.
- Le transfert de ces données vers une base de données.
- La récupération de ces données par une application Android.
- La visualisation de ces données sur un site web pour l'analyse et le filtrage par date et heure.

## 2.2 Les outils et logiciels utilisés durant ce travail

- La carte ARDUINO Méga et le Shield wifi ESP8266.
- Logiciel ARDUINO ide (ARDUINO 1.8.5).
- Logiciel Fritzing.
- Breadboard, Jumper male ET female, des résistances.
- Les Capteurs de température(LM35), de Gaz et des fumées (MQ-2).
- Serveur Web.
- Android Studio.

## Partie I : Serveur web et création d'une base de données

### 1. Hébergement de notre serveur web de Télémétrie

Nous avons hébergé un serveur web pour que nous puissions stocker nos données collectées depuis notre ARDUINO méga dans une base de données MySQL et créer un site web pour la visualisation et l'analyse des données. Pour cela nous avons choisi l'hébergeur 000webhost.com qui propose un service d'hébergement gratuit et sans publicité. C'est probablement l'unique service d'hébergement sur le web s'auto finançant par un système de donations d'utilisateurs.



Figure IV. 2: Interface D'accueil du 000webhost

## 2. Création de la base de données

Nous avons donc créé notre base de données, que nous appelons télémétrie en utilisant l'interface PhpMyAdmin. Après la connexion sur notre serveur nous allons arriver à une page similaire à celle-ci :

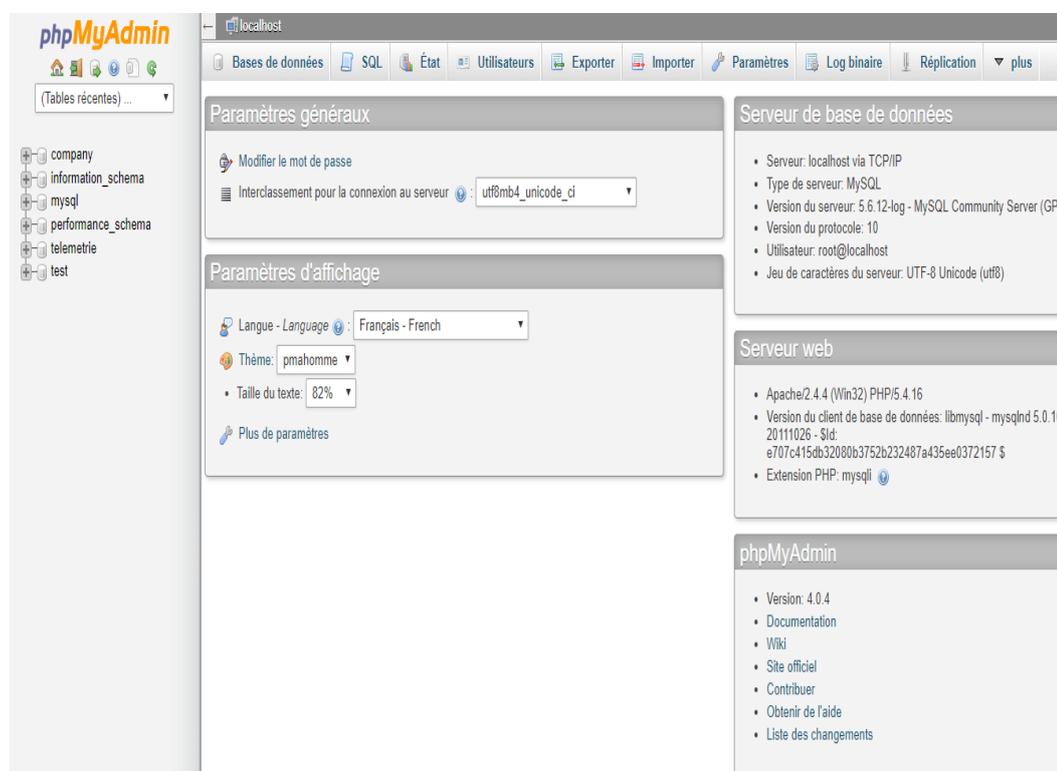
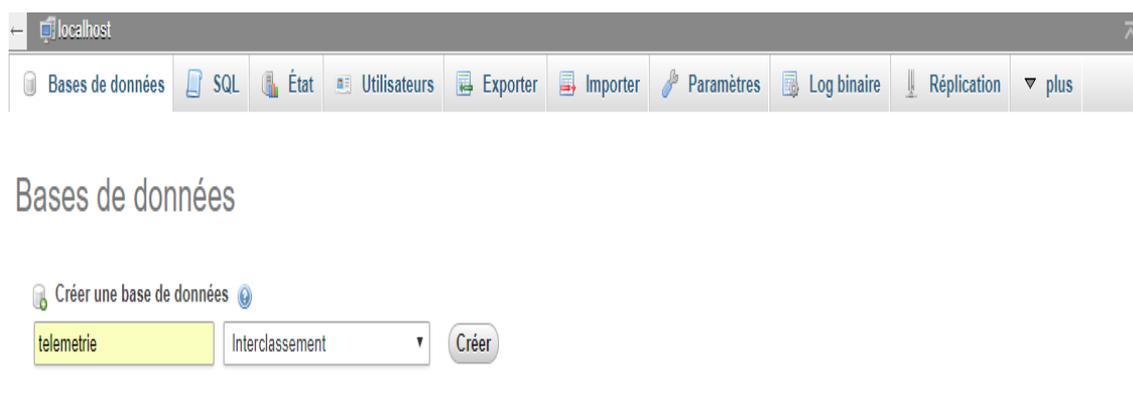


Figure IV. 3: interface phpMyAdmin

- Nous cliquons sur base de données et nous saisissons le nom de notre base de données dans le champ spécifique puis nous cliquons sur le bouton créer.



### 2.1 Création des tables SQL

D'abord, il faut sélectionner la base de données < télémétrie > dans laquelle nous travaillons dans la liste < Bases de données > sur la gauche. Puis, nous indiquons un nom et le nombre de champs pour la table et de même manière que la création de la base, on clique sur exécuter, des champs à remplir vont s'afficher. Nous remplissons les champs avec les paramètres qui correspondent et nous cliquons sur sauvegarder.

# CHAPITRE 4 : Réalisation Du Système de télémétrie

Dans notre projet nous avons créé deux tables :

- La première table sera dédiée pour les données du capteur de gaz <LM335>.

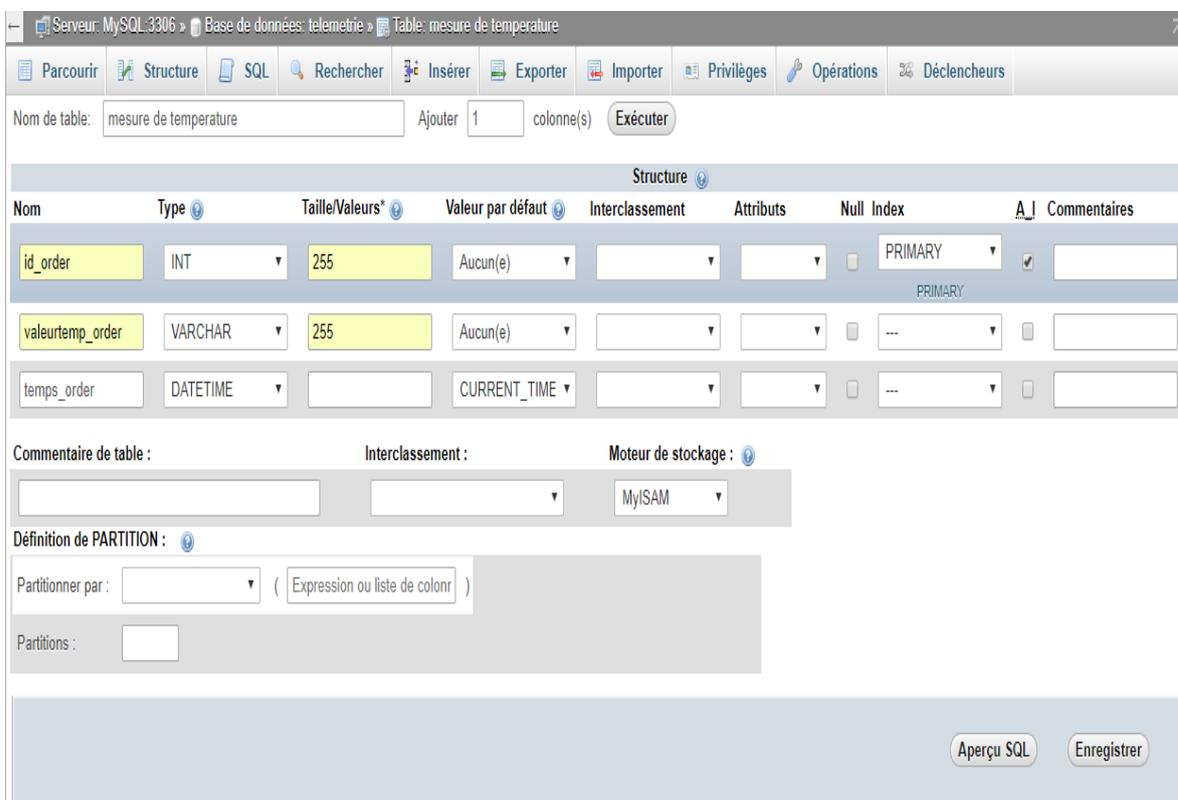


Figure IV. 4: création de table <mesure de température>.

- La deuxième table va charger les données du capteur de gaz < MQ-2 >.

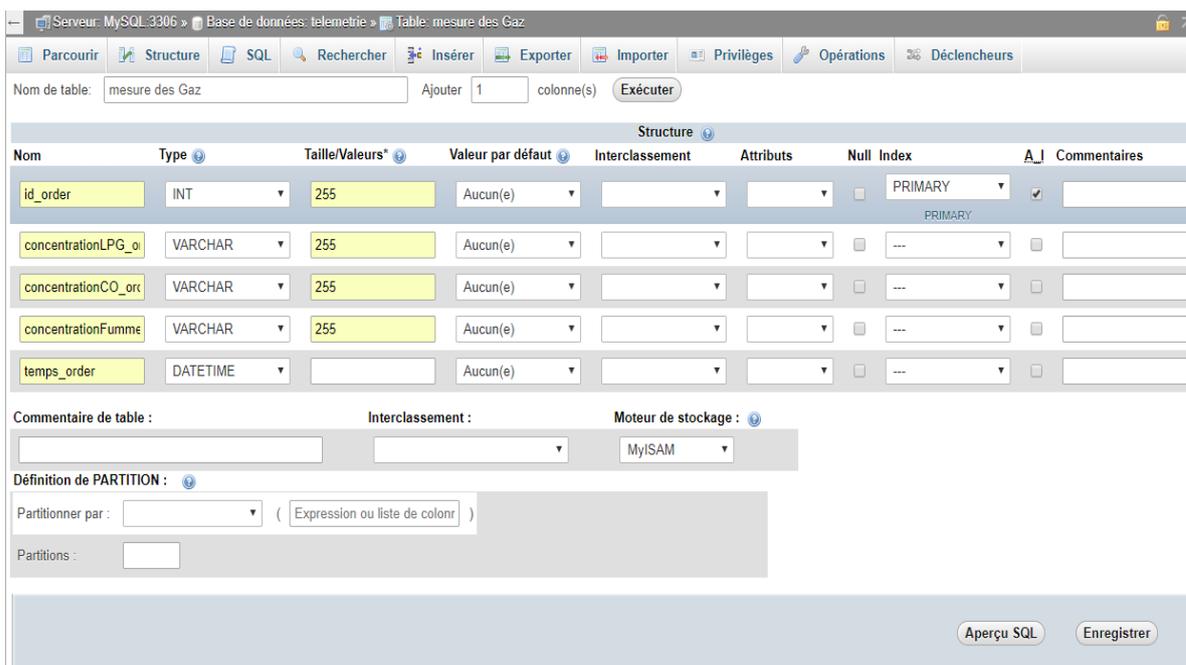


Figure IV. 5: création de table <mesure des Gaz>

## 3. Scripts web PHP et SQL dans le serveur

Il convient de signaler que les requêtes échangées entre le circuit ARDUINO et la base de données d'une part et l'application Android avec la base de données dans une seconde part sont établis en utilisant des scripts PHP et des scripts SQL.

