

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

**Par** : NOUALI IBRAHIM YASSINE

### **Sujet**

**Conception et réalisation d'un prototype IoT pour la surveillance à distance d'un lieu via une application Web**

Soutenu publiquement, le 24/06/2018, devant le jury composé de :

Mr. S.M.H. Irid	MCB	Univ. Tlemcen	Président
Mr. M. Hadjila	MCB	Univ. Tlemcen	Encadrant
Mr. F. Derraz	MCB	Univ. Tlemcen	Encadrant
Mr. R. Merzougui	MCA	Univ. Tlemcen	Examineur

# REMERCIEMENTS

Je tiens tout d'abord à remercier ALLAH le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, je tiens à remercier sincèrement Messieurs, M.Hadjila et F.Derraz, qui, en tant que Directeurs de mémoire, se sont toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu me consacrer et sans leurs contributions ce mémoire n'aurait jamais vu le jour.

J'adresse mes sincères remerciements à mes parents, ainsi que mon frère d'avoir toujours été là et de m'avoir tant aidé et encouragé. Si je suis ici aujourd'hui, c'est grâce à eux.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions. Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis qui m'ont toujours encouragé au cours de la réalisation de ce mémoire.

Merci à toutes et à tous.

*À ma famille et à mes amis.*

## ملخص

حالياً، ظهرت تكنولوجيا جديدة في مجال الإلكترونيات وشبكات الكمبيوتر التي تمكننا من تطوير خدمات وتطبيقات جديدة تربط العالم المادي بالعالم الافتراضي. هذه التكنولوجيا هي إنترنت الأشياء (IoT). إنها تسمح برسم خريطة رقمية للعالم الحقيقي، وإعطاء هوية إلكترونية لأشياء وأماكن البيئة المادية.

في هذه المذكرة، تم تطبيق مفهوم إنترنت الأشياء وتم توسيع نطاق استخدامه بإنجاز نموذج قائم على هذه التكنولوجيا والذي يمنح المستخدم القدرة على المراقبة عن بعد من خلال تطبيق ويب.

يتكون النموذج المقترح من ثلاث عقد مسؤولة عن استشعار مؤشرات فيزيائية مثل درجة الحرارة والرطوبة والحركة والغاز والدخان. تتم عملية المراقبة عبر تطبيق الويب حيث يمكن للمستخدم معاينة هذه المؤشرات في الوقت الحالي على واجهة بيانية كما يمكنه إذا شغل التطبيق في الخلفية استلام إشعار على هاتفه الذكي أو على جهاز الكمبيوتر الخاص به في حالة تجاوز العتبة المحددة للمؤشرات أو وجود حركة في الموقع المراد.

**الكلمات المفتاحية :** إنترنت الأشياء، الأشياء المتصلة، تطوير الويب، تطبيق الويب، المستشعرات، أردوينو، المراقبة.

# RÉSUMÉ

Actuellement, une nouvelle technologie remarquable dans le domaine de l'électronique et des réseaux informatiques offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel. Cette technologie est l'Internet des objets (IoT). Il permet de dessiner une carte numérique du monde réel, donnant ainsi une identité électronique aux objets et aux lieux de l'environnement physique.

Dans ce mémoire, on a appliqué le concept de l'Internet des objets et on a élargi son spectre d'utilisation en réalisant un prototype basé sur l'IoT afin de donner à l'utilisateur la possibilité de surveiller un lieu distant à travers une application Web.

Notre système est composé de trois nœuds chargés de capter des grandeurs physiques telles que température, humidité, mouvement et gaz/fumée. Le monitoring s'effectue via une application Web où l'utilisateur pourra consulter ces grandeurs en temps réel sur une interface graphique ou sera alerté en recevant une notification sur son Smartphone ou sur son ordinateur si l'application se tourne en arrière plan suite au déclenchement d'un événement tel qu'un dépassement de seuil d'une grandeur ou une détection d'un intrus dans le lieu surveillé.

**Mots clés :** Internet des objets, Objets connectés, Développement Web, Application Web, Capteurs, Arduino, Surveillance.

# ABSTRACT

Currently, a remarkable new technology in the field of electronics and computer networks offers a great potential for the development of new services and applications connecting the physical world to the virtual world. This technology is the Internet of Things (IoT). It allows to draw a digital map of the real world, giving an electronic identity to the objects and places of the physical environment.

In this master thesis, the concept of the Internet of Things has been applied and its spectrum of use has been expanded by making an IoT based prototype that gives the user the ability to monitor a distant location through a Web application.

Our system is composed of three nodes responsible for sensing physical quantities such as temperature, humidity, motion and gas/smoke. The monitoring is done via a Web application where the user can consult these quantities in real time on a graphical interface or will be alerted if the application runs in the background by receiving a notification on his Smartphone or his computer following the triggering of an event such as exceeding a threshold of a quantity or detecting an intruder in the monitored location.

**Keywords :** Internet of Things, Connected objects, Web Development, Web Application, Sensors, Arduino, Monitoring.

# TABLE DES MATIÈRES

<b>INTRODUCTION</b>	<b>1</b>
<b>I NOTIONS SUR L'INTERNET DES OBJETS</b>	<b>2</b>
I.1 Evolution historique de l'Internet des objets . . . . .	3
I.2 Définition de l'Internet des objets . . . . .	4
I.3 Les composants d'un modèle d'Internet des objets . . . . .	4
I.3.1 La couche 1, collecter et actionner . . . . .	4
I.3.2 La couche 2, communiquer . . . . .	4
I.3.3 La couche 3, visualiser . . . . .	5
I.4 Domaines d'application de l'Internet des objets . . . . .	5
I.4.1 Ville intelligente . . . . .	6
I.4.2 Domotique . . . . .	7
I.4.3 Agriculture intelligente . . . . .	8
I.4.4 Industrie intelligente . . . . .	8
I.4.5 Santé intelligente . . . . .	9
I.5 Les enjeux de l'Internet des objets . . . . .	10
I.5.1 La sécurité . . . . .	10
I.5.2 La protection de la vie privée des utilisateurs . . . . .	10
I.5.3 L'hétérogénéité . . . . .	10
I.5.4 L'interopérabilité . . . . .	10
I.5.5 La virtualisation . . . . .	11
I.5.6 La transparence . . . . .	11
I.6 Le Web des objets, un niveau plus élevé . . . . .	11
I.7 Internet des objets versus le Web des objets . . . . .	12
I.8 Conclusion . . . . .	13
<b>II ÉTUDE DE LA PARTIE MATÉRIELLE ET LOGICIELLE DU PROJET</b>	<b>14</b>
II.1 Présentation du projet . . . . .	15
II.2 Étude de la partie matérielle . . . . .	16
II.2.1 Les cartes de contrôle . . . . .	16

II.2.1.1	Arduino Uno . . . . .	16
II.2.1.1.a	Caractéristiques . . . . .	16
II.2.1.1.b	Alimentation . . . . .	17
II.2.1.1.c	Mémoire . . . . .	18
II.2.1.1.d	Entrées et sorties numériques . . . . .	18
II.2.1.1.e	Broches analogiques . . . . .	19
II.2.1.1.f	Communication . . . . .	19
II.2.1.2	WeMos D1 Mini . . . . .	19
II.2.1.2.a	Caractéristiques . . . . .	20
II.2.1.2.b	Broches d'entrée/sortie . . . . .	21
II.2.2	Les capteurs . . . . .	21
II.2.2.1	Capteur DHT11 . . . . .	21
II.2.2.2	Capteur MQ-2 . . . . .	22
II.2.2.3	Capteur HC-SR501 . . . . .	23
II.2.3	Le module ESP-01 . . . . .	23
II.2.3.1	Caractéristiques . . . . .	24
II.2.3.2	Broches d'entrée/sortie . . . . .	24
II.2.3.3	Les commandes AT . . . . .	25
II.2.4	Point d'accès . . . . .	25
II.3	Étude de la partie logicielle . . . . .	26
II.3.1	L'environnement de développement Arduino . . . . .	26
II.3.1.1	Structure d'un programme en Arduino . . . . .	27
II.3.2	Application Web . . . . .	28
II.3.3	Cloud Computing . . . . .	28
II.3.3.1	Définition . . . . .	28
II.3.3.2	Caractéristiques . . . . .	29
<b>III</b>	<b>ÉTUDE DES PROTOCOLES DE COMMUNICATION</b>	<b>32</b>
III.1	Protocoles de communication utilisés . . . . .	33
III.1.1	Wi-Fi . . . . .	33
III.1.1.1	Définition . . . . .	33
III.1.1.2	Les modes d'opération . . . . .	33
III.1.1.2.a	Le mode ad hoc . . . . .	33
III.1.1.2.b	Le mode infrastructure . . . . .	33
III.1.1.3	Pourquoi le Wi-Fi? . . . . .	34
III.1.2	HTTP . . . . .	35
III.1.2.1	Définition . . . . .	35
III.1.2.2	La requête HTTP . . . . .	35
III.1.2.3	La réponse HTTP . . . . .	36
III.1.2.4	Pourquoi le HTTP? . . . . .	38

## TABLE DES MATIÈRES

---

<b>IV RÉALISATION DU PROJET</b>	<b>39</b>
IV.1 La phase de réalisation du prototype . . . . .	40
IV.1.1 Les nœuds communicants . . . . .	40
IV.1.1.1 Le nœud 1 . . . . .	40
IV.1.1.1.a Branchement du l'ESP-01 et du capteur DHT11 . . . . .	40
IV.1.1.1.b Algorithme d'acquisition de la température et d'humidité . . . . .	41
IV.1.1.2 Le nœud 2 . . . . .	44
IV.1.1.2.a Branchement du capteur HC-SR501 . . . . .	44
IV.1.1.2.b Algorithme de détection de mouvement . . . . .	44
IV.1.1.3 Le nœud 3 . . . . .	47
IV.1.1.3.a Branchement du capteur MQ-2 . . . . .	47
IV.1.1.3.b Algorithme de détection de gaz/fumée . . . . .	47
IV.1.2 Application Web . . . . .	50
IV.1.2.1 Présentation des interfaces . . . . .	50
IV.1.2.2 Principe de fonctionnement . . . . .	57
<b>CONCLUSION</b>	<b>61</b>
<b>BIBLIOGRAPHIE</b>	<b>62</b>
<b>WEBOGRAPHIE</b>	<b>64</b>

# TABLE DES FIGURES

I.1	Prévision de nombre d'objets connectés selon CisCo IBSG . . . . .	3
I.2	Les composants d'un modèle d'Internet des objets. . . . .	5
I.3	Domaines d'application de l'Internet des objets. . . . .	6
I.4	Ville intelligente. . . . .	7
I.5	Maison intelligente. . . . .	7
I.6	Capteur de température et de la vitesse du vent et taux d'humidité. . . . .	8
I.7	Les quatre révolutions industrielles. . . . .	9
I.8	Santé intelligente . . . . .	9
II.1	Schéma synoptique global du prototype proposé. . . . .	15
II.2	La carte Arduino Uno. . . . .	16
II.3	La WeMos D1 Mini. . . . .	20
II.4	Capteur de température et d'humidité DHT11. . . . .	22
II.5	Capteur MQ-2. . . . .	22
II.6	Capteur HC-SR501. . . . .	23
II.7	Module ESP-01. . . . .	24
II.8	Les huit broches de l'ESP-01. . . . .	24
II.9	Connexion de la WeMos D1 Mini et l'ESP-01 à l'Internet via le point d'accès. . .	25
II.10	L'interface du logiciel Arduino. . . . .	26
II.11	La barre des boutons du logiciel Arduino. . . . .	27
II.12	L'organigramme d'exécution d'un programme en Arduino. . . . .	27
II.13	Schéma donnant un aperçu sur les facteurs principaux du Cloud Computing. . .	28
II.14	Caractéristiques du Cloud Computing. . . . .	30
III.1	Mode de fonctionnement ad hoc du Wi-Fi. . . . .	34
III.2	Mode de fonctionnement infrastructure du Wi-Fi. . . . .	34
III.3	Schéma représentatif des deux scénarios. . . . .	35
IV.1	Schéma électrique du branchement du capteur DHT11 et l'ESP-01 avec l'Uno. . .	41
IV.2	PCB du shield de l'ESP-01 et le capteur DHT11. . . . .	41