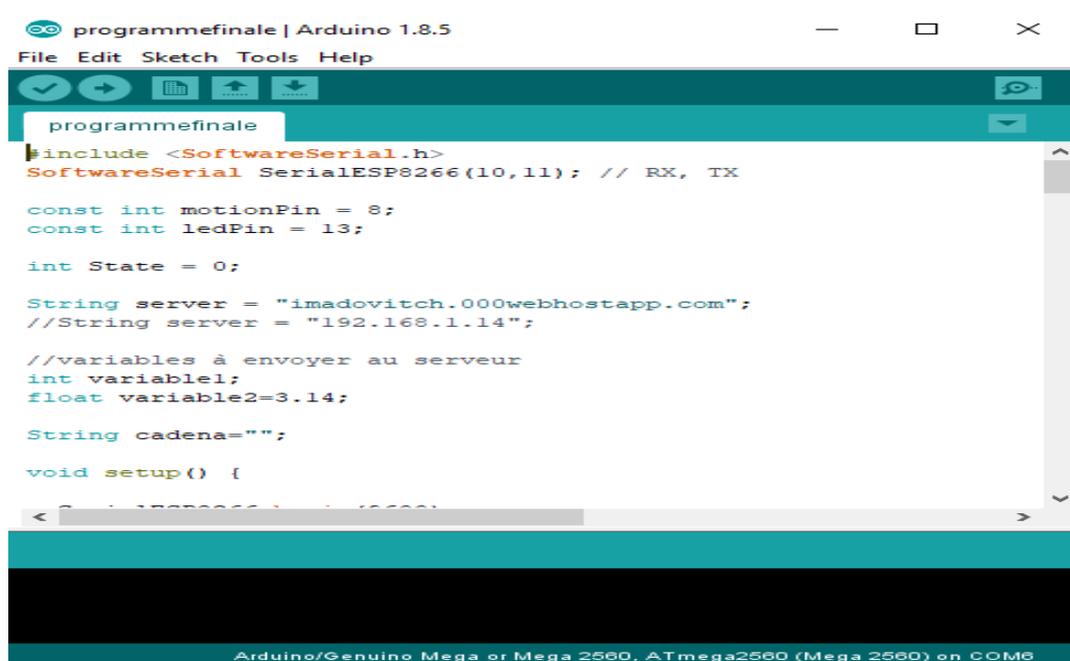
En ce qui concerne la communication entre la carte ARDUINO et la base de données hébergée dans le serveur web, nous avons intégré des scripts PHP pour faire la récupération des données collectées et des scripts SQL pour les stocker dans la base de données.

Et pour la communication entre l'application et le circuit ARDUINO des scripts PHP se chargent de convertir les informations stockées dans la base de données en un format JSON pour que l'application les récupère en étant des fichiers JSON.

## Partie II : Programmation du microcontrôleur ARDUINO

### 1. L'environnement de travail

Comme décrit précédemment dans le deuxième chapitre la programmation et l'exploitation de la carte ARDUINO méga sera faite dans le logiciel open-source ARDUINO.



```
programmefinale | Arduino 1.8.5
File Edit Sketch Tools Help
#include <SoftwareSerial.h>
SoftwareSerial SerialESP8266(10,11); // RX, TX

const int motionPin = 8;
const int ledPin = 13;

int State = 0;

String server = "imadovitch.000webhostapp.com";
//String server = "192.168.1.14";

//variables à envoyer au serveur
int variable1;
float variable2=3.14;

String cadena="";

void setup() {
```

Figure IV. 6: Vue sur le programme téléversé dans la carte ARDUINO Méga

### 2. Récupération des données depuis les capteurs

La carte ARDUINO va se charger de la lecture de la température, et la concentration de gaz et de CO et de fumée. Les capteurs LM335 et MQ-2 sont branchés sur les bornes de lecture analogique A0 et A1 et mesurent ces deux paramètres puis ils vont donner des valeurs sous forme de niveaux de voltage en mV.

Ces valeurs vont être converties en degrés Celsius (C°) pour le capteur de température et en PPM (partie par millions) pour le capteur de gaz en utilisant des formules précises.

# CHAPITRE 4 : Réalisation Du Système de télémétrie

---

- Dans ce qui suit un extrait du code utilisé pour relever les valeurs de température et de gaz depuis les capteurs :

```
void setup() {
  Serial.begin(9600);
}

void loop() {

  variable=analogRead(A0); //Port analogique pour lire la temperature
  float temperature = ((variable*5.0)/10.24)-273.15; // La conversion
  Serial.print(temperature);
  Serial.println(" °C");
  delay(1000); // Le délais en milli seconde entre chaque nouvelle valeur
}
```

```
#include <MQ2.h> //Déclaration de la bibliotheque MQ-2

int pin = A1; //Lecture des mesures par la broche A1
int lpg, co, smoke; //Déclaration des trois parametres de gaz

MQ2 mq2(pin);
void setup(){

  Serial.begin(9600);

  mq2.begin();
}

void loop(){
  float* values= mq2.read(true);

  lpg = mq2.readLPG(); //Valeur du LPG

  co = mq2.readCO(); //Valeur du CO

  smoke = mq2.readSmoke(); //Valeur de la fumée

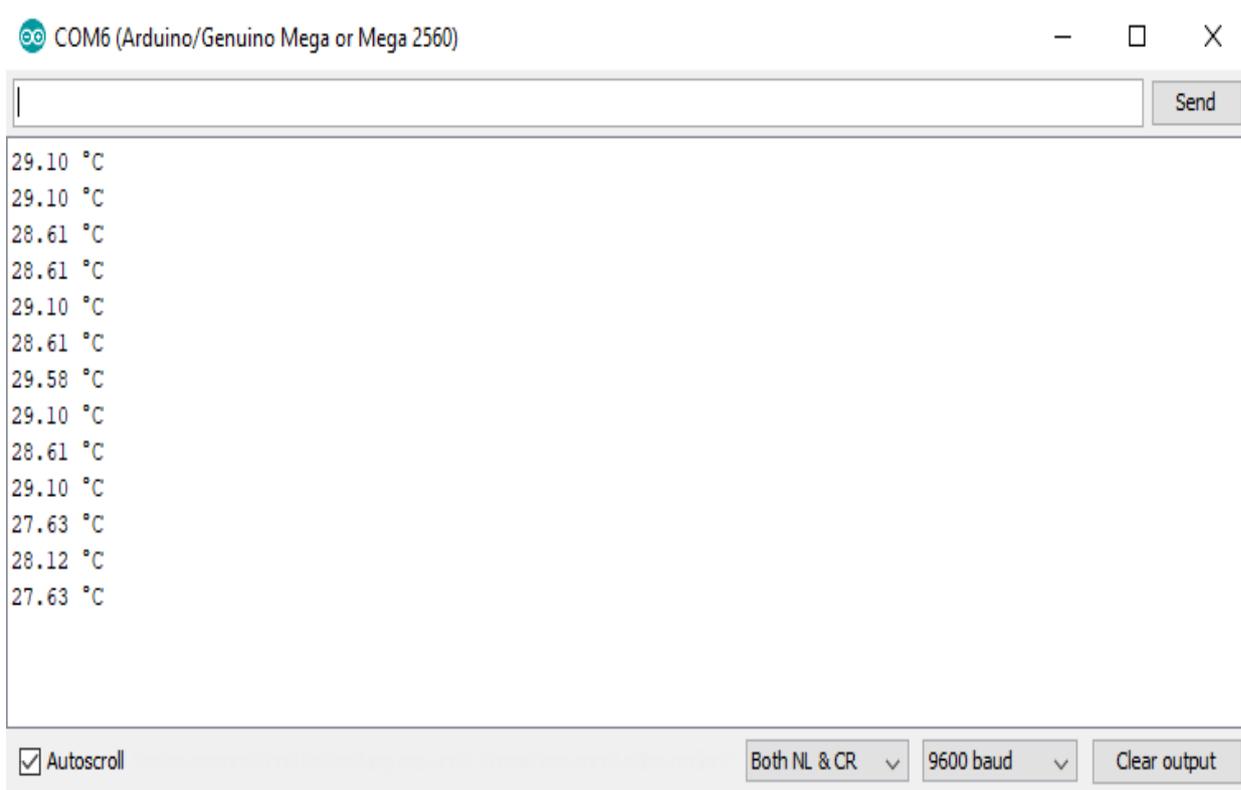
  delay(1000);
}
```

Une fois le programme est compilé et transféré vers la carte ARDUINO (via USB), nous pouvons accéder aux résultats du programme via le terminal incorporé au logiciel.

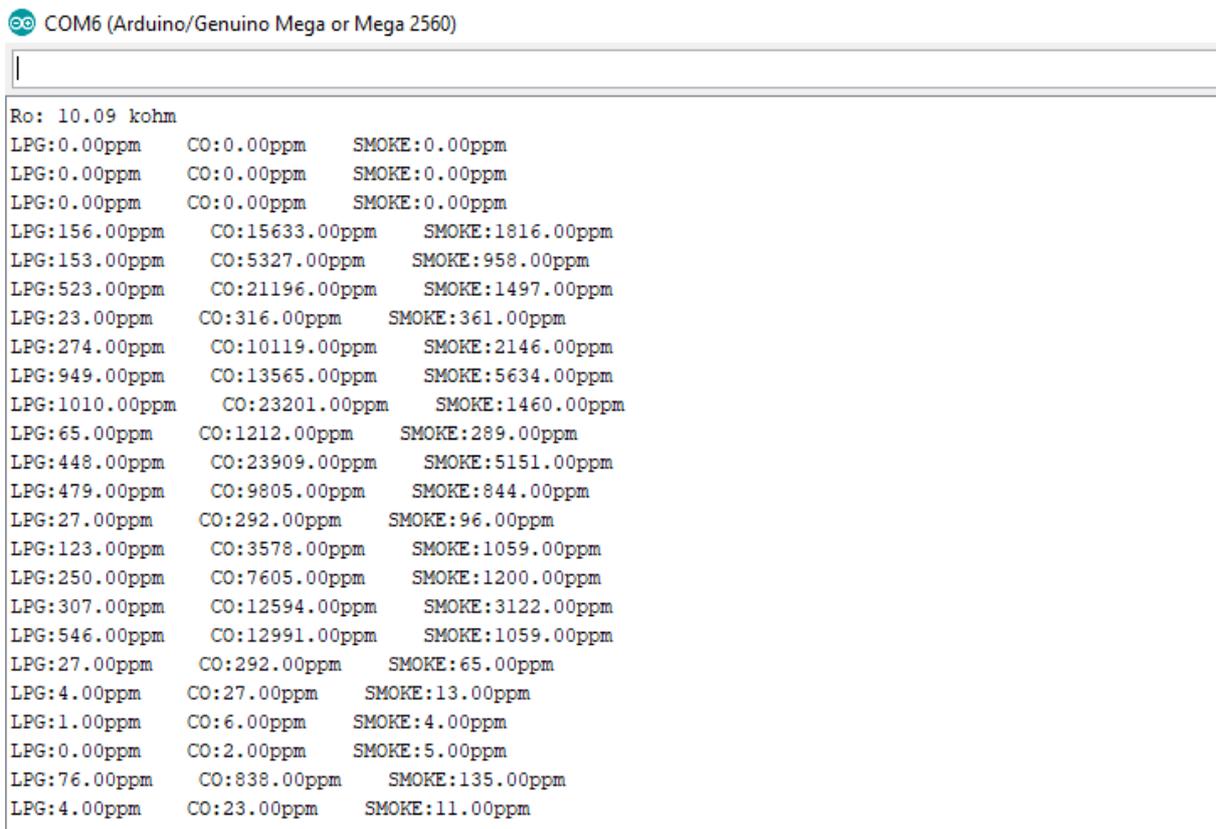
Le résultat du programme renvoie la température et la concentration de gaz, de CO et de fumée dans l'air après chaque seconde.