## TABLE DES FIGURES

---

IV.3	Organigramme d'acquisition de la température et d'humidité. . . . .	42
IV.4	Capture du moniteur série. . . . .	43
IV.5	Schéma électrique du branchement de la D1 Mini avec le capteur HC-SR501. . .	44
IV.6	Organigramme de détection de mouvement. . . . .	45
IV.7	Capture du moniteur série. . . . .	46
IV.8	Schéma électrique du branchement de la D1 Mini avec le capteur MQ2. . . . .	47
IV.9	Organigramme de détection de gaz/fumée. . . . .	48
IV.10	Capture du moniteur série. . . . .	49
IV.11	Interface d'authentification de la version ordinateur de l'application Web. . . . .	51
IV.12	Interface d'authentification de la version mobile de l'application Web. . . . .	52
IV.13	Interface de contrôle de la version ordinateur de l'application Web. . . . .	53
IV.14	Interface de contrôle de la version mobile de l'application Web. . . . .	54
IV.15	Courbe représentative des dix dernières valeurs de la température. . . . .	55
IV.16	Courbe représentative des dix dernières valeurs de l'humidité. . . . .	56
IV.17	Organigramme d'authentification. . . . .	57
IV.18	Organigrammes des fonctions de contrôle. . . . .	58
IV.19	Organigramme de contrôle. . . . .	59

# LISTE DES TABLEAUX

I.1	Du Web 1.0 au Web 4.0. . . . .	12
II.1	Les caractéristiques de la carte Arduino Uno. . . . .	17
II.2	Broches d'alimentation de la carte Arduino Uno . . . . .	17
II.3	Présentation des broches entrées/sorties numériques de la carte Arduino Uno . . . . .	18
II.4	Les caractéristiques de la WeMos D1 Mini. . . . .	20
II.5	Les différentes broches de la WeMos D1 Mini. . . . .	21
II.6	Les caractéristiques de l'ESP-01. . . . .	24
II.7	Description des différentes broches de l'ESP-01. . . . .	25
III.1	Des méthodes HTTP et leurs fonctions. . . . .	36
III.2	Signification des codes de statut des réponses HTTP. . . . .	37

# LISTE DES ABRÉVIATIONS

<b>AVR</b>	Advanced Virtual RISC
<b>AT</b>	ATtention
<b>API</b>	Application Programming Interface
<b>AP</b>	Access Point
<b>AJAX</b>	Asynchronous JavaScript And XML
<b>BSS</b>	Basic Service Set
<b>CSS</b>	Cascading Style Sheet
<b>CoAP</b>	Constrained Application Protocol
<b>DC</b>	Direct Courant
<b>EEPROM</b>	Electrically Erasable Programmable Read Only Memory
<b>FTDI</b>	Future Technology Devices International
<b>GUI</b>	Graphical User Interface
<b>GPIO</b>	General Purpose Input Output
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTTPS</b>	Hypertext Transfer Protocol Secure
<b>IoT</b>	Internet of Things
<b>ICT</b>	Information and Communication Technologies
<b>ICSP</b>	In-Circuit Serial Programming
<b>I2C</b>	Inter-Integrated Circuit
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IBSS</b>	Independent Basic Service Set
<b>JSON</b>	JavaScript Object Notation
<b>JSP</b>	JavaServer Pages
<b>LED</b>	Light Emitting Diode
<b>MOSI</b>	Master Out Slave In

## LISTE DES ABRÉVIATIONS

---

<b>MISO</b>	Master In Slave Out
<b>MQTT</b>	Message Queuing Telemetry Transport
<b>NTC</b>	Negative Temperature Coefficient
<b>NIST</b>	National Institute of Standards and Technology
<b>OWL</b>	Web Ontology Language
<b>OSI</b>	Open Systems Interconnection
<b>PHP</b>	Hypertext PreProcessor
<b>PWM</b>	Pulse Width Modulation
<b>PCB</b>	Printed Circuit Board
<b>PIR</b>	Passive Infrared Sensor
<b>RFID</b>	Radio frequency IDentification
<b>RSS</b>	Really Simple Syndication
<b>RDF</b>	Resource Description Framework
<b>REST</b>	Representational State Transfer
<b>SWRL</b>	Semantic Web Rule Language
<b>SRAM</b>	Static Random Access Memory
<b>SPI</b>	Serial Peripheral Interface
<b>SS</b>	Slave Select
<b>SCK</b>	Source Clock
<b>SDA</b>	Serial Data Acces
<b>SCL</b>	Serial Clock
<b>SoC</b>	System-on-Chip
<b>SQL</b>	Structured Query Language
<b>SSID</b>	Service Set IDentifier
<b>SSL</b>	Secure Sockets Layer
<b>TTL</b>	Transistor-Transistor Logic
<b>TLS</b>	Transport Layer Security
<b>URL</b>	Uniform Resource Locator
<b>URI</b>	Unified Resource Identifier
<b>VoT</b>	Virtualization of Things
<b>WoT</b>	Web of Things
<b>WPA</b>	Wi-Fi Protected Access
<b>XML</b>	eXtensible Markup Language

# INTRODUCTION

De nos jours, des millions de personnes utilisent l'Internet pour divers usages : de la recherche d'informations sur le Web au jeu en ligne; de l'envoi et de la réception de courriels à des applications sociales et de nombreuses autres activités. Alors que des millions d'appareils nous offrent ces possibilités, un grand pas en avant est fait en ce qui concerne l'utilisation d'Internet en tant que plateforme mondiale permettant aux objets du quotidien de se coordonner et de communiquer entre eux. Et de ce point de vue est né l'Internet des objets (IoT). L'IoT a gagné une part importante de l'esprit, et encore moins de l'attention, dans le milieu universitaire et dans l'industrie en particulier au cours des dernières années. Les raisons de cet intérêt sont les capacités potentielles que l'IoT promet d'offrir. Sur le plan personnel, il brosse une image d'un monde futur où toutes les choses dans notre environnement ambiant sont connectées à l'Internet et communiquent de façon transparente entre elles pour fonctionner intelligemment. Le but ultime est de permettre aux objets autour de nous de capter efficacement notre environnement, de communiquer à peu de frais et de créer un meilleur environnement pour nous, un environnement où les objets quotidiens agissent en fonction de ce dont nous avons besoin sans instructions explicites.

Ce mémoire vise à retracer les aspects fondamentaux de l'Internet des objets, en réalisant un prototype de surveillance à distance basé sur l'IoT, qui permet de relier différents types d'appareils via une application Web au Cloud qui stocke des données envoyées par des cartes de contrôle. Grâce à l'accès en temps réel de l'application Web au Cloud, il est possible de surveiller efficacement un lieu à n'importe quelle distance.

Le développement du prototype est traité à travers quatre chapitres. Le premier chapitre présente une description générale de l'Internet des objets et les concepts qui tournent autour de ce domaine.

Dans le deuxième chapitre, une approche du prototype est établie, les composants matériels et les outils logiciels utilisés sont présentés.

Le troisième chapitre définit les protocoles de communication choisis et les critères de ce choix.

Finalement, le quatrième chapitre détaille la phase de réalisation du prototype, sa construction, l'application Web et les organigrammes associés aux programmes implémentés.

# Chapitre I

## NOTIONS SUR L'INTERNET DES OBJETS

---

Dans ce chapitre, on va aborder les notions de base sur l'Internet des objets, en décortiquant son architecture et les composants basiques qui le constituent avec les différentes applications associées, tout en citant les enjeux les plus marquants évoqués par ce concept. Ainsi, on va terminer ce chapitre par le Web des objets qui représente un niveau supérieur de l'Internet des objets.

---

## I.1 Evolution historique de l'Internet des objets

L'Internet des objets est un concept concrétisant la vision de l'informatique ubiquitaire telle qu'imaginée en 1991 par Mark Weiser, où la technologie s'efface peu à peu dans l'environnement des utilisateurs, intégrée naturellement à l'intérieur des objets du quotidien. La technologie n'est plus alors représentée par un objet unique, l'ordinateur personnel, mais se présente au contraire sous la forme d'appareils spécialisés et simples d'emploi, capables de communiquer au travers de plusieurs types de réseaux sans fil : liseuses numériques, télévisions et montres connectées, ordinateurs de bord, téléphones intelligents, etc.

À l'origine, le terme Internet des objets a été utilisé pour la première fois en 1999 par Kevin Ashton pour décrire des objets équipés de puces d'identification par radiofréquence (RFID). Chaque objet, identifié de manière unique et universelle, peut alors être rattaché à un ensemble d'informations le concernant, ces dernières étant lisibles par d'autres machines. Caractéristiques, état courant et position sont alors autant de métadonnées échangées entre les objets, formant un nouveau réseau qui leur dédié : l'Internet des objets.

Le concept a toutefois évolué avec le temps et s'est généralisé vers une approche consistant à connecter un très grand nombre d'objets du quotidien au réseau Internet, les dotant ainsi d'une identité propre et leur permettant, entre autres, d'offrir des services et de collecter des informations de manière autonome [B1].

En ce qui concerne l'avenir, CisCo IBSG estime que 25 milliards d'appareils seront connectés à l'Internet d'ici à 2015 et 50 milliards, d'ici à 2020. Il est important de noter que ces estimations ne tiennent pas compte des progrès rapides d'Internet ni des avancées technologiques, mais reposent uniquement sur les faits avérés à l'heure actuelle [W1].

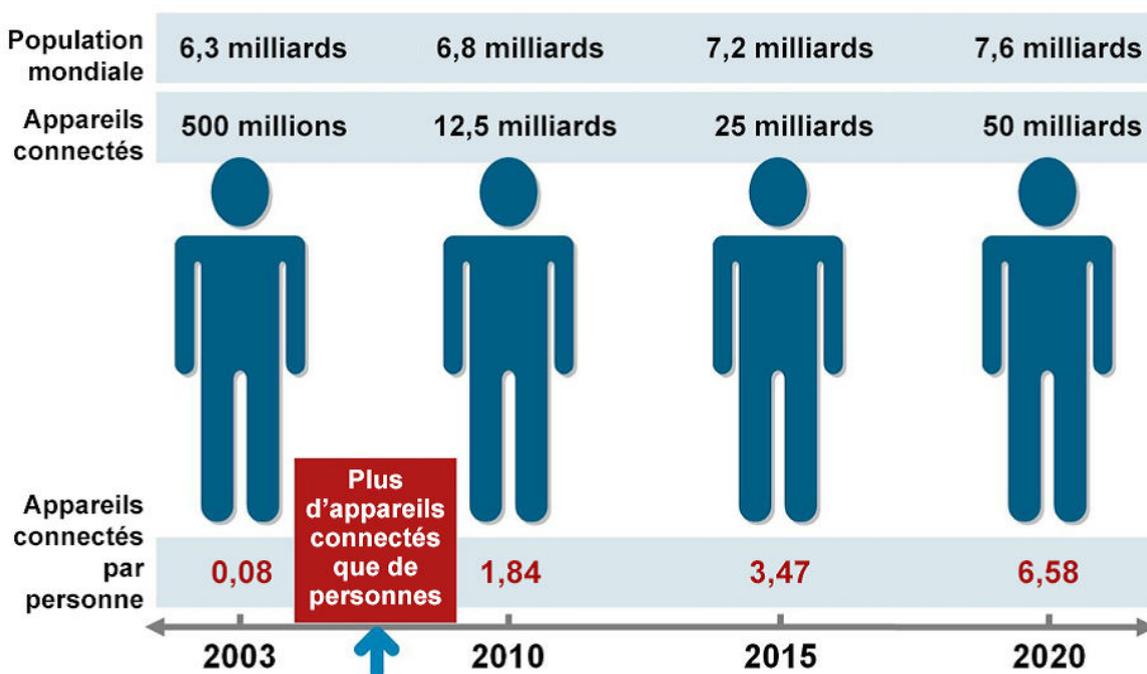


FIGURE I.1 – Prévision de nombre d'objets connectés selon CisCo IBSG [W1].

## I.2 Définition de l'Internet des objets

L'Internet des objets sera le résultat de l'interconnexion du monde physique et d'Internet au moyen de capteurs mémoires, etc. Grâce à de telles structures, le réseau lui-même devient à la fois calculateur universel, mémoire universelle et bibliothèque ouverte de toutes les applications ; en un mot, il devient le siège d'une nouvelle forme d'intelligence collective [B2].

Certains définissent l'Internet des objets comme des objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer. D'autres font l'hypothèse que l'IoT représente une révolution car il permet de connecter les gens et les objets n'importe où, n'importe quand, par n'importe qui. Ces définitions, qui mettent l'accent sur la dimension ubiquitaire de l'IoT, personnifient les objets en leur attribuant intelligence et capacité de communiquer. Elles ne reflètent pas encore la dimension concrète liée aux usages de l'IoT.

Techniquement, l'Internet des objets est une extension du système de nommage Internet et traduit une convergence des identifiants numériques au sens où il est possible d'identifier de manière unifiée des éléments d'information numérique (adresses URL de sites Web par exemple) et des éléments physiques (comme une palette dans un entrepôt, ou encore un mouton dans un cheptel). Mais l'identification est directe grâce à l'utilisation d'un système d'identification électronique (puces RFID, processeur et communication Bluetooth, etc.). Il n'y a pas besoin de saisir manuellement le code de l'objet. Le réseau s'étend jusqu'à lui et permet ainsi de créer une forme de passerelle entre les mondes physique et virtuel [B3].

## I.3 Les composants d'un modèle d'Internet des objets

Tous les composants sont classés ci-dessous sur la base d'un modèle à trois couches (Figure I.2).

### I.3.1 La couche 1, collecter et actionner

Cette couche se compose de :

- Les capteurs : ils détectent et répondent à un type d'entrée provenant de l'environnement physique en le convertissant en signaux numériques.
- Les actionneurs : ils exécutent des actions à des moments précis.
- End-devices : ils sont des petites cartes électroniques avec un microcontrôleur ( $\mu\text{C}$ ) intégré qui a pour rôle de traiter les données venant du capteur et d'envoyer des commandes vers l'actionneur.

### I.3.2 La couche 2, communiquer

Cette couche comprend :

- Les protocoles de communication.
- Passerelle : pour relayer l'information entre les End-devices et l'Internet.

### I.3.3 La couche 3, visualiser

Cette couche comprend :

- Les plateformes IoT Cloud : conçues pour stoker et traiter des données volumineuses générées par les End-devices, et donnent la visualisation de ces données (tableau, graphe) pour les utilisateurs finaux.
- L'application software : fournit des interfaces graphiques (GUI) pour la surveillance et le contrôle des End-devices.

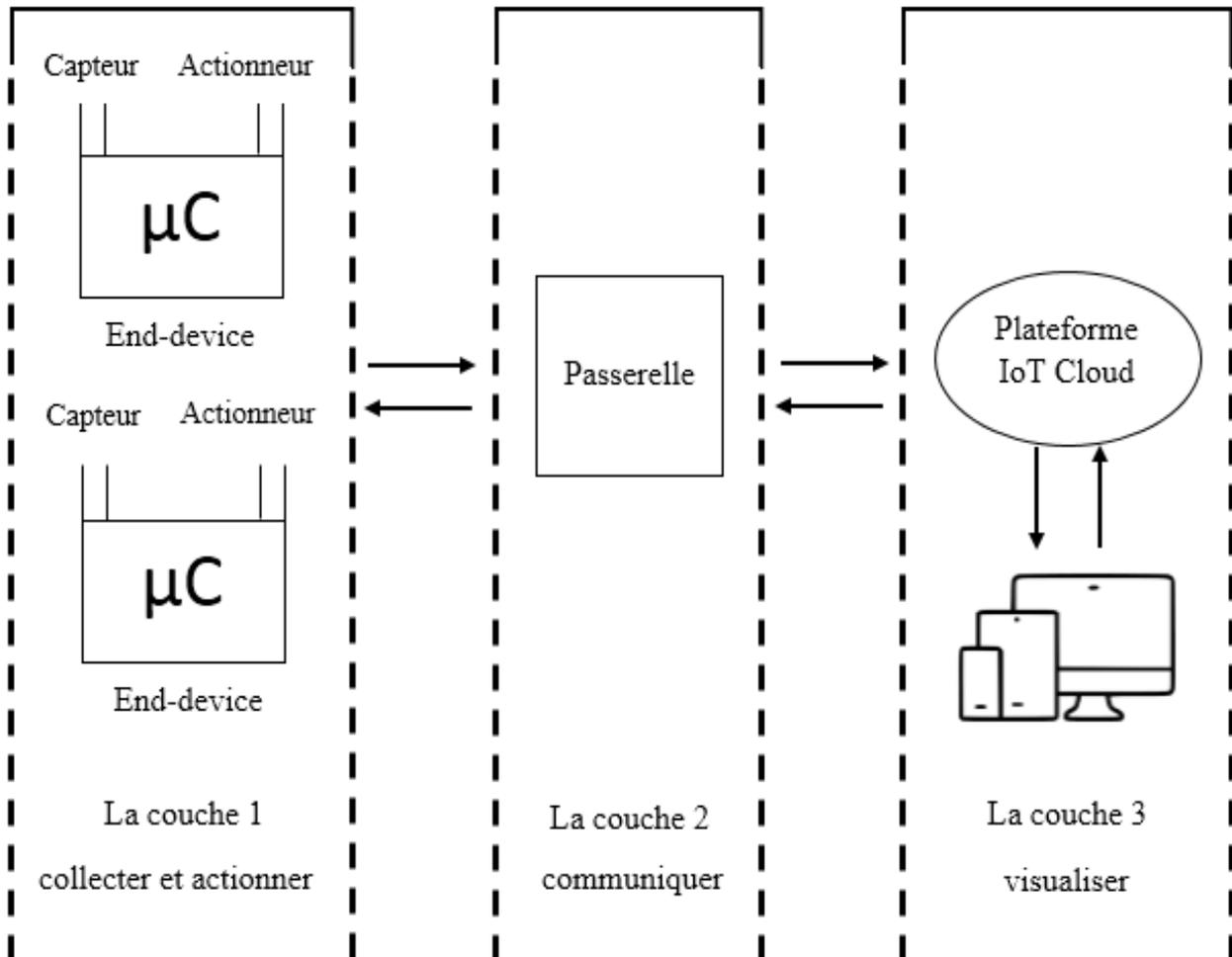


FIGURE I.2 – Les composants d'un modèle d'Internet des objets.

## I.4 Domaines d'application de l'Internet des objets

Aujourd'hui la notion de l'Internet des objets est en pleine explosion du fait que l'existence des objets intelligents nous permet d'atteindre nos objectifs dans la vie quotidienne plus facilement. Ainsi, les domaines d'applications de l'IoT sont différents (Figure I.3).

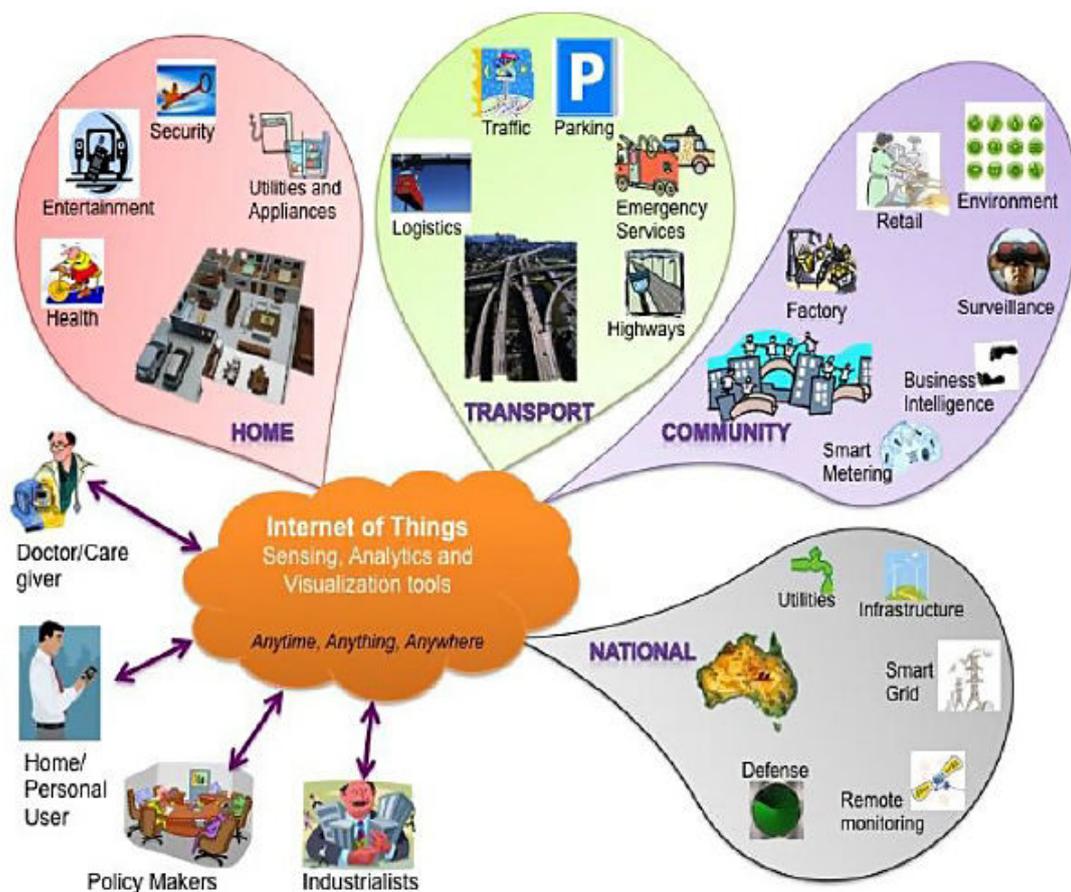


FIGURE I.3 – Domaines d'application de l'Internet des objets.

Nous pouvons affirmer que l'Internet peut être connecté à n'importe quel objet. De ce fait, les domaines d'applications de l'IoT sont nombreux. On cite ci-dessous quelques applications.

### I.4.1 Ville intelligente

Une ville intelligente est une zone urbaine qui utilise différents capteurs de collecte de données électroniques pour fournir des informations permettant de gérer efficacement les ressources et les actifs. Cela comprend les données collectées auprès des citoyens, des dispositifs mécaniques, des actifs, traitées et analysées pour surveiller et gérer les systèmes de circulation et de transport, les centrales électriques, les réseaux d'approvisionnement en eau, la gestion des déchets, les systèmes d'information, les écoles, les bibliothèques et les hôpitaux.

La technologie des villes intelligentes permet aux représentants municipaux d'interagir directement avec les infrastructures communautaires et urbaines et de surveiller la ville et son évolution. Les technologies de l'information et de la communication (ICT) sont utilisées pour améliorer la qualité, la performance et l'interactivité des services urbains, réduire les coûts et la consommation de ressources et accroître les contacts entre les citoyens et le gouvernement [B4].