# CHAPITRE 4 : Réalisation Du Système de télémétrie

- Affichage des mesures de température dans le serial Monitor :



- Affichage des mesures de GLP, CO et Fumée dans le serial Monitor :



## 3. Le transfert de ces données vers une base de données

Une fois les valeurs des capteurs sont acquises, il faudra désormais les envoyer vers notre serveur pour les stocker dans une base de données MySQL, Pour cela, nous allons utiliser le Shield wifi ESP8266 ce qui va nous permette de connecter notre carte ARDUINO Méga à un réseau Wifi et établir une connexion TCP/IP en utilisant Les commandes AT.

## 4. Connexion entre l'ARDUINO méga et l'ESP8266

Pour connecter notre ESP8266 nous utiliserons une breadboard pour nous aider avec le câblage, voici un tableau qui décrit les connexions entre la carte méga et l'ESP8266 :

ARDUINO	ESP8266
RX(10)	TX
TX(11)	RX
3.3V	CH_PD
3.3V	VCC
GND	GND

Tableau IV. 1: Branchement de la carte méga et l'ESP8266

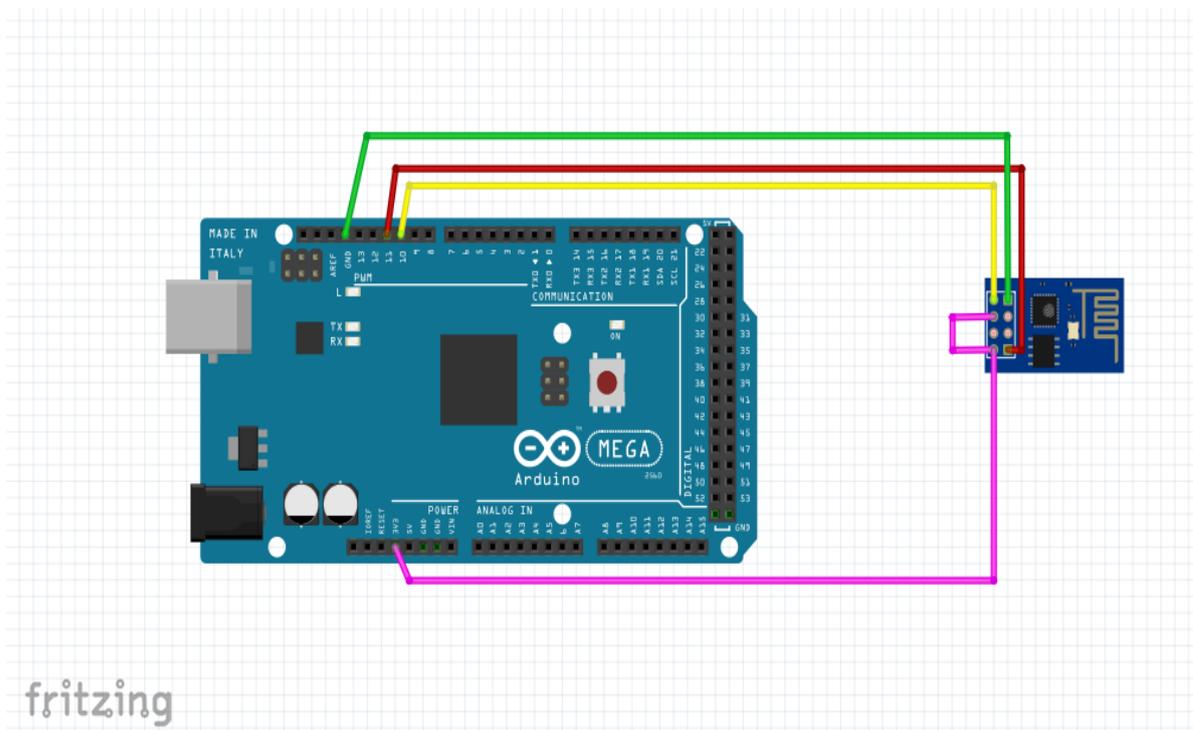


Figure IV. 7: Câblage de l'ARDUINO avec ESP8266

## 5. Câblage de la carte ARDUINO avec les deux capteurs

- Câblage du capteur de température avec ARDUINO :

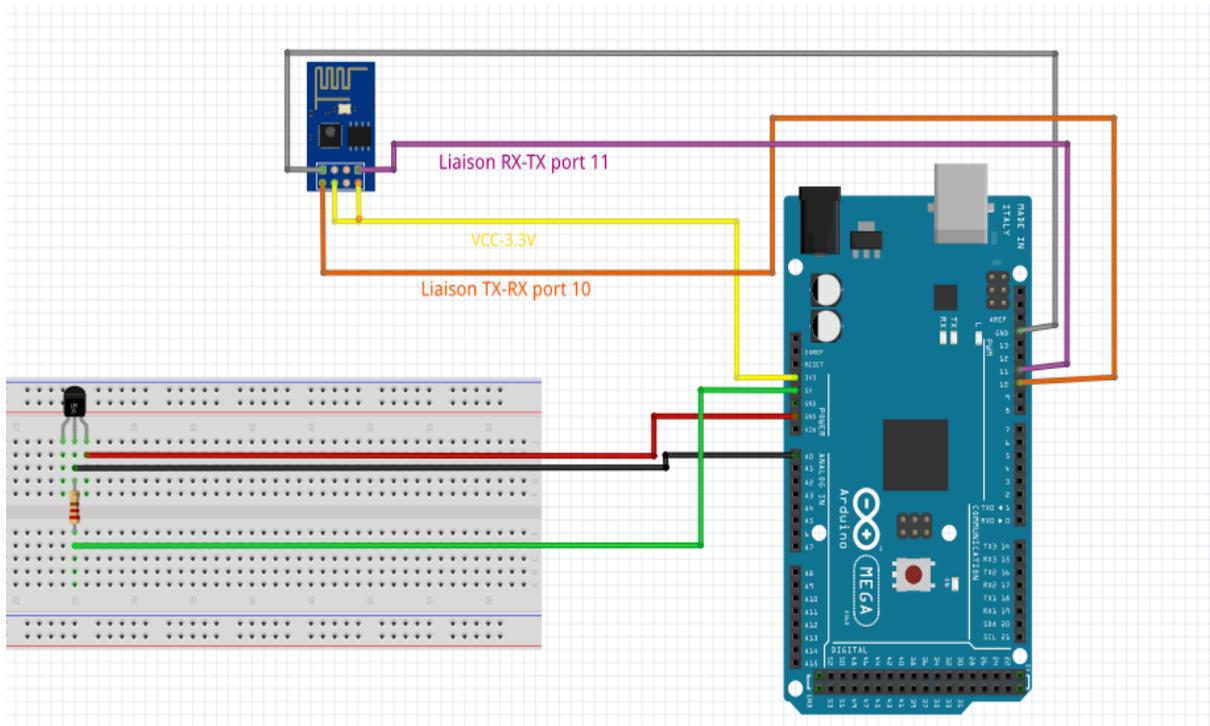


Figure IV. 8: Branchement du capteur de température

- Câblage du capteur de gaz avec ARDUINO :

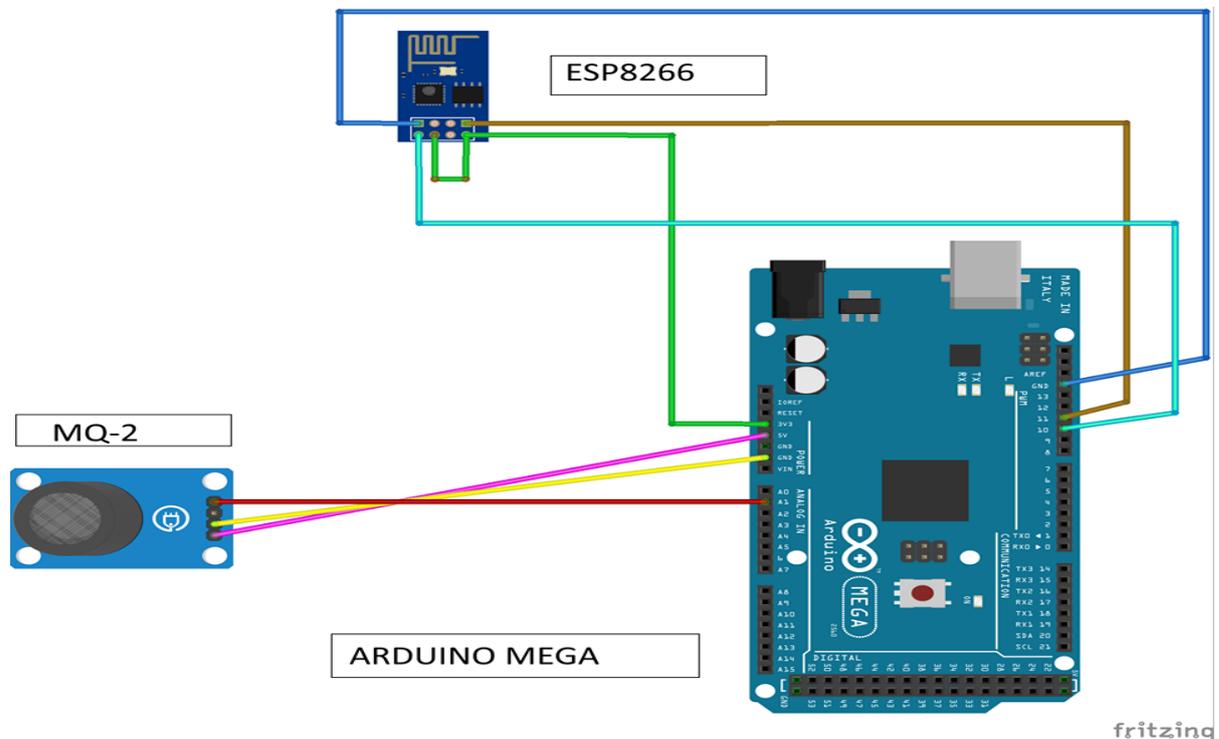


Figure IV. 9: Branchement du capteur de gaz

## 5.1 ARDUINO en tant que client web en utilisant l'ESP8266

Nous avons utilisé l'ESP8266 en tant que client, cela signifie que notre ARDUINO doit se connecter à un serveur à travers l'ESP8266 par exemple [www.google.com](http://www.google.com) ou [www.naylamp.com](http://www.naylamp.com).

Il y a plusieurs façons de communiquer entre le serveur client et le serveur web hébergé, nous communiquerons via HTTP, qui est de la même manière que les navigateurs Web (Chrome, Mozilla, etc.) communiquent avec les serveurs.

## 5.2 L'envoi des données dans une boîte de dialogue HTTP

Il y a deux façons simples d'envoyer des données ensemble avec une requête HTTP, ce sont la méthode GET et POST, La différence principale entre ces deux méthodes est que dans la méthode GET, les données sont envoyées dans l'URL, tout en étant visible.

Par contre si on utilise la méthode POST, les données sont cachées de l'utilisateur. Les deux méthodes sont largement utilisées et sont presque toujours implémentées dans les pages web que nous naviguons.

- Dans notre projet on a utilisé la méthode GET.

## 5.3 Effectuation des requêtes sur le serveur en utilisant les commandes AT

Les étapes à suivre pour initialiser le shield wifi ESP8266 à transmettre les données :

- Vérifications si l'ESP8266 répond.

```
//Nous vérifions si l'ESP8266 répond

SerialESP8266.println("AT");
if(SerialESP8266.find("OK"))
    Serial.println("Le Shild Wifi est en marche");
else
    Serial.println("Erreur avec connexion dans le shield ESP8266");
```

# CHAPITRE 4 : Réalisation Du Système de télémétrie

- Configuration du réseau :
- Mettre l'ESP8266 en mode client (connexion a un réseau existant).
- Connexion a un réseau Wifi.
- Désactivation de la connexion Multiple.

```
//-----Configuration du réseau-----//Nous pouvons commenter si l'ESP est déjà configuré

//ESP8266 en mode client (connection a un reseau existant)
SerialESP8266.println("AT+CWMODE=1");
if (SerialESP8266.find("OK"))
    Serial.println("Le shield ESP8266 est en mode Client");

//Connexion a un reseau Wifi

SerialESP8266.println("AT+CWJAP=\"ADN-4101\", \"EMB/413!!*\");
Serial.println("Tentative de connection au reseau Wifi...");
SerialESP8266.setTimeout(10000);
if (SerialESP8266.find("OK")) {
    Serial.println("Le système est conecté au résau WIFI");
}
else{
    Serial.println("Erreur lors de la connexion au réseau WIFI");
}
SerialESP8266.setTimeout(2000);
//Désactivation de la connexion Multiple
SerialESP8266.println("AT+CIPMUX=0");
if (SerialESP8266.find("OK"))
    Serial.println("Connexion Multiple désactivé");

//-----fin de configuration-----
```

- Connexion au serveur et l'envoi des données au serveur.

```
/-----l'envoi des variables au serveur -----

//Connexion au serveur:

SerialESP8266.println("AT+CIPSTART=\"TCP\", \"\" + server + "\",80");
if ( SerialESP8266.find("OK"))
{
    Serial.println();
    Serial.println();
    Serial.println();
    Serial.println("Le système est connecté au serveur Web...");

    // l'entête de la requête Http
    String requeteHTTP= "GET /GET.php?a=";
    requeteHTTP=requeteHTTP+String(temperature)+"C"+"&b="+String(State)+" HTTP/1.1\r\n";
    requeteHTTP=requeteHTTP+"Host:"+server+"\r\n\r\n";
}
```

# CHAPITRE 4 : Réalisation Du Système de télémétrie

---

Après la compilation du programme et le téléversement vers la carte ARDUINO, nous vérifions le moniteur série sur logiciel ARDUINO pour voir si tout est bien passé.

```
COM6 (Arduino/Genuino Mega or Mega 2560)

Le Shild Wifi est en marche
Tentative de connection au reseau Wifi...
Le système est conecté au résau WIFI
Connection Multiple désactivé
32.51 °C

Le système est connecté au serveur Web...
l'envoi de la requete HTTP . . .
Requête HTTP envoyée:

GET /GET2.php?a=30.56°C HTTP/1.1
Host:telemetriel3.000webhostapp.com

J'attends une réponse...
```

Figure IV. 10: Envoie d'une requête http

```
+IPD,311:HTTP/1.1 200 OK
Date: Fri, 18 May 2018 13:56:18 G
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Server: awex
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-RequestID: efb7893a5884dd70b223a727e4aa7921

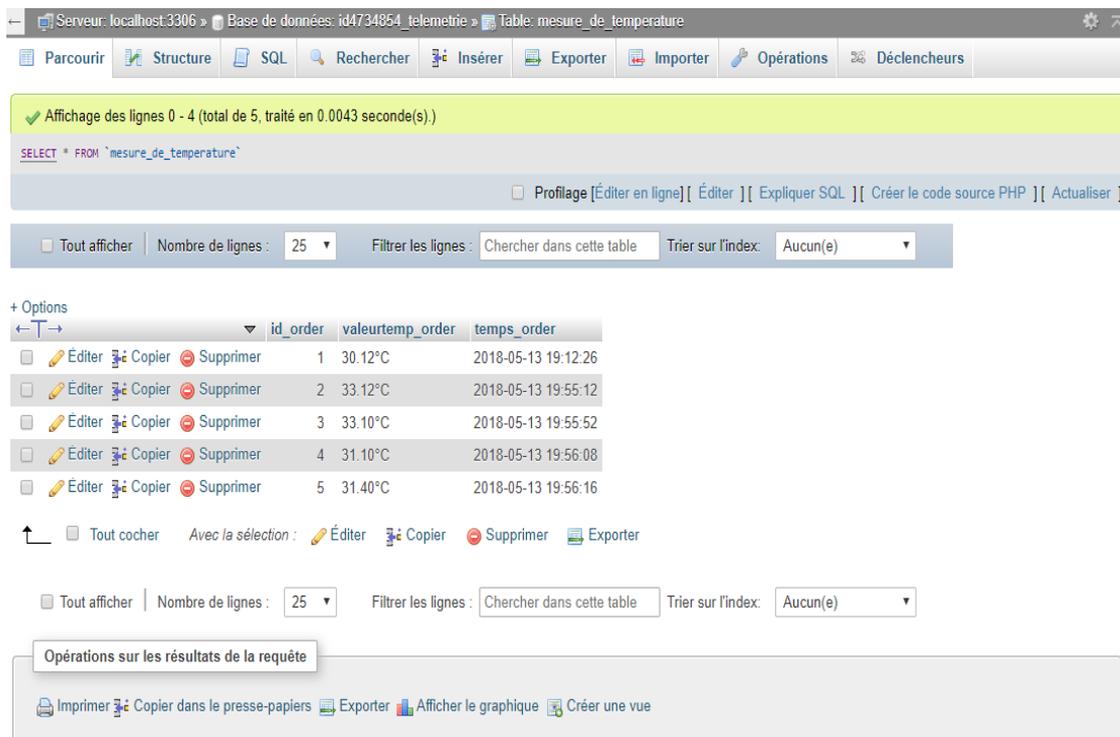
ld
 nouvelle valeur est ajoutée

OK
```

Figure IV. 11: Réponse Http « nouvelle valeur ajoutée » dans la base de données

# CHAPITRE 4 : Réalisation Du Système de télémétrie

Une fois les données sont reçues par le serveur, un script PHP implémenté dans le serveur permet de stocker directement les données dans une base de données MySQL qu'on a créé auparavant.

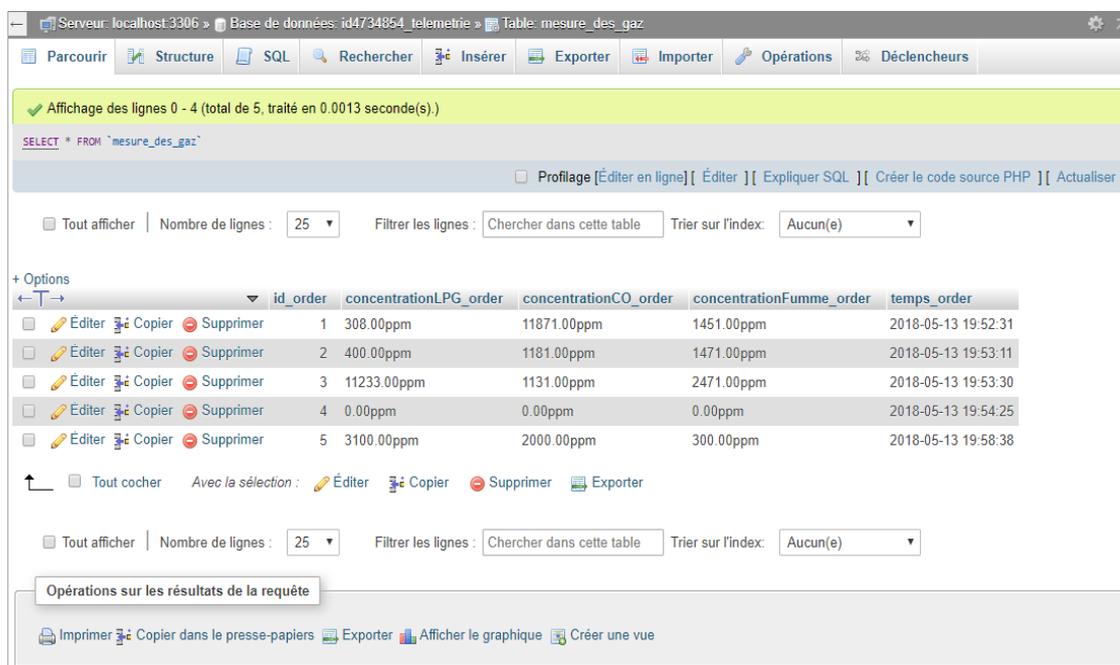


The screenshot shows a MySQL database interface with the following details:

- Server: localhost:3306
- Database: id4734854\_telemetrie
- Table: mesure\_de\_temperature
- Query: `SELECT * FROM 'mesure_de_temperature'`
- Results table with columns: `id_order`, `valeurtemp_order`, `temps_order`

	id_order	valeurtemp_order	temps_order
<input type="checkbox"/>	1	30.12°C	2018-05-13 19:12:26
<input type="checkbox"/>	2	33.12°C	2018-05-13 19:55:12
<input type="checkbox"/>	3	33.10°C	2018-05-13 19:55:52
<input type="checkbox"/>	4	31.10°C	2018-05-13 19:56:08
<input type="checkbox"/>	5	31.40°C	2018-05-13 19:56:16

Figure IV. 12: les mesures de températures après l'envoi



The screenshot shows a MySQL database interface with the following details:

- Server: localhost:3306
- Database: id4734854\_telemetrie
- Table: mesure\_des\_gaz
- Query: `SELECT * FROM 'mesure_des_gaz'`
- Results table with columns: `id_order`, `concentrationLPG_order`, `concentrationCO_order`, `concentrationFumme_order`, `temps_order`

	id_order	concentrationLPG_order	concentrationCO_order	concentrationFumme_order	temps_order
<input type="checkbox"/>	1	308.00ppm	11871.00ppm	1451.00ppm	2018-05-13 19:52:31
<input type="checkbox"/>	2	400.00ppm	1181.00ppm	1471.00ppm	2018-05-13 19:53:11
<input type="checkbox"/>	3	11233.00ppm	1131.00ppm	2471.00ppm	2018-05-13 19:53:30
<input type="checkbox"/>	4	0.00ppm	0.00ppm	0.00ppm	2018-05-13 19:54:25
<input type="checkbox"/>	5	3100.00ppm	2000.00ppm	300.00ppm	2018-05-13 19:58:38

Figure IV. 13: les mesures du gaz, fumée et CO après l'envoi

## Partie III : La Conception et le développement d'une application Android

### Android

#### 1. Introduction

Après l'envoi des mesures dans la base de données nous avons réalisé une application Android pour récupérer et afficher la dernière mesure sur le Smartphone en temps réel.

Le principal avantage de cette application c'est la récupération de ces données à distance et dans n'importe quel endroit, il suffit juste de se connecter à l'internet.

#### 2. L'environnement de travail

Pour développer sur Android il y a plusieurs environnements tels qu'Eclipse, NetBeans, WinDev et Android Studio. On a choisi Ce dernier car il est le plus utilisé et le performants.