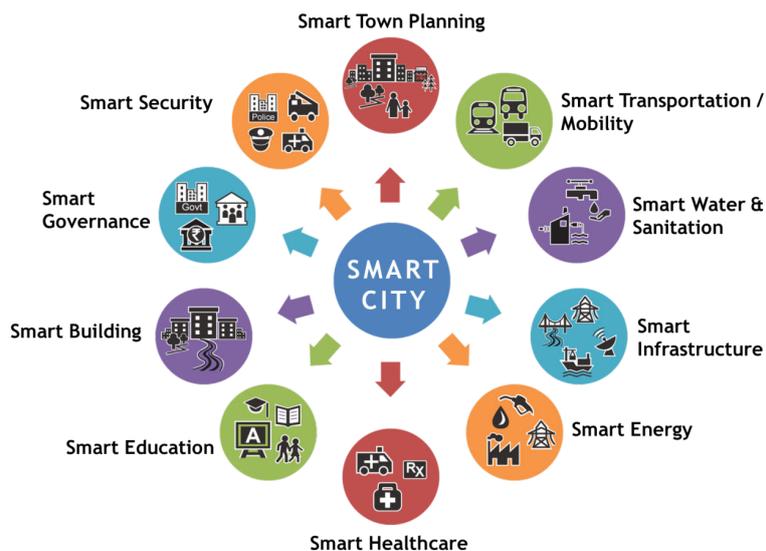


FIGURE I.4 – Ville intelligente.

### I.4.2 Domotique

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins interopérables et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc [B5].

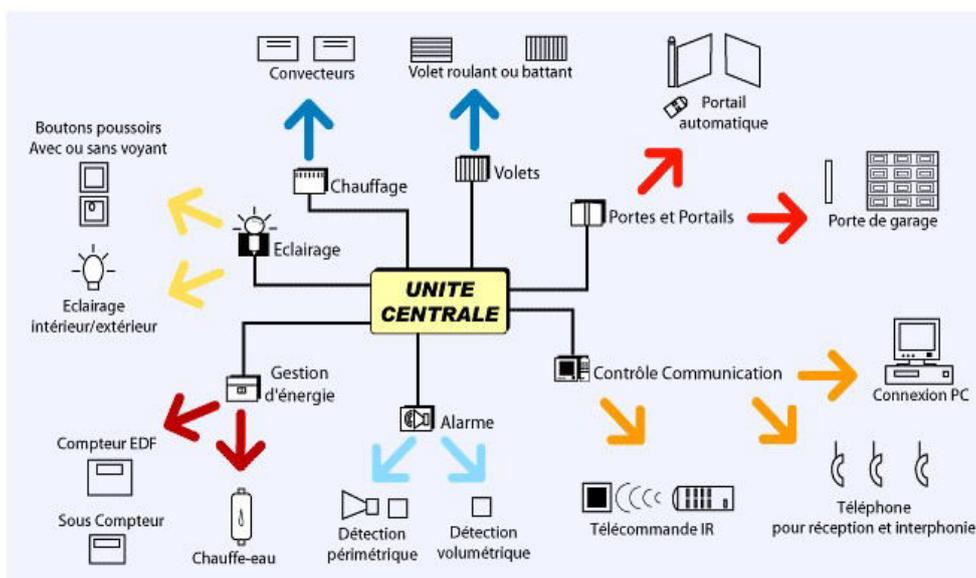


FIGURE I.5 – Maison intelligente.

### I.4.3 Agriculture intelligente

L'agriculture intelligente a pour objet de renforcer la capacité des systèmes agricoles, de contribuer à la sécurité alimentaire, en intégrant le besoin d'adaptation et le potentiel d'atténuation dans les stratégies de développement de l'agriculture durable [W2].

Cet objectif requiert l'utilisation de nouvelles technologies, telles que l'imagerie satellitaire et l'informatique. Il s'appuie sur des capteurs qui vont sonder l'environnement et remonter les informations utiles sur l'état du sol, taux d'humidité, taux des sels minéraux, etc. au fermier. Ce dernier va exploiter ces informations pour assurer une bonne production [B6].



FIGURE I.6 – Capteur de température et de la vitesse du vent et taux d'humidité.

### I.4.4 Industrie intelligente

L'industrie intelligente ou l'industrie 4.0, c'est la quatrième révolution industrielle. La première, c'est la production mécanique en utilisant les machines à vapeur, alors que la seconde s'est servie de l'arrivée de l'électricité pour faire la production de masse. La troisième, c'est la production automatisée. La quatrième révolution industrielle, pour sa part, ne résulte pas d'une cassure technologique comme les révolutions précédentes. Elle vient plutôt de l'utilisation des nouvelles technologies basées sur l'Internet des objets pour interconnecter et synchroniser les différents systèmes de l'usine. L'IoT donne aux fabricants la possibilité de suivre des objets, pour savoir comment les consommateurs utilisent certains produits et pour déterminer quelles caractéristiques

sont primordiales. Ils peuvent ainsi mieux cerner les ajustements à apporter pour augmenter les taux d'adoption et d'achat. Les entreprises veulent avoir connaissance de ce que font les utilisateurs avec les produits pour en tirer parti et l'IoT leur en donne la possibilité [W3].

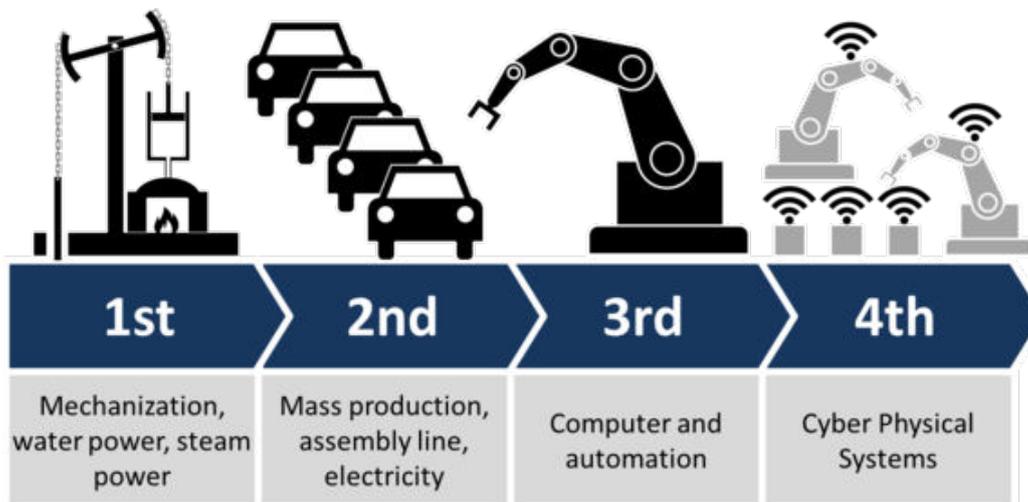


FIGURE I.7 – Les quatre révolutions industrielles.

### I.4.5 Santé intelligente

Dans le domaine de la santé, les objets connectés peuvent servir à plusieurs choses. On peut par exemple les utiliser comme dispositif de surveillance ou de monitoring sous forme de bracelet connecté ou de montre connectée qui permettront de suivre à tout moment nos activités physiques en nous informant de signes irréguliers.

Si vous êtes une personne qui prend des médicaments ces objets connectés peuvent également vous informer quand il est temps de prendre vos médicaments. En outre, cela pourrait éventuellement informer votre médecin d'une situation d'urgence lui permettant ainsi de vous localiser grâce à votre appareil [B7].

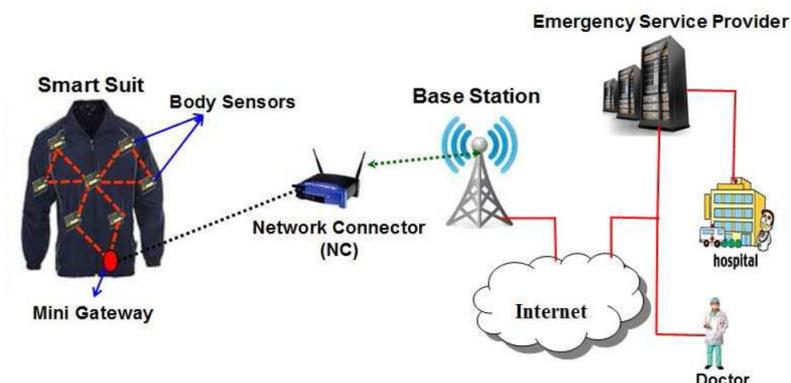


FIGURE I.8 – Santé intelligente

## **I.5 Les enjeux de l'Internet des objets**

Bien que l'Internet des objets soit un concept qui est à la fois avantageux et prometteur, et qui pourra apporter des solutions efficaces aux problèmes de suivi et de télésurveillance dans différents domaines. En contrepartie, l'IoT soulève certaines questions décisives, étroitement liées à sa maturité et son acceptabilité [B8]. On cite ci-dessous les enjeux les plus marquants.

### **I.5.1 La sécurité**

La sécurité des personnes, des communications, des données, des services, des réseaux et des équipements continue à présenter un problème majeur dans l'Internet courant. Aujourd'hui avec la naissance de l'Internet des objets, l'amplitude du problème va prendre un autre ordre de gravité. Des milliers d'objets contraints connectés en permanence à l'Internet et intégrés dans toute sorte d'objets dans notre vie quotidienne, vont porter le risque d'être ciblés par les menaces classiques de l'Internet. Il est même possible que de nouvelles générations d'attaques apparaissent. Donc, les objets intelligents dans l'IoT, la transmission et le stockage de leurs données sur l'Internet devraient être sécurisés.

D'autre part, l'IoT lui-même peut menacer la sécurité des individus et les institutions. L'armée chinoise proscrit les officiers et les soldats de porter des objets connectés (comme les montres et les lunettes connectées à l'Internet) et considère leur utilisation comme une violation de la réglementation sur le secret dans les casernes.

### **I.5.2 La protection de la vie privée des utilisateurs**

Un grand nombre de capteurs connectés à l'Internet et intégrés dans des objets d'usage quotidien révèlent nos habitudes, notre état de santé, notre localisation géographique, et autres types d'informations privées. Il devra absolument y avoir des mécanismes robustes qui assurent la confidentialité des données que l'utilisateur qualifie être sensibles. Les utilisateurs devraient également pouvoir savoir qui accède quelles données (concernant les utilisateurs) sur l'Internet et pour quelle raison.

### **I.5.3 L'hétérogénéité**

Des dispositifs de divers types ayant des capacités variées et appartenant à des réseaux de différentes natures, vont intégrer l'Internet en utilisant différentes technologies de communication (filaire, sans fil, satellitaire, etc.). Avec toutes ces formes d'hétérogénéités matérielles et technologiques, il serait primordial de mettre en place des mécanismes bien avertis qui soient capables d'en cacher et gérer.

### **I.5.4 L'interopérabilité**

L'interopérabilité constitue l'un des plus grands défis de la réalisation de l'Internet des objets.

Elle est, en réalité, la cohabitation des dispositifs, des systèmes et des mécanismes disjoints et la possibilité de les faire coopérer et interagir en toute flexibilité. Une tendance récente tend vers la standardisation et l'unification des systèmes et protocoles opérationnels dans l'IoT et de présenter un accès libre afin de faciliter la collaboration entre les objets connectés, ainsi que le couplage avec les entités externes qui se trouvent sur l'Internet.

### **I.5.5 La virtualisation**

Plusieurs capteurs connectés peuvent représenter un seul capteur virtuel qui rapporte une mesure virtuelle résultant de l'agrégation de plusieurs états secondaires. Par exemple, un capteur virtuel qui nous dit si l'état de santé d'un patient est bon ou non. Cette information n'est qu'une combinaison de plusieurs informations fournies par plusieurs capteurs médicaux réels incorporés dans le corps du patient. Ainsi, un modèle générique de virtualisation des objets connectés à l'IoT, nommé VoT permet une représentation abstraite des objets et l'accumulation des données qui en proviennent, depuis différents endroits, pour faciliter leur contrôle.

### **I.5.6 La transparence**

L'objectif de l'informatique transparente est de rendre les systèmes informatiques des boîtes noires transparentes à travers des communications sans fil, automatiques et invisibles ne nécessitant pas d'interaction avec les utilisateurs. La transparence est la base de l'informatique pervasive qui est à son tour un facteur essentiel dans l'Internet des objets.