##### **2.1 Android studio**

Android studio est un environnement de développement pour développer des applications Android annoncé le 15 mai 2013 lors du Google I/O .il est basé sur l'environnement de développement IntelliJ IDEA, de la société JetBrains.distribué gratuitement par Google il intègre plusieurs outils qui couvèrent l'essentiel des besoins du développeur, Android Studio a pour fonctions d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose également des outils pour gérer le développement Studio d'applications :

- Le moteur de production (Build system en anglais) Gradle, qui permet de produire plusieurs versions d'une même application.
- Un outil pour construire les interfaces utilisateurs visuellement.
- Un ensemble de modèles d'applications intégrant les principales structures de code utilisés dans le développement Android. [4]

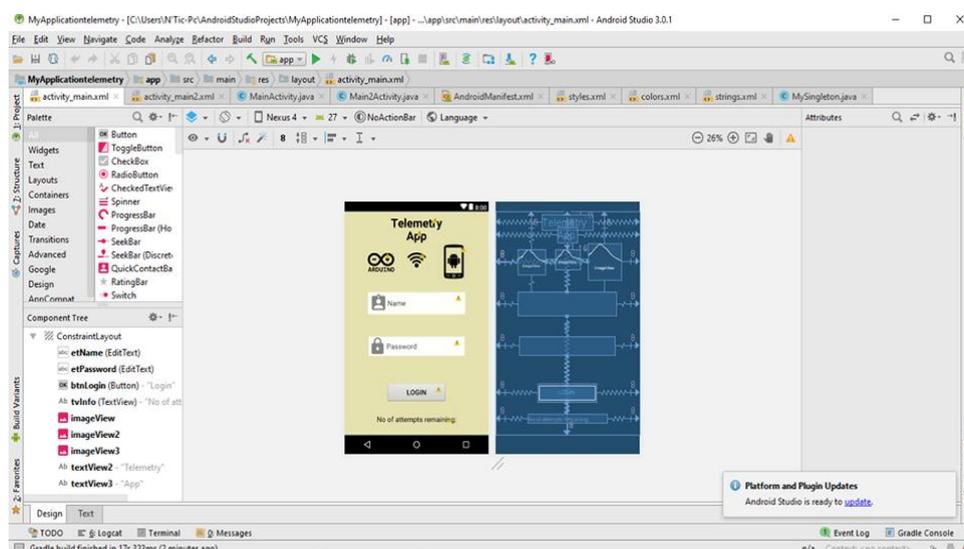


Figure IV. 14: Vue sur l'interface de logiciel Android Studio

## 2.2 La Communication entre l'application Android et la base de données MySQL

La communication entre l'application et la base de données se fait par une requête HTTP, l'application Android envoie une requête vers le serveur pour récupérer la dernière mesure sous forme JSON, Cette requête est la même que celle que nous faisons à partir d'un navigateur web, il suffit de saisir une URL et nous attendons à recevoir une réponse.

Une fois la requête est arrivée à destination, un script PHP implémenté dans le serveur se charge de faire le dialogue avec la base de données, il récupère la dernière valeur stocker dans la base de données et le converti en un format de données JSON. Il existe plusieurs méthodes pour établir les requêtes HTTP en Android, alors on a utilisé la bibliothèque Volley qui est très récente et beaucoup utilisé en ce moment.

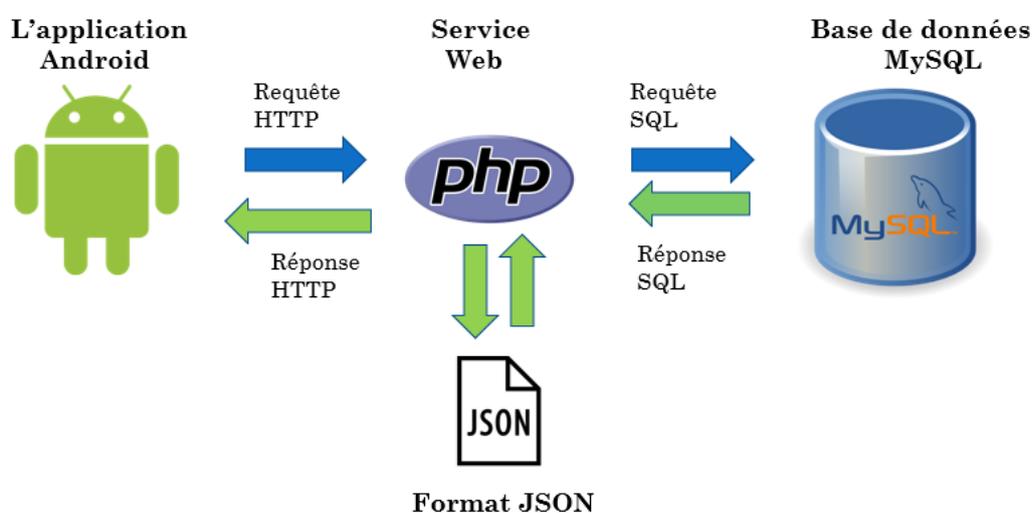
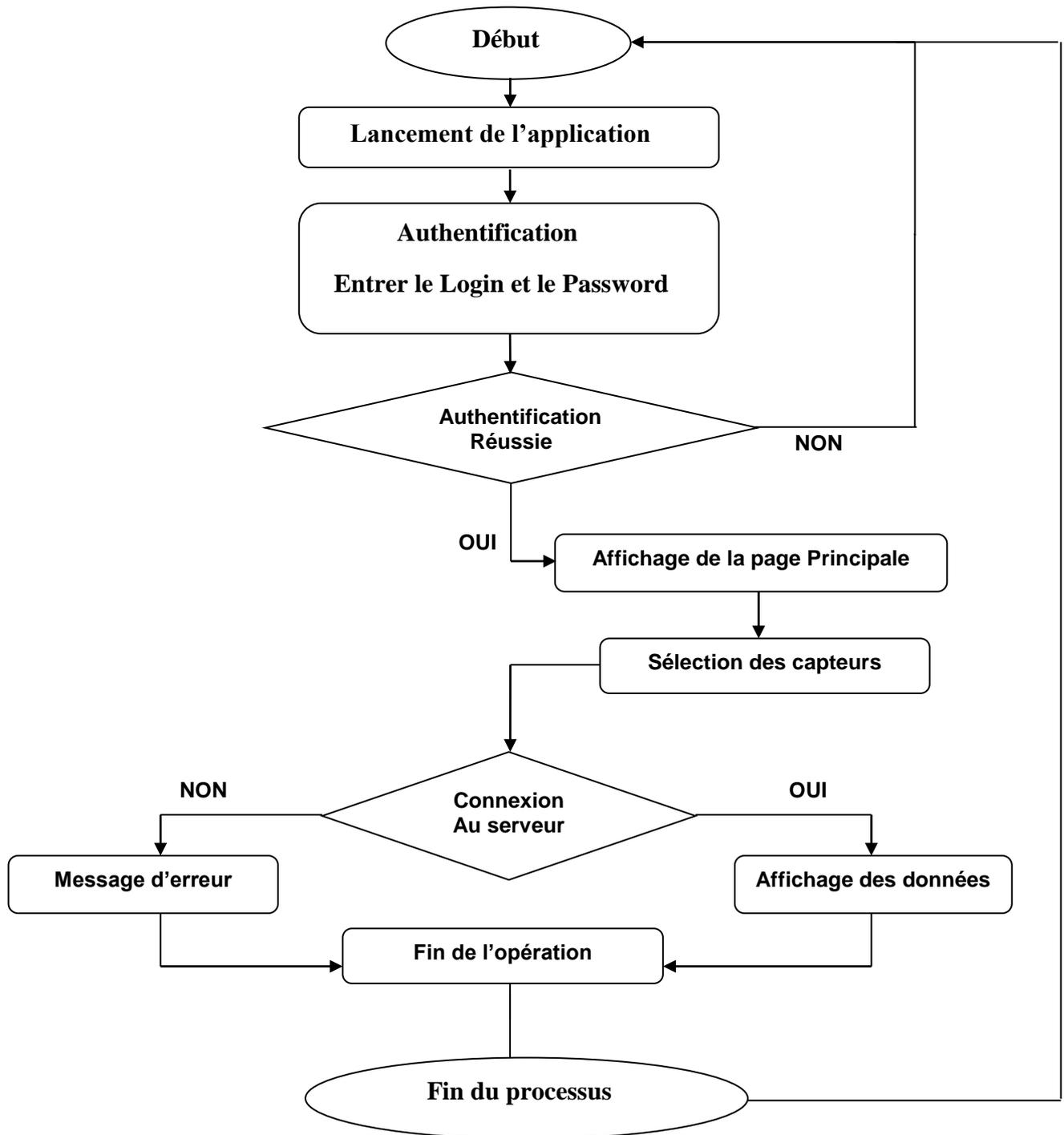


Figure IV. 15: Echange de requêtes entre l'application et la base de données

# CHAPITRE 4 : Réalisation Du Système de télémétrie

## 2.3 Organigramme d'activité de l'application Android



## 2.4 Présentation des interfaces de l'application

### 2.4.1 La Page d'authentification

L'application débute par le lancement de la fenêtre d'authentification qui consiste à la sécurisation de notre application pour les accès non autorisés, une fois l'identifiant et le mot de passe vérifiés, l'utilisateur peut alors accéder à toutes les fonctionnalités de l'application.

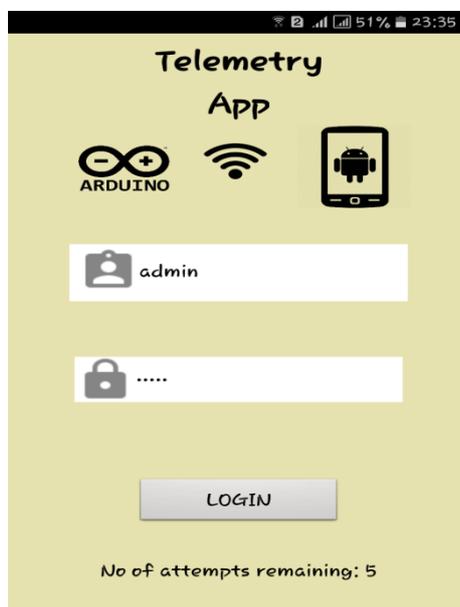


Figure IV. 16: Page d'authentification

### 2.4.2 La Page Principale

Une fois l'établissement de connexion est fait, l'utilisateur va être dirigé vers l'interface principale sur laquelle il peut choisir les capteurs disponibles dans notre système de télémétrie et consulter leurs valeurs avec temps réel.

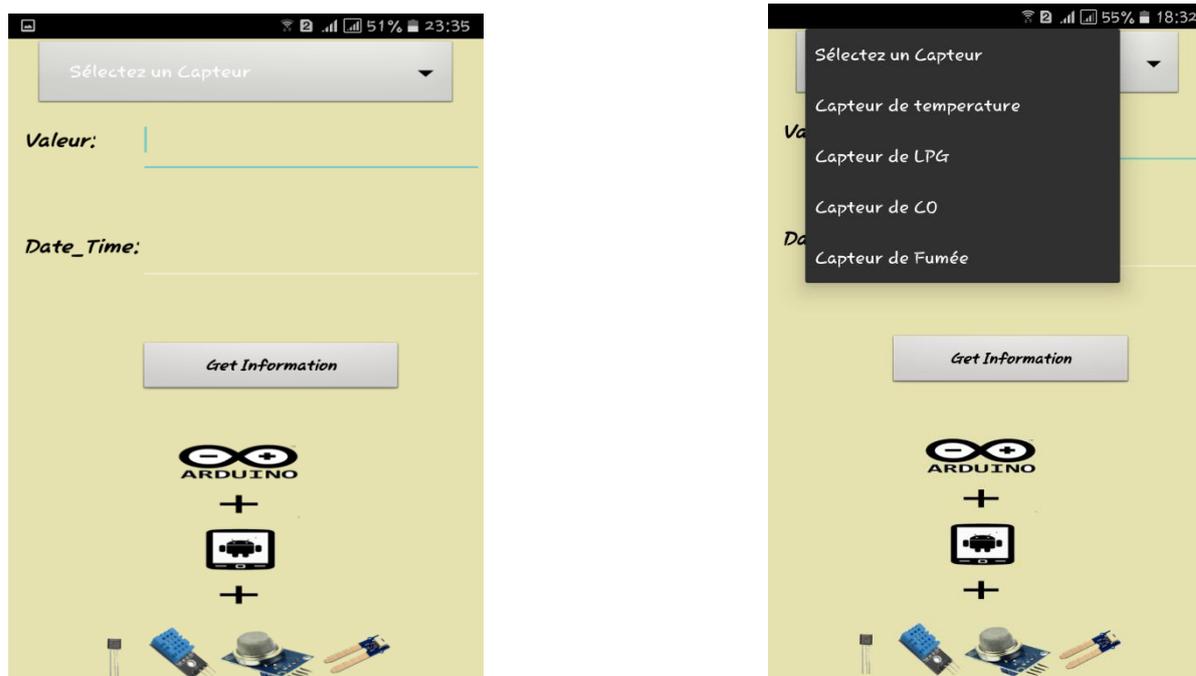


Figure IV. 17: Vue sur l'interface principale

## 2.5 Récupération des données

Après la sélection du capteur désiré, nous cliquons sur le bouton < Get Information> et cela nous donne la dernière valeur envoyée par l'ARDUINO avec la date et l'heure de la mesure.

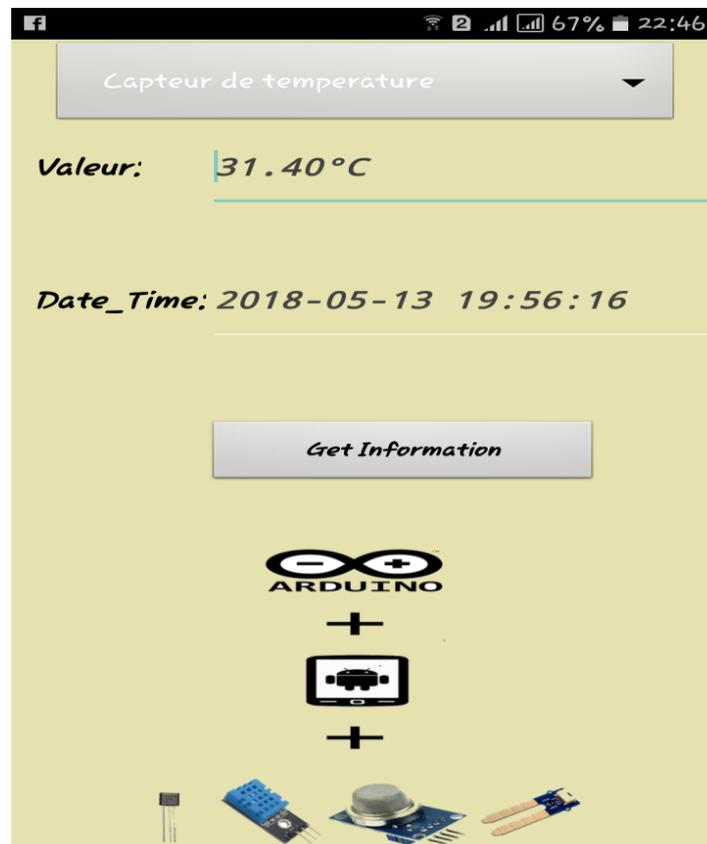


Figure IV. 18: Visualisation de la mesure reçue

## **Partie IV : Conception et réalisation d'un site web**

### **1. Introduction et objectif**

Après la réalisation de l'application Android nous avons choisit d'ajouter un autre moyen de visualisation de toutes nos données et cela sera la conception d'un site web.

La conception de ce site web a été faite en utilisant plusieurs langages web :

- Le HTML.
- Le CSS.
- Le JavaScript.
- L'Ajax.
- Le JQuery.
- Le PHP.
- Le SQL.

Ce site va contenir plusieurs pages de consultation de données et dans chaque page on installe un filtre pour l'objectif de limiter la recherche de nos données sur des périodes précise.

### **2. Programmation web**

Nous avons dans un premier temps mis en place la structure du site en créant la page d'identification et la page d'accueil avec les cinq pages du menu et leurs designs. Cette première partie est principalement axée sur le côté HTML, CSS et Bootstrap.

Ensuite, dans un second temps, nous avons attaqués la programmation en PHP et SQL pour assurer la sécurité de notre site web et leur page.

Puis, nous avons utilisé le JavaScript et la bibliothèque jQuery et la méthode AJAX avec PHP et SQL pour communiquer avec la base de données et afficher toutes les mesures dans des tableaux classés par ordre décroissant de temps en ajoutant la fonctionnalité de filtrer par période de date et heure.

### **3. Architecture du site**

#### **3.1 Page d'authentification**

Dans cette page l'utilisateur doit saisir son login et son mot de passe afin d'accéder au site.



Figure IV. 19: Page d'authentification

## 3.2 Page d'accueil de site web

La page d'accueil elle s'affiche directement après l'authentification, permet d'ouvrir toutes les autres pages, elle contient un menu de navigation déroulant constituée de quatre parties principales :

- Accueil : nous dirige vers notre page d'accueil.
- Afficher : pour visualiser toutes les mesures en ordre décroissant de temps.
- Rechercher : en cliquant sur ce bouton il nous affiche un sous menu horizontal pour rechercher les mesures par date et période de date et période d'heure.
- Contact : affiche la page de nos contacts en tant que créateur de ce site web.
- Plus : nous donner la main pour savoir d'autre différentes fonctionnalités de ce site (se déconnecter, à propos..)

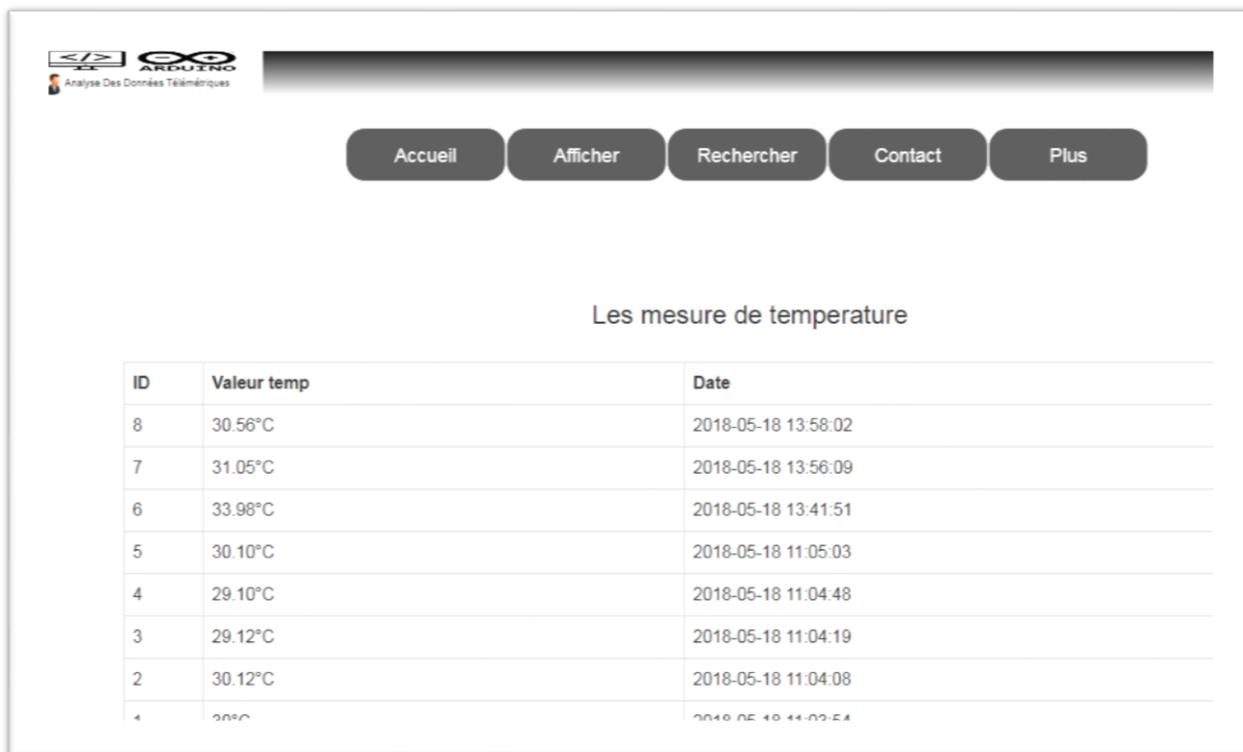


Figure IV. 20: Page d'accueil

## 3.3 Page d'affichage de toutes les mesures

Pour afficher toutes les données mesurées par nos capteurs il suffit de d'aller cliquer sur la barre du menu sur l'icône Afficher toutes les mesures.

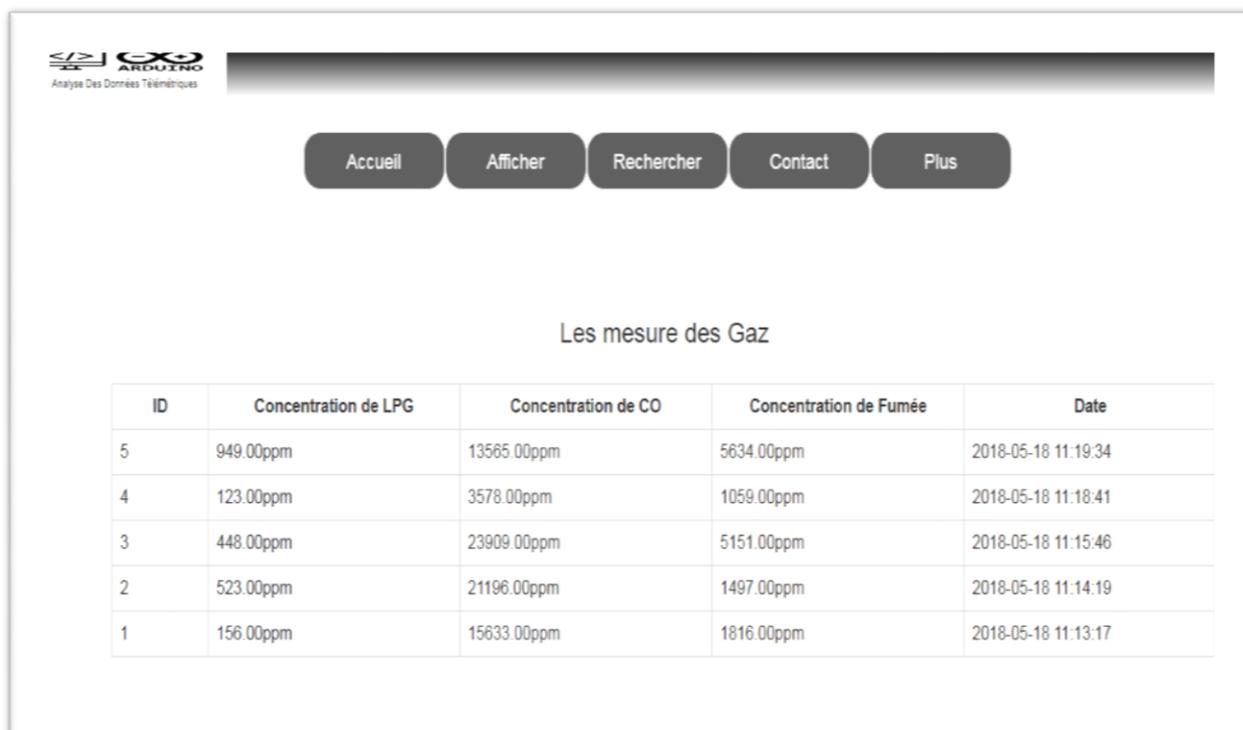
➤ Capteur de température :



ID	Valeur temp	Date
8	30.56°C	2018-05-18 13:58:02
7	31.05°C	2018-05-18 13:56:09
6	33.98°C	2018-05-18 13:41:51
5	30.10°C	2018-05-18 11:05:03
4	29.10°C	2018-05-18 11:04:48
3	29.12°C	2018-05-18 11:04:19
2	30.12°C	2018-05-18 11:04:08
1	29.12°C	2018-05-18 11:03:54

Figure IV. 21: page d'affichage de toutes les mesures de température

➤ Capteur de Gaz :