## **I.6 Le Web des objets, un niveau plus élevé**

D'après [B9], l'idée Web des objets (WoT) est d'intégrer des objets au Web, afin de les faire communiquer avec des humains ou d'autres machines. Le Web, ou World Wide Web pour être exact, étant un ensemble de technologies permettant d'interroger des ressources, identifiées par des URI, accessibles sur Internet. Ces ressources contiennent des informations textes structurées (HTML, JSON, XML) ou binaires (images, vidéos, etc.). Dans bien des cas, le Web est considéré comme un système d'information, composé de ressources liées entre elles par des hyper-liens.

Un point important du Web des objets est de fournir un protocole de communication commun entre les objets. En effet, l'utilisation du HTTP, le même que pour le Web, permet une unification au niveau de la communication, de l'interrogation de ressources et de la conception. Puisque tous les objets parlent la même langue, donc ils répondent de la même manière à la même requête.

L'utilisation des technologies Web adaptées aux objets fournit la même facilité d'interaction que pour n'importe quel site Web. Par exemple, l'utilisateur a uniquement besoin d'un navigateur Web, de plus n'importe quel moteur d'indexation, tel Google, peut y accéder et tous les clients HTTP sont utilisables. Un moteur de recherche pourrait tout à fait permettre la recherche d'objets selon différents critères et la création d'applications client n'est pas plus compliquée que pour un site classique.

L'utilisation d'un protocole différent, ou propriétaire ne permet pas cette facilité d'utilisation, c'est la raison principale d'utiliser le Web des objets.

Cependant, des inconvénients existent également comme l'importante exigence en ressources informatiques (mémoire, processeurs, etc.). Ainsi, le protocole de communication HTTP est de type énergivore par rapport à certains protocoles développés pour l'Internet des objets [B10].

	<b>Apparition</b>	<b>Description</b>	<b>Technologies</b>
<b>Web 1.0</b>	1995	Page HTML statique	HTML, HTTP
<b>Web 1.5</b>	1997	Contenu HTML dynamique	Coté client (JavaScript, Flash, etc.) Coté serveur (PHP, JSP, etc.)
<b>Web 2.0</b>	2003	Partage, échange d'informations et de contenus (image, video, text, etc.)	Web services, blogs, fils RSS, XML, RDF, OWL
<b>Web 3.0</b>	2008	Web sémantique	Amélioration des technologies RDF, OWL et l'apparition du SWRL
<b>Web 4.0 (WoT)</b>	2010	Des objets physiques sont complètement intégrés au Web	Evolution vers des standards ouverts, vers un langage universel (URI, REST)

TABLE I.1 – Du Web 1.0 au Web 4.0.

## I.7 Internet des objets versus le Web des objets

Vouloir comparer le Web des objets à l'Internet des objets ne fait pas de sens. Le niveau d'abstraction du WoT étant plus élevé que celui du l'IoT, la comparaison devient impossible. Tout comme le Web fait partie d'Internet, avec un niveau d'abstraction plus élevé, le WoT appartient également à l'IoT [B9]. Le WoT est un domaine du l'IoT, dans lequel la communication se fait à travers le protocole HTTP. Cependant, au vu des développements actuels, certaines personnes tendent à opposer l'IoT et le WoT. Dans certains articles et prises de positions, l'IoT est considéré comme toutes les technologies relatives au monde des objets, exceptées les technologies Web.

Les avantages du Web des objets sont les faiblesses de l'Internet des objets, et vice versa. En effet, les avantages du WoT (simplicité d'utilisation, façon de communiquer très normée, même langage utilisé entre objets, etc.) peuvent être des désavantages dans certaines situations. Tandis que la grande souplesse de l'IoT peut aussi être un avantage. Par exemple, un objet fonctionnant sur batterie, ayant peu d'énergie électrique à disposition devra être conçu de manière à économiser cette énergie et utilisant un protocole plus économe que le HTTP.

Tout dépend de l'utilisation que l'on veut en faire, du type d'objet à connecter et le but dans lequel il a été construit. C'est également une affaire de philosophie suivant que l'on veut restreindre l'accès à l'objet en utilisant un protocole propriétaire, ou au contraire, permettre au plus de monde possible d'y accéder.

### **I.8 Conclusion**

Tout au long de ce chapitre, on a vu les notions de base sur l'Internet des objets, on a illustré son architecture, on a donné quelques domaines d'applications et les enjeux les plus importants. Ensuite, on a clarifié le concept de Web des objets pour enlever l'ambiguïté entre l'Internet des objets et le Web des objets.

# Chapitre **II**

## ÉTUDE DE LA PARTIE MATÉRIELLE ET LOGICIELLE DU PROJET

---

Ce chapitre est dédié à la présentation du schéma synoptique du projet. En premier lieu, on va décrire les différents composants matériels qui le constituent, puis, on va énumérer les outils informatiques utilisés lors de la réalisation du projet.

---

## II.1 Présentation du projet

Comme on a vu dans le chapitre précédent, un modèle d'Internet des objets est constitué de trois principales couches. Notre prototype proposé suit cette même décomposition.

La première couche est conçue autour de trois nœuds (End-device); chacune est équipée d'un microcontrôleur, un module Wi-Fi, et un capteur pour collecter différents types d'informations (température et humidité, mouvement, gaz/fumée). La constitution de chaque nœud est comme suit :

- Nœud 1 : une carte Arduino Uno, un module ESP-01 et un capteur DHT11.
- Nœud 2 : une carte WeMos D1 Mini et un capteur HC-SR501.
- Nœud 3 : une carte WeMos D1 Mini et un capteur MQ-2.

Pour la deuxième couche, on va se servir d'un point d'accès sans fil (Wi-Fi) pour connecter les nœuds à l'Internet afin d'envoyer les informations collectées.

Dans la troisième couche, on a utilisé des plateformes IoT Cloud pour recevoir les informations envoyées par les nœuds, de plus, on a développé une application Web qui accomplit les fonctions suivantes :

- Créer une visualisation des données stockées dans le Cloud.
- Afficher des notifications à des événements spécifiques.

Le prototype proposé est schématisé dans la figure ci-dessous :

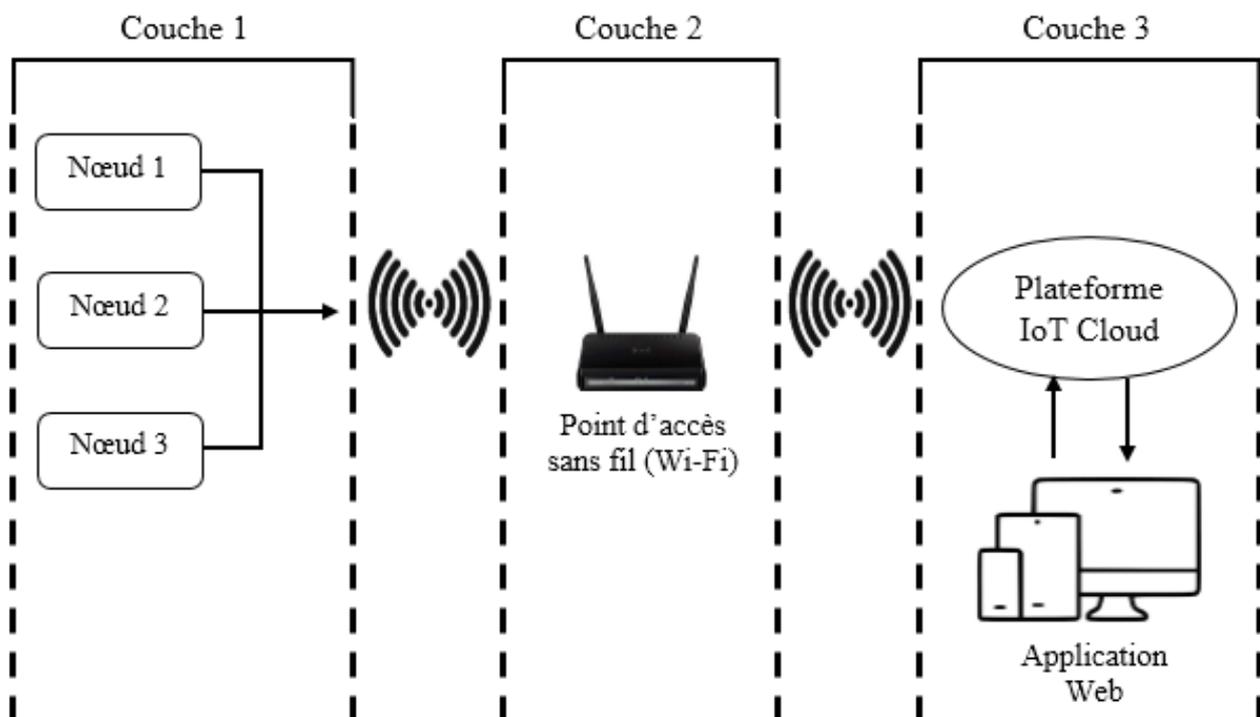


FIGURE II.1 – Schéma synoptique global du prototype proposé.

## II.2 Étude de la partie matérielle

Dans cette partie, on va décrire les différents composants utilisés dans notre projet.

### II.2.1 Les cartes de contrôle

#### II.2.1.1 Arduino Uno

La carte Arduino est un circuit électronique imprimé distribué sous la licence du matériel libre (les plans de la carte sont disponible gratuitement), sur lequel se trouve un microcontrôleur (mini-ordinateur) ATMEL AVR qui peut être programmé pour analyser et produire des signaux électriques de manière à effectuer des tâches (contrôler des appareils domestiques, faire un jeu de lumières, télécommander un appareil mobile, etc.).

L'Arduino Uno (Figure II.2) est l'une des cartes les plus populaires de la famille Arduino dont le cœur est un microcontrôleur de référence ATmega328.

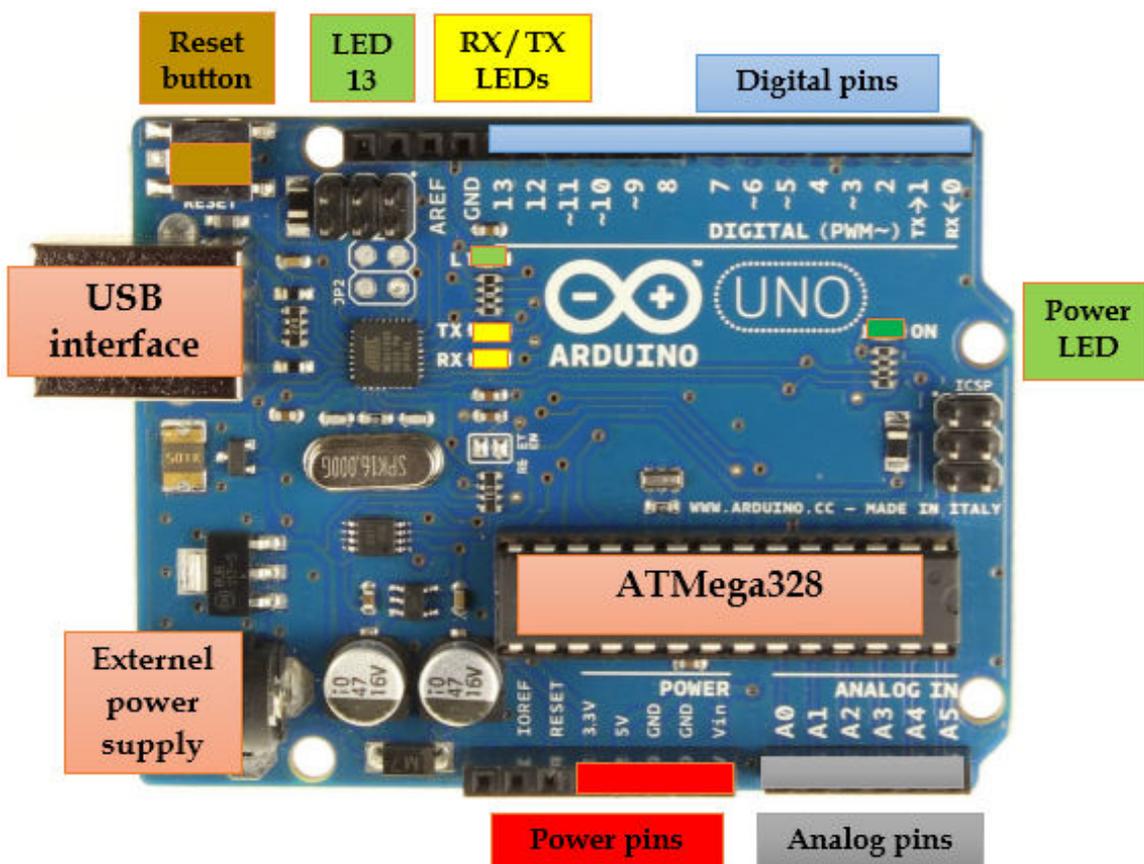


FIGURE II.2 – La carte Arduino Uno.