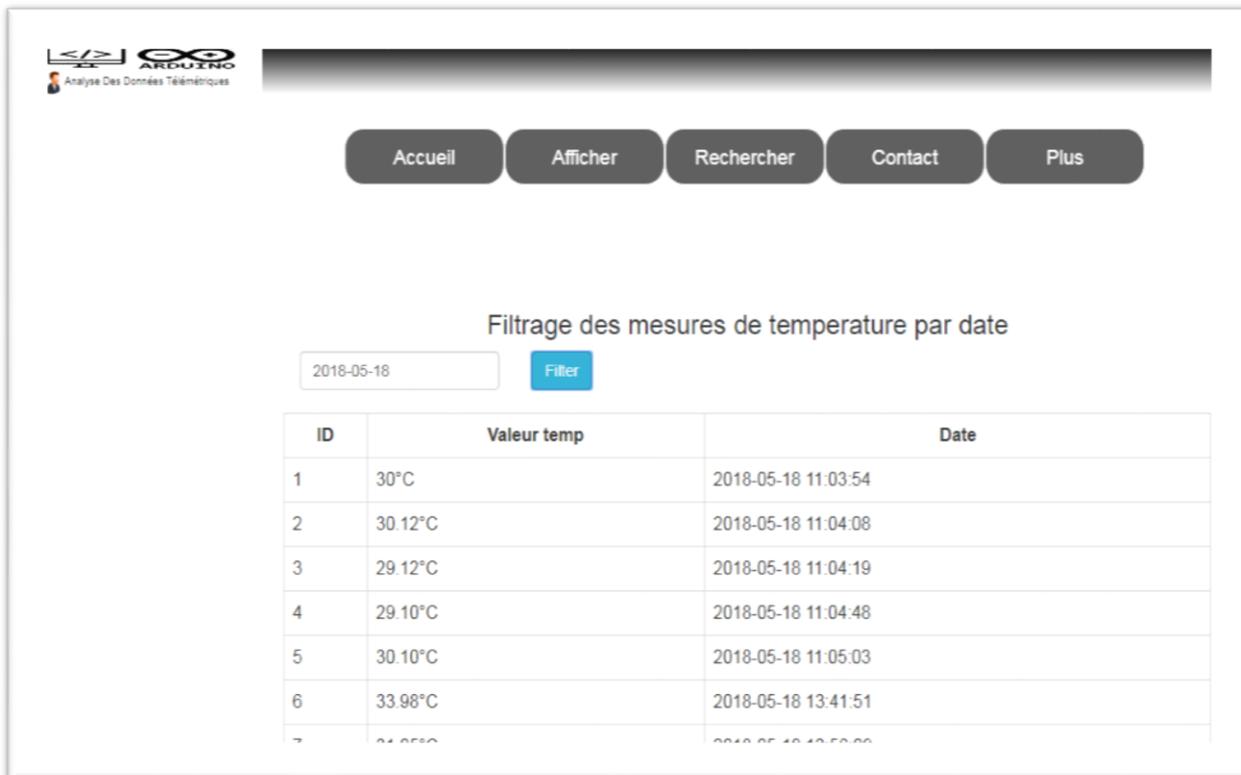
ID	Concentration de LPG	Concentration de CO	Concentration de Fumée	Date
5	949.00ppm	13565.00ppm	5634.00ppm	2018-05-18 11:19:34
4	123.00ppm	3578.00ppm	1059.00ppm	2018-05-18 11:18:41
3	448.00ppm	23909.00ppm	5151.00ppm	2018-05-18 11:15:46
2	523.00ppm	21196.00ppm	1497.00ppm	2018-05-18 11:14:19
1	156.00ppm	15633.00ppm	1816.00ppm	2018-05-18 11:13:17

Figure IV. 22: page d'affichage de toutes les mesures de gaz

## 3.4 Page de recherche par date

Cette page permet de filtrer la recherche. L'utilisateur doit choisir une date, et en revanche il obtient toutes les données mesurées pendant ce jour.

➤ Capteur de température :



ID	Valeur temp	Date
1	30°C	2018-05-18 11:03:54
2	30.12°C	2018-05-18 11:04:08
3	29.12°C	2018-05-18 11:04:19
4	29.10°C	2018-05-18 11:04:48
5	30.10°C	2018-05-18 11:05:03
6	33.98°C	2018-05-18 13:41:51
7	34.05°C	2018-05-18 13:50:00

Figure IV. 23: filtrage de données de température par date

➤ Capteur de Gaz :



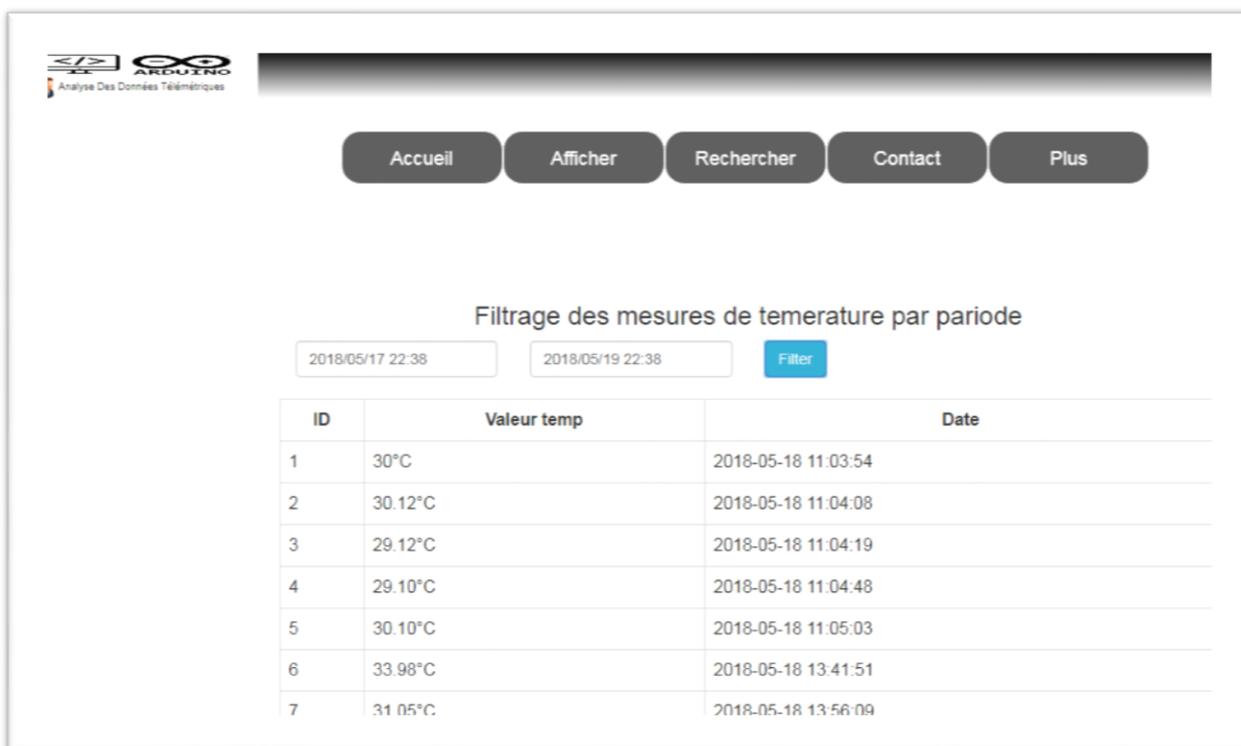
ID	Concentration de LPG	Concentration de CO	Concentration de Fumée	Date
1	156.00ppm	15633.00ppm	1816.00ppm	2018-05-18 11:13:17
2	523.00ppm	21196.00ppm	1497.00ppm	2018-05-18 11:14:19
3	448.00ppm	23909.00ppm	5151.00ppm	2018-05-18 11:15:46
4	123.00ppm	3578.00ppm	1059.00ppm	2018-05-18 11:18:41
5	949.00ppm	13565.00ppm	5634.00ppm	2018-05-18 11:19:34

Figure IV. 24: filtrage de données de gaz par date

## 3.5 Page de recherche une période par date

Ce filtre permet à l'utilisateur de limiter de plus en plus la recherche en donnant seulement les mesures d'une période par jour (un domaine de temps par jour).

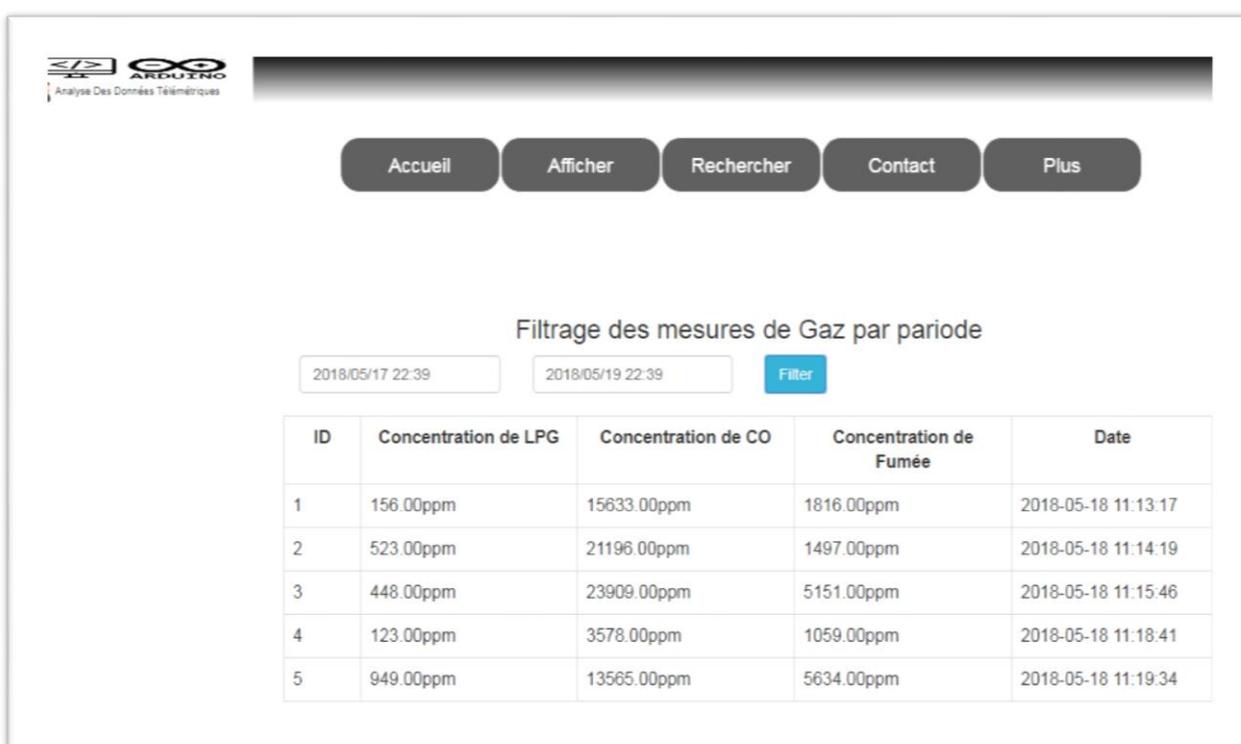
- Capteur de température :



ID	Valeur temp	Date
1	30°C	2018-05-18 11:03:54
2	30.12°C	2018-05-18 11:04:08
3	29.12°C	2018-05-18 11:04:19
4	29.10°C	2018-05-18 11:04:48
5	30.10°C	2018-05-18 11:05:03
6	33.98°C	2018-05-18 13:41:51
7	31.05°C	2018-05-18 13:56:09

Figure IV. 25: filtrage de données de température (période par jour)

- Capteur de Gaz :



ID	Concentration de LPG	Concentration de CO	Concentration de Fumée	Date
1	156.00ppm	15633.00ppm	1816.00ppm	2018-05-18 11:13:17
2	523.00ppm	21196.00ppm	1497.00ppm	2018-05-18 11:14:19
3	448.00ppm	23909.00ppm	5151.00ppm	2018-05-18 11:15:46
4	123.00ppm	3578.00ppm	1059.00ppm	2018-05-18 11:18:41
5	949.00ppm	13565.00ppm	5634.00ppm	2018-05-18 11:19:34

Figure IV. 26: filtrage de données de gaz (période par jour)

## 3.6 Page de recherche une période par heure

Le dernier filtre donne les mesures sur un intervalle de temps en minutes.