##### II.2.1.1.a Caractéristiques

Le tableau suivant résume les caractéristiques de la carte Arduino Uno.

<b>Microcontrôleur</b>	ATMega328
<b>Tension de fonctionnement</b>	5V
<b>Tension d'alimentation</b>	7-12V (recommandée) 6-20V (limite)
<b>Entrées /sorties numériques</b>	14 (dont 6 peuvent être utilisées en sortie PWM)
<b>Entrées analogiques</b>	6
<b>Courant DC nécessaire pour les broches entrée /sortie</b>	40 mA
<b>Mémoire FLASH</b>	32 KB (dont 0.5 KB utilisé par le bootloader)
<b>Mémoire SRAM</b>	2 KB
<b>Mémoire EEPROM</b>	1 KB
<b>Fréquence d'horloge</b>	16 MHz
<b>Longueur</b>	68.6 mm
<b>Largeur</b>	53.4 mm

TABLE II.1 – Les caractéristiques de la carte Arduino Uno.

### II.2.1.1.b Alimentation

On peut alimenter la carte Arduino Uno soit par une alimentation interne via la connexion USB ou externe (non-USB) par le biais d'un adaptateur secteur ou batterie. La source d'alimentation est automatiquement sélectionnée par la carte.

Les broches d'alimentation sont les suivantes :

<b>VIN</b>	La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe. On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
<b>5V</b>	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.
<b>3V3</b>	Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATMega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA.
<b>GND</b>	Broche de masse (0V)

TABLE II.2 – Broches d'alimentation de la carte Arduino Uno [W4].

### II.2.1.1.c Mémoire

La carte Arduino Uno a une mémoire FLASH de 32 KB utilisée pour stocker l’image du programme et toutes les données (dont 0.5 KB utilisé par le bootloader). Comme on trouve aussi 2 KB de mémoire SRAM et 1 KB d’EEPROM.

### II.2.1.1.d Entrées et sorties numériques

Dans la carte Arduino Uno, on trouve des broches numériques (numérotées de 0 à 13) qui fonctionnent en 5V, chacune de ces broches fournissent ou reçoivent au maximum 40mA d’intensité, elles peuvent être aussi utilisées comme des entrées numériques, ou comme des sorties, en utilisant les instructions `pinMode(broche, OUTPUT)` pour la sortie et `pinMode(broche, INPUT)` pour l’entrée du langage Arduino.

De plus, certaines broches ont des fonctions spécialisées :

<b>Communication Série</b>	Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l’interface entre les niveaux TTL et le port USB de l’ordinateur.
<b>Interruptions Externes</b>	Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. (Voir la fonction <code>attachInterrupt()</code> ).
<b>Impulsion PWM</b>	Broches 3, 5, 6, 9, 10, et 11. Elles fournissent une impulsion PWM 8 bits à l’aide de la fonction <code>analogWrite()</code> .
<b>SPI</b>	Broches 10 (SS), 11 (MOSI), 13 (SCK), 11 (MISO). 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 Mega.
<b>I2C</b>	Broches 4 (SDA) et 5 (SCL). Elles supportent les communications de protocole I2C, disponible en utilisant la librairie <code>Wire/I2C</code> .
<b>LED</b>	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.

TABLE II.3 – Présentation des broches entrées/sorties numériques de la carte Arduino Uno [W4].

### II.2.1.1.e Broches analogiques

La carte Arduino Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c-à-d sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre 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 la fonction `analogReference()` du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19 [W4].

Il existe deux autres broches disponibles sur la carte, la broche étiquetée AREF qui représente une tension de référence pour les entrées analogiques. Utilisée avec la fonction `analogReference()` du langage Arduino et la broche étiquetée RESET, cette broche permet de réinitialiser l'Arduino. Pour cela, il suffit de passer cette broche à 0V puis de la repasser à 5V.

### II.2.1.1.f Communication

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATMega328 dispose d'un émetteur-récepteur asynchrone universel (UART) 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 vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATMega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier `.inf` est requis.

Le logiciel Arduino inclut une fenêtre terminal série ou moniteur série sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'Arduino (mais pas pour les communications série sur les broches 0, 1) [W4].

### II.2.1.2 WeMos D1 Mini

La WeMos D1 Mini (Figure II.3) est une carte de développement Wi-Fi basée sur le microcontrôleur ESP-8266EX; cette carte est une solution très compacte pour le prototypage de petits objets intelligents liés au World Wide Web grâce aux fonctionnalités ESP8266 Wi-Fi.

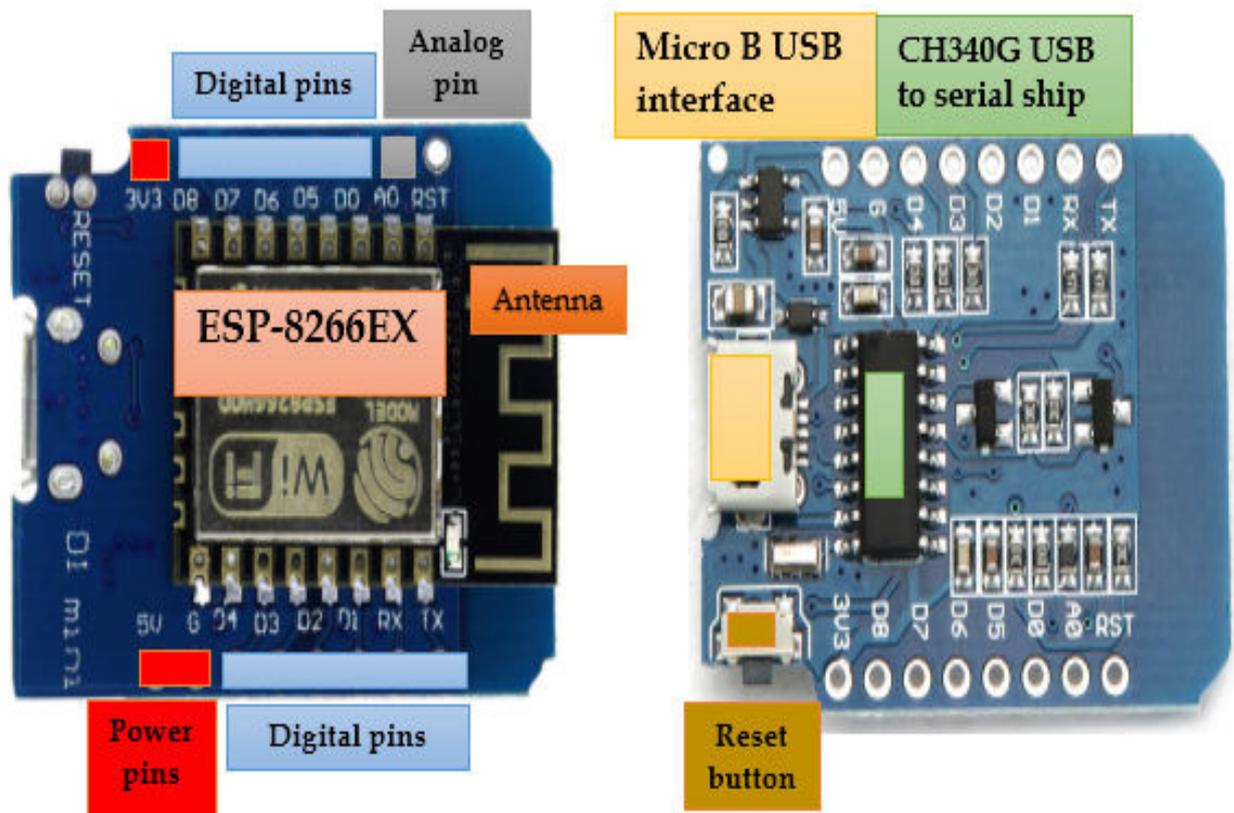


FIGURE II.3 – La WeMos D1 Mini.

### II.2.1.2.a Caractéristiques

Parmi les caractéristiques de la WeMos D1 Mini, on trouve :

<b>Microcontrôleur</b>	ESP-8266EX
<b>Convertisseur(série-USB)</b>	CH340G
<b>Wi-Fi</b>	802.11 b/g/n
<b>Type d'antenne</b>	Antenne PCB
<b>Tension de fonctionnement</b>	3.3V
<b>Entrées /sorties numériques</b>	11 (3.3V max)
<b>Entrées analogiques</b>	1 (3.2V max)
<b>Mémoire FLASH</b>	4 MB
<b>Fréquence d'horloge</b>	80/160 MHz
<b>Longueur</b>	34.2 mm
<b>Largeur</b>	25.6 mm

TABLE II.4 – Les caractéristiques de la WeMos D1 Mini.

La WeMos D1 Mini est alimentée par 5V via une connexion micro USB de type B ; elle embarque une mémoire FLASH de 4 MB pour stocker les programmes.

### II.2.1.2.b Broches d'entrée/sortie

Le tableau suivant regroupe toutes les broches de la WeMos D1 Mini en indiquant leurs fonctions.

<b>TX</b>	Donnée série, transmettre
<b>RX</b>	Donnée série, recevoir
<b>A0</b>	Entrée analogique, max 3.2V
<b>D0</b>	Entrée/sortie numérique
<b>D1</b>	Entrée/sortie numérique (SCL)
<b>D2</b>	Entrée/sortie numérique (SDA)
<b>D3</b>	Entrée/sortie numérique, résistance pull-up (résistance de tirage) de 10k
<b>D4</b>	Entrée/sortie numérique, résistance pull-up (résistance de tirage) de 10k, LED interne
<b>D5</b>	Entrée/sortie numérique (SCK), cette broche supporte la communication SPI (Interface Série Périphérique)
<b>D6</b>	Entrée/sortie numérique (MISO), cette broche supporte la communication SPI (Interface Série Périphérique)
<b>D7</b>	Entrée/sortie numérique (MOSI), cette broche supporte la communication SPI (Interface Série Périphérique)
<b>D8</b>	Entrée/sortie numérique (SS), résistance Pull-up (résistance de tirage) de 10k, cette broche supporte la communication SPI (Interface Série Périphérique)
<b>G</b>	La masse (0V)
<b>5V</b>	5V
<b>3.3V</b>	3.3V
<b>RST</b>	Placer à la masse pour réinitialiser

TABLE II.5 – Les différentes broches de la WeMos D1 Mini.

## II.2.2 Les capteurs

Pour des raisons de coût et principalement de disponibilité, on a choisi les capteurs suivants : DHT11, MQ-2 et HC-SR501.

### II.2.2.1 Capteur DHT11

Le DHT11 (Figure II.4) est un capteur d'humidité et de température très répandu dans le contrôle de climatisation. Il comprend un élément résistif et des dispositifs NTC de mesure de la température et de détection de l'humidité.

L'interface physique du capteur est réalisée par un connecteur à 4 broches, seulement 3 broches sont utilisées : Vcc, GND et S. Les deux premières servent à l'alimentation et la masse, la troisième est pour les données.

Ce capteur est caractérisé par :

- Une tension d'alimentation : 3.3-5V.
- Température : de 0 à 50°C, précision : +/- 2 °C.
- Humidité : de 20 à 95% RH, précision : +/- 5% RH.

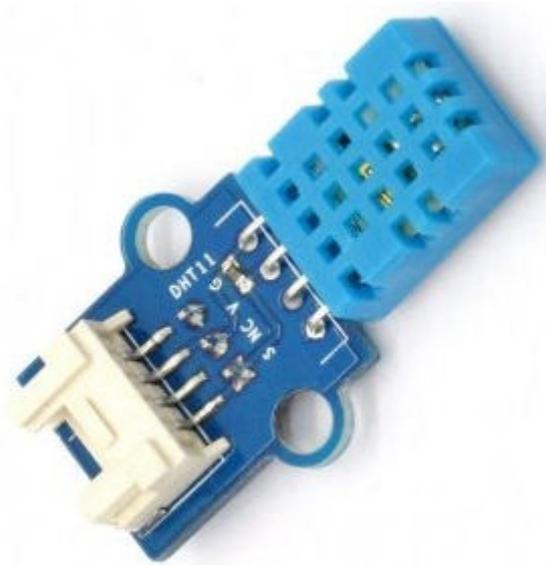


FIGURE II.4 – Capteur de température et d'humidité DHT11.

### II.2.2.2 Capteur MQ-2

Le MQ-2 (Figure II.5) est un capteur de gaz inflammable et de fumée. Il détecte la concentration des différents gaz comme le GPL (LPG), l'i-butane, le propane, le méthane, l'alcool, l'hydrogène et les fumées dans l'air, puis il renvoie sa lecture comme une tension analogique. MQ-2 a une grande sensibilité avec un temps de réponse rapide. Sa sensibilité peut d'ailleurs être ajustée par un potentiomètre. Ce capteur fonctionne à des températures de -10 à 50°C et consomme moins de 150 mA à 5V.

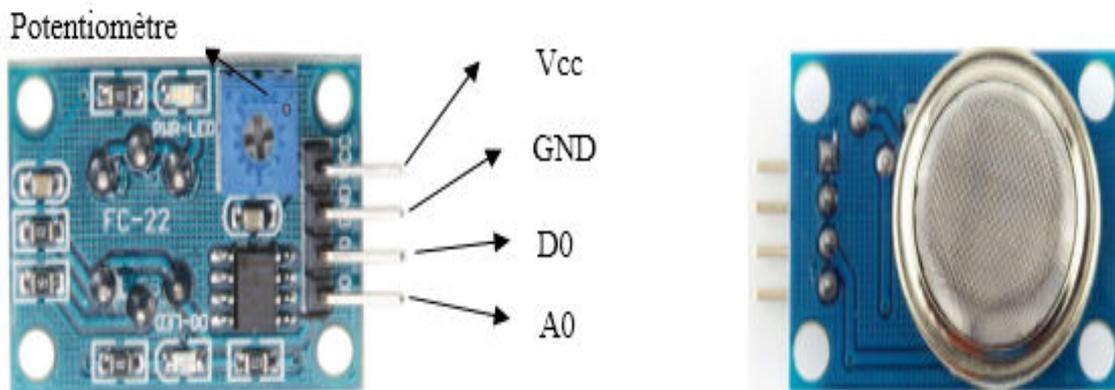


FIGURE II.5 – Capteur MQ-2.

### II.2.2.3 Capteur HC-SR501

Le HC-SR501 (Figure II.6) est un capteur passif d'infrarouge (capteur PIR) qui détecte le rayonnement infrarouge également désigné par rayonnement thermique émis par des corps présents dans son champ de vision, ce qui en fait un capteur idéal pour détecter un mouvement ou une présence. Ce capteur renvoie un signal qui possède deux états : un état bas (0V) lorsque le capteur est au repos, et un état haut (3.3V) si un mouvement est détecté. Il a la possibilité d'être paramétré par son utilisateur. Pour cela, il dispose de deux potentiomètres pour régler la distance de détection qui sera comprise entre 3 et 7m et la durée de l'état haut qui sera comprise entre 3 et 300s. Le HC-SR501 s'alimente avec une tension comprise entre 5 et 20V et il fonctionne à des températures de -15 à 70°C.

Il faut noter que ce capteur prend une minute d'initialisation pendant laquelle il va renvoyer un signal à l'état haut (3.3V) 0 à 3 fois, puis il va passer en état de veille. Comme il faut aussi éviter le vent, les lumières et d'autres sources d'interférence à proximité directe du capteur [W5].

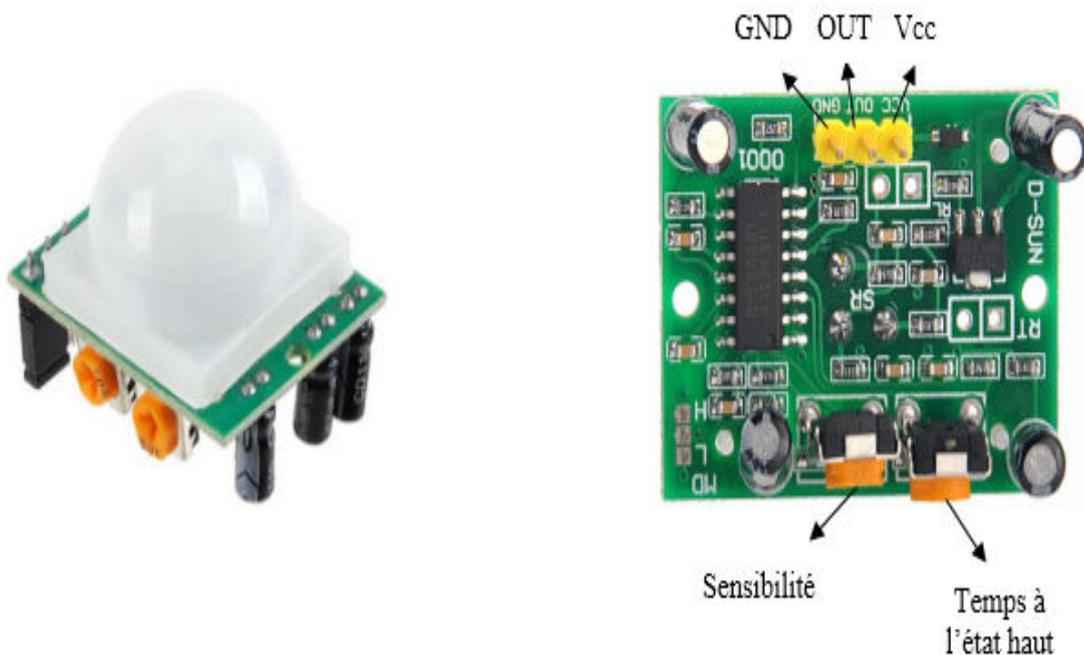


FIGURE II.6 – Capteur HC-SR501.

### II.2.3 Le module ESP-01

L'ESP-01 (Figure II.7) est un petit module qui permet de connecter n'importe quel microcontrôleur du moment qu'il possède un émetteur-récepteur asynchrone universel (UART) à un réseau Wi-Fi, et d'établir des connexions TCP/IP avec des commandes AT. Plus précisément ce module est un système-sur-puce (SoC) qui fonctionne dans les bandes de fréquences de 2.4 GHz, il contient un microcontrôleur interne qui donne l'accès sans fil aux réseaux, avec d'autres options plus développées.

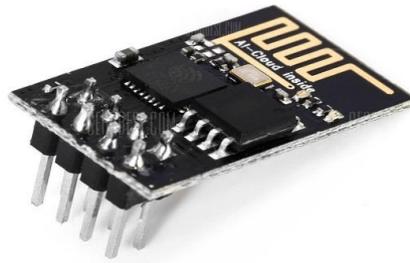


FIGURE II.7 – Module ESP-01.

### II.2.3.1 Caractéristiques

<b>Microcontrôleur</b>	Tensilica L106
<b>Wi-Fi</b>	802.11 b/g/n
<b>Modes de sécurité Wi-Fi</b>	WPA, WPA2
<b>Type d'antenne</b>	Antenne PCB
<b>Tension de fonctionnement</b>	3-3.6V
<b>Tension d'alimentation</b>	3-3.6V (ne tolère pas 5V)
<b>Broches entrées /sorties</b>	8
<b>Mémoire FLASH</b>	1 MB
<b>Fréquence d'horloge</b>	80 MHz
<b>Longueur</b>	14.3 mm
<b>Largeur</b>	24.8 mm

TABLE II.6 – Les caractéristiques de l'ESP-01.

### II.2.3.2 Broches d'entrée/sortie

Le module ESP-01 a huit broches (Figure II.8), chacune de ces broches est décrite dans le tableau ci-dessous.

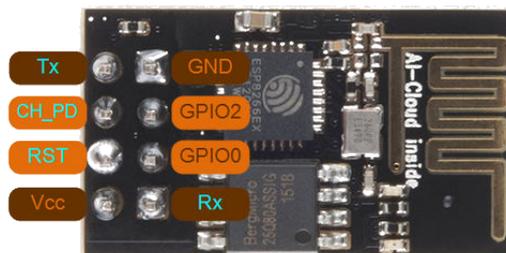


FIGURE II.8 – Les huit broches de l'ESP-01.

1	<b>TX</b>	Donnée série, transmettre
2	<b>GND</b>	La masse (0V)
3	<b>CH_PD</b>	doit être mise à Vcc (3.3V) pour activer le mode fonctionnement du module
4	<b>GPIO 2</b>	Entrée/sortie numérique
5	<b>RST</b>	Placer à la masse pour réinitialiser
6	<b>GPIO 0</b>	Entrée/sortie numérique
7	<b>VCC</b>	Alimentation 3-3.6V
8	<b>RX</b>	Donnée série, recevoir

TABLE II.7 – Description des différentes broches de l’ESP-01.

### II.2.3.3 Les commandes AT

Une fois le module ESP-01 est relié avec un microcontrôleur externe, la communication se fait par l’intermédiaire des commandes AT pour effectuer des opérations d’une façon automatique (accéder à un réseau Wi-Fi, vérifier l’adresse IP du module, quitter le réseau, etc.). Une commande AT commence toujours par "AT+" suivie de la commande à envoyer, en retour le module renvoie "OK" pour indiquer le bon déroulement de la commande.

### II.2.4 Point d’accès

Pour connecter l’ESP-01 et la WeMos D1 Mini à l’Internet, on a besoin d’un point d’accès sans fil (Wi-Fi) qui permet de relayer l’information depuis et vers l’Internet.

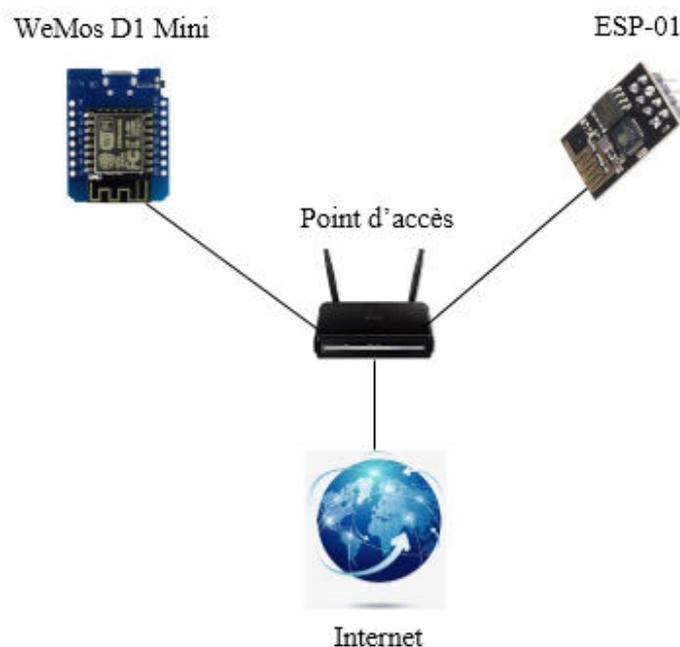


FIGURE II.9 – Connexion de la WeMos D1 Mini et l’ESP-01 à l’Internet via le point d’accès.