- Capteur de température :

ID	Valeur temp	Date
1	30°C	2018-05-18 11:03:54
2	30.12°C	2018-05-18 11:04:08
3	29.12°C	2018-05-18 11:04:19
4	29.10°C	2018-05-18 11:04:48
5	30.10°C	2018-05-18 11:05:03

Figure IV. 27: filtrage de données de température (période d'heure par jour)

- Capteur de Gaz :

ID	Concentration de LPG	Concentration de CO	Concentration de Fumée	Date
1	156.00ppm	15633.00ppm	1816.00ppm	2018-05-18 11:13:17
2	523.00ppm	21196.00ppm	1497.00ppm	2018-05-18 11:14:19
3	448.00ppm	23909.00ppm	5151.00ppm	2018-05-18 11:15:46

Figure IV. 28: filtrage de données de gaz (période d'heure par jour)

## 4. Conclusion

Dans ce dernier chapitre nous avons présenté tout le travail que nous avons accompli en spécifiant toutes les étapes de notre projet sur 4 parties. Nous avons commencé par l'hébergement du serveur web et la création de la base de données plus les scripts que nous avons utilisés pour l'échange des requêtes. Ensuite, l'utilisation du microcontrôleur programmable ARDUINO jusqu'à la réalisation de l'application Android et la création du siteweb.

# Conclusion générale

---

## Conclusion et perspectives

Dans ce travail nous avons réussi à réaliser un système de télémétrie. Ce système est constitué de plusieurs dispositifs et chaque dispositif aura un rôle important dans notre système, l'objectif de notre projet consiste à transmettre des informations qui sont sous la forme de valeurs d'un paramètre physique ou chimique tel que la concentration du CO, la température, l'humidité, Et qui est mesuré par des capteurs et lu par un microcontrôleur ARDUINO. La transmission de ces informations se fait sur la base d'un module Wifi et la récupération et la visualisation de ces informations sera faite au niveau d'**Une application Android et Un site Web.**

Afin de réaliser un tel système nous avons divisé notre travail sur 4 parties principales, Dans la 1<sup>er</sup> partie nous nous sommes occupés de l'hébergement du serveur web où nous avons intégré dedans notre base de donnée et les scripts PHP pour le traitement des requêtes http et le stockage des infos dans cette base de donnée, ensuite nous avons programmé le dispositif ARDUINO et le module Wifi de façon à nous permettre de lire des donnée dans un premier lieu et les envoyer vers notre base de données.

En ce qui concerne les deux dernières parties et comme cité précédemment nous avons choisi de concevoir deux moyens pour visualiser nos mesures :

**Une application Android** pour la récupération des dernières mesures envoyées par l'ARDUINO vers la base de données en temps réel et **Un site web** pour l'affichage de toutes les mesures stockées dans la base de données. De plus, Ce site contient des filtres pour limiter notre recherche selon la période désirée par l'utilisateur.

Nous notons que ce travaille nous a donné l'opportunité de découvrir des nouvelles connaissances telle que la programmation dans logiciel ARDUINO ide ainsi que la découverte du langage JAVA et d'autre part, approfondir nos connaissances concernant les technologies web tel que l'Ajax, le JQuery, le PHP, le SQL.

**Perspectives du projet,** Nous espérons développer notre système de façon à ajouter plus de fonctionnalité à ce système actuel tel que :

- Le contrôle de la carte ARDUINO via l'application Android et faire la lecture et l'envoi des données à la demande de l'utilisateur.
- De plus, mettre le système sur un drone et cela nous donne la possibilité de relevé les valeurs (données) dans des endroits souvent inaccessibles.
- Ajouter la fonctionnalité de localisation en mettant un module GPS dans le système ce qui permet de connaitre l'endroit exact où les valeurs ont été prises.
- L'implémentation de notre application androïde pour d'autre système d'exploitation tel que IOS, Windows Phone, ...etc.
- Notre projet peut également servir d'autres projets universitaires plus complexes comme étant une plateforme de base.

# Liste des abréviations

---

## Liste des abréviations

UAL : unité arithmétique et logique

ROM: Read-Only Memory (memoire non volatile)

OTPROM: One time programming read only memory (programmable une seule fois)

UVPROM: ultra violet programming read only memory

EEPROM: electrally erasable programming read only memory (memoire effaceable électriquement)

UART : Universal Asynchronous Receiver Transmitter (émetteur-récepteur asynchrone universel)

ICSP : In-Circuit Serial Programming(programmation en circuit )

FTDI : Future Technology Devices International

TX : voie de transsmision

RX : voie de réception

TTL : transistor logic (transistor logique)

PWM : pulse width modulation (modulation par largeur d'impulsion)

SPI : Interface Série Périphérique (interface de périphériques séries)

AREF : analogique reference

TWI : Two Wire Interface (interface à 2 fils)

PIC : Peripheral Interface Controller (contrôleur d'interface périphérique)

FSR: Force Resistor Sensing (capteur de force)

HTML: HyperText Markup Language

CSS: Cascading Style Sheets

PHP: HyperText Preprocessor

Http: HyperText Transfer Protocol

IIS : Internet Information Service

JSON : JavaScript Object Notation

XML : eXtensible Markup Language

AJAX : Asynchronous JavaScript and XML

IP : internet Protocol

TCP : transmission control Protocol)

OSI : open systems interconnection (system d'interconnexion)

UDP : user datagramme Protocol (protocole de datagramme utilisateur)

URL Uniform Resource Locator (localisateur uniforme de ressource)

DB : DataBase (base de données)

SGBD : système de gestion de bases de données

# Liste des abréviations

---

SQL: STRUCTURED QUERY LANGUAGE

OS: Operating system (system d'exploitation)

OHA : l'Open Hand Set Alliance

SDK : Software développement kit (kit de développement matériel)

NDK: native developpement kit

REST: Representational State Transfer

SOAP: Service Oriented Architecture

VBNET: visial basic NET

WIFI : Wireless Fidelity

## Bibliographies

[1]. Mémoire de fin d'études KRAMA Abdelbasset, GOUGUI Abdelmoumen, **Etude et réalisation d'une carte de contrôle par ARDUINO via le système Android**, Master en électronique industriel. Année universitaire 2014/2015.

[2]. Nazim BENBOURAHLA, Septembre 2012, Android les fondamentaux du développement d'application Java, Edition ENI.

[3]. Christophe GILBERT, février 2017, Conception d'interface pour mobiles- graphisme et développement Des applications natives, web et hybrides, Edition ENI.

[4]. Sylvain HÉBUTERNE, Aout 2016, Android Guide de développement d'applications Java pour Smartphones et tablettes, Edition ENI.

## Webographies

[1<sup>°</sup>] <http://www.techno-science.net/?onglet=glossaire&definition=6737>

Consulté le 02/04/2018

[2<sup>°</sup>] <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/robotique/chapitres/MicrocontroleurWhat.html>

Consulté le 02/04/2018

[3<sup>°</sup>] [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.MaterielMega2560](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560)

Consulté le 06/03/2018

[4<sup>°</sup>] [http://www.mon-club-elec.fr/pmwiki\\_reference\\_arduino/pmwiki.php?n=Main.DebuterPresentationLogiciel](http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DebuterPresentationLogiciel)

Consulté le 06/03/2018

[5<sup>°</sup>] <https://www.ekito.fr/people/presentation-de-l-esp8266>

Consulté le 06/023/2018

[6<sup>°</sup>] <http://www.electro-info.ovh/index.php?id=220>

Consulté le 26/03/2018

[7<sup>°</sup>] <http://domotique.caron.ws/cartes-microcontroleurs/les-commandes-at>

Consulté le 15/03/2018

[8<sup>°</sup>] [https://developer.mozilla.org/fr/Apprendre/Qu\\_est-ce\\_qu\\_un\\_serveur\\_web](https://developer.mozilla.org/fr/Apprendre/Qu_est-ce_qu_un_serveur_web)

Consulté le 20/03/2018

[9<sup>°</sup>] <http://www.amoks.com/rep-lexique/ido-192/ajax.html>

Consulté le 05/04/2018

[10<sup>°</sup>] [http://www.academia.edu/21537628/Rsx\\_OSI\\_TCPIP\\_cours](http://www.academia.edu/21537628/Rsx_OSI_TCPIP_cours)

Consulté le 16/04/2018

[11<sup>°</sup>] <https://openclassrooms.com/courses/les-requetes-http>

Consulté le 20/04/2018

# Bibliographies et Webographies

---

[12<sup>°</sup>] [https://edutechwiki.unige.ch/fr/Le\\_fonctionnement\\_du\\_protocole\\_HTTP](https://edutechwiki.unige.ch/fr/Le_fonctionnement_du_protocole_HTTP)

Consulté le 20/04/2018

[13<sup>°</sup>] <http://www.commentcamarche.com/contents/104-bases-de-donnees-introduction>

Consulté le 15/04/2018

[14<sup>°</sup>] <https://www.numerama.com/tech/132165-les-versions-dandroid-les-plus-utilisees.html>

Consulté le 25/03/2018

[15<sup>°</sup>] <http://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html>

Consulté le 25/03/2018

[16<sup>°</sup>] [https://fr.wikipedia.org/wiki/Android\\_Lollipop](https://fr.wikipedia.org/wiki/Android_Lollipop)

Consulté le 25/03/2018

[17<sup>°</sup>] [https://fr.wikipedia.org/wiki/Android\\_Marshmallow](https://fr.wikipedia.org/wiki/Android_Marshmallow)

Consulté le 26/03/2018

[18<sup>°</sup>] <https://fr.yeeply.com/blog/langages-de-programmation-creer-une-application-android/>

Consulté le 26/03/2018

[19<sup>°</sup>] <https://developer.android.com/reference/java/net/URLConnection>

Consulté le 06/04/2018

[20<sup>°</sup>] <http://tutos-android-france.com/introduction-a-okhttp/>

Consulté le 06/04/2018

[21<sup>°</sup>] <http://blog.webwag.com/2017/02/14/introduction-a-volley-gson/>

Consulté le 06/04/2018

## المخلص

في هذا العمل المتواضع تمكنا من إنجاز نموذج إلكتروني مصغر باستعمال منصة أردوينو يتمثل في قراءة بيانات معينة عن بعد (قياسات فزيائية أو كيميائية) وإرسالها عبر شبكة الأنترنت الى قاعدة بيانات خاصة بالمشروع لتخزينها، هذا بالإضافة إلى تصميم تطبيق خاص بمنصة الهواتف الذكية أندرويد إلى جانب إنشاء موقع على شبكة الأنترنت والهدف منهما هو تفحص وعرض البيانات المرسله والمخزنة داخل قاعدة البيانات.

## الكلمات المفتاحية :

القياس عن بعد، أردوينو، اندرويد، موقع ويب، خادم ويب، قاعدة بيانات.

## Résumé

Dans ce modeste travail, nous avons pu réaliser un système de télémétrie qui procède à la lecture de certaines données (paramètres physiques ou chimiques) et les envoyer via le réseau Internet vers une base de données afin de stocker ces informations, en plus on a créé une application pour la plateforme Android et créé également un site web pour l'objectif de récupérer et afficher les données envoyées et Stockées dans la base de données.

## Mots Clés

Télémétrie, Télémésure, ARDUINO, ANDROID, Site web, Serveur web, Base de données.

## Abstract

In this modest work, we developed a telemetry system that reads some data and send them on the Internet to a database in order to store these informations, in addition we have created an application for the Android platform and also created a website for the purpose of retrieving and displaying the data sent from the system and stored in the database.

## Key Words:

Telemetry, ARDUINO, ANDROID, Web Site, Web Server, Data base.