## II.3 Étude de la partie logicielle

Cette partie est consacrée à la présentation des plateformes informatiques utilisées dans notre projet.

### II.3.1 L'environnement de développement Arduino

Le logiciel de programmation des cartes Arduino est une application Java, libre et multiplateforme, servant d'éditeur de code et de compilateur qui peut transférer le programme (croquis ou sketch) au travers de la liaison série asynchrone (RS-232, Bluetooth ou USB selon la carte). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande [B11][B12]. Le langage de programmation Arduino est basé sur les langages C/C++ adaptés à la carte et ses entrées/sorties.

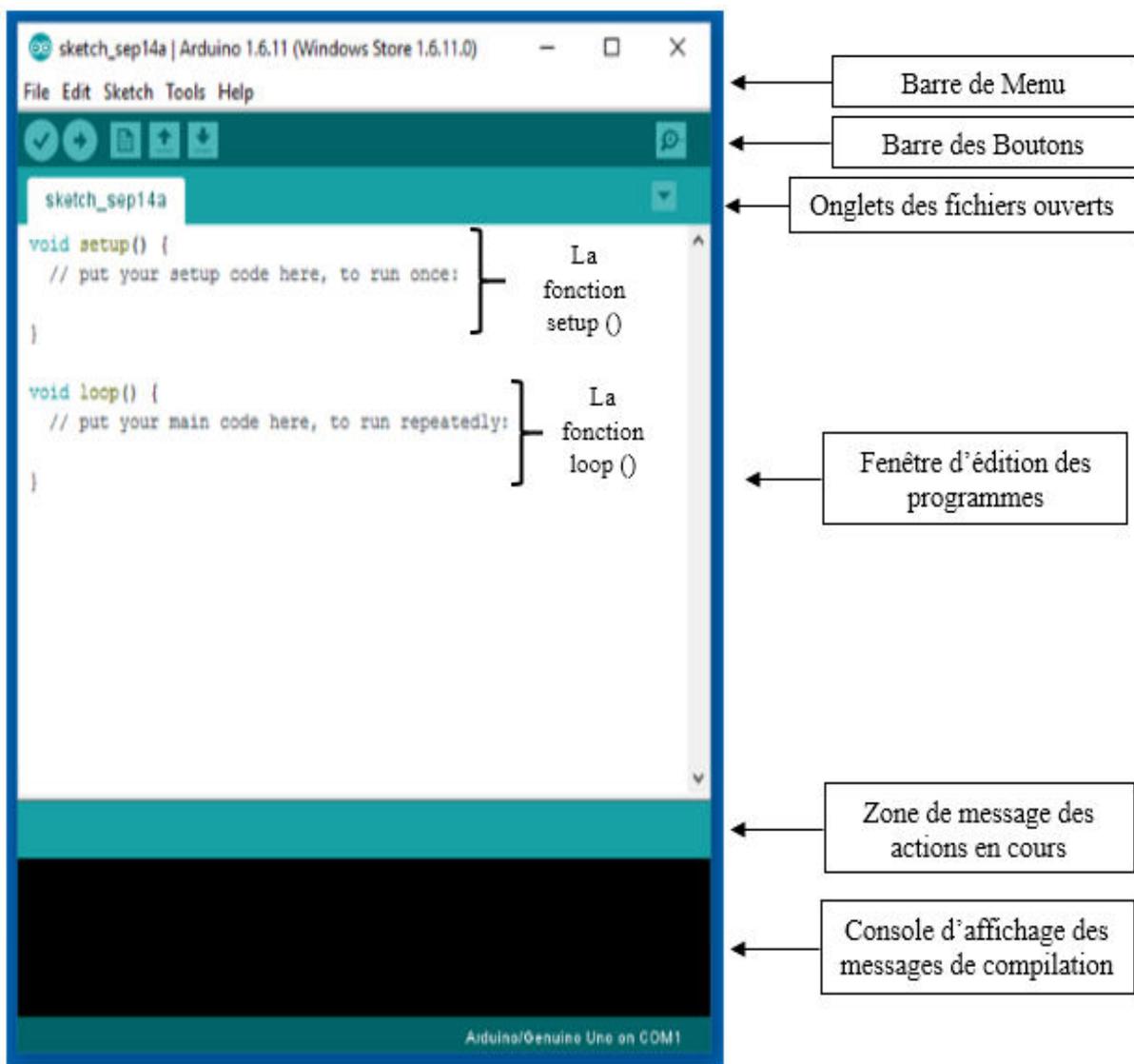


FIGURE II.10 – L'interface du logiciel Arduino.

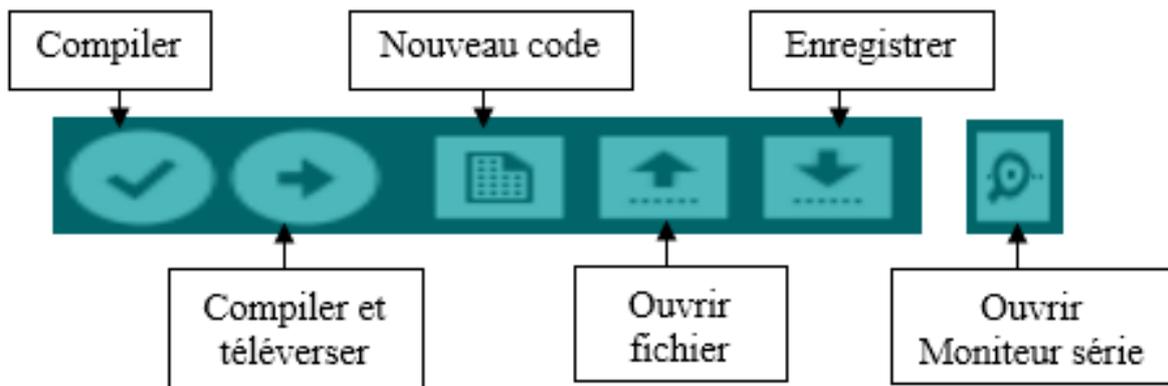


FIGURE II.11 – La barre des boutons du logiciel Arduino.

### II.3.1.1 Structure d'un programme en Arduino

Un programme en Arduino doit être composé au minimum de deux fonctions obligatoires :

- une fonction `setup()` qui est exécutée une seule fois, elle initialise et fixe les valeurs du démarrage du programme.
- une fonction `loop()` qui est exécutée en boucle sans fin une fois que la fonction `setup()` a été exécutée une fois.

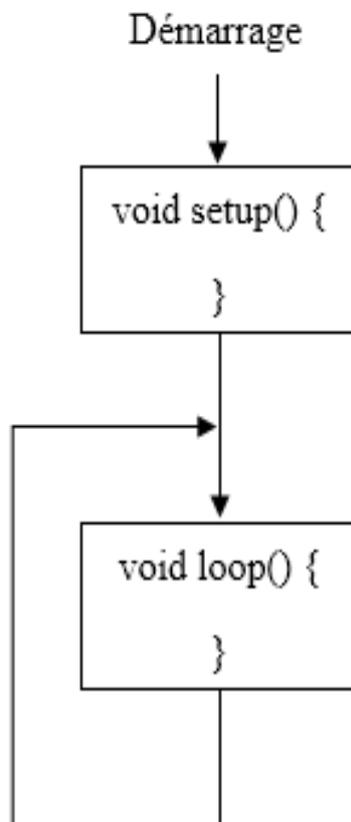


FIGURE II.12 – L'organigramme d'exécution d'un programme en Arduino.

## II.3.2 Application Web

Une application Web est une application accessible via un navigateur Web. Contrairement à d'autres applications, une application Web est basée sur la technologie client-serveur, elle est mise en place sur un serveur et immédiatement utilisable par les clients sans procédure d'achat et d'installation sur leur propre ordinateur ou Smartphone. Il suffit de se connecter à l'application à l'aide de leurs navigateurs favoris (Google Chrome, Mozilla Firefox, Internet Explorer, etc.) pour pouvoir y accéder.

Le développement des applications Web nécessite la maîtrise parfaite des différents langages de codage comme le HTML, CSS, JavaScript côté client et le PHP, SQL côté serveur. En résumé, les applications Web sont des pages vivantes qui réagissent à vos sollicitations et vous obéissent « au doigt et à l'œil », cela nécessite donc deux critères d'utilisation : simplicité et efficacité.

## II.3.3 Cloud Computing

### II.3.3.1 Définition

Le Cloud Computing selon [W6], est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée. L'ordinateur de bureau ou portable, le téléphone mobile, la tablette tactile et autres objets connectés deviennent des points d'accès pour exécuter des applications ou consulter des données qui sont hébergées sur les serveurs.

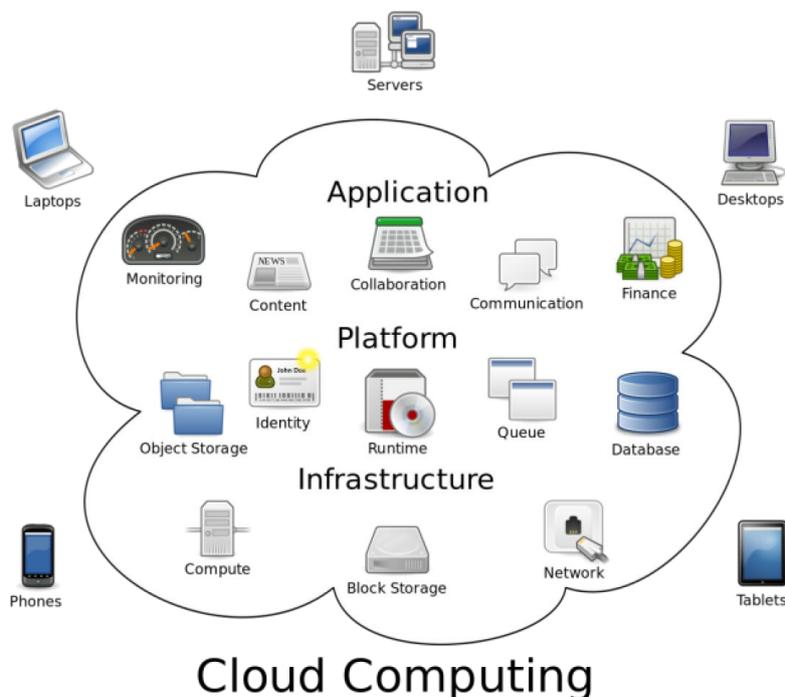


FIGURE II.13 – Schéma donnant un aperçu sur les facteurs principaux du Cloud Computing.

Pour le grand public, le Cloud Computing se matérialise notamment par les services de stockage et de partage de données numériques type Box, Dropbox, Microsoft OneDrive ou Apple iCloud sur lesquels les utilisateurs peuvent stocker des contenus personnels (photos, vidéos, musiques, documents, etc.) et y accéder n'importe où dans le monde depuis n'importe quel terminal connecté.

### II.3.3.2 Caractéristiques

Généralement, un service, une solution ou un environnement d'exécution devrait satisfaire une liste de caractéristiques pour qu'il soit considéré comme étant du Cloud Computing. Parmi ces caractéristiques, il y a celles qui sont reconnues comme fondamentales [B13]. Par exemple, NIST définit cinq caractéristiques essentielles qui sont :

- **Ressources à la demande**

Un utilisateur peut allouer unilatéralement des ressources informatiques (serveurs, réseau, stockage, environnement d'exécution, application, etc.) au besoin, de façon automatique et sans nécessité d'interaction humaine avec chaque fournisseur de services.

- **Large accès réseau**

Les ressources Cloud Computing sont disponibles à travers le réseau et accessibles via des mécanismes standards qui favorisent leurs utilisations à partir des appareils clients hétérogènes, voire légers (ordinateurs portables, téléphones, tablettes, etc.).

- **Mutualisation des ressources**

Les ressources informatiques du fournisseur Cloud Computing sont mutualisées pour servir plusieurs clients en utilisant un modèle multi-tenant. Ces ressources, physiques ou virtuelles, sont allouées et libérées dynamiquement selon la demande du consommateur. Généralement, l'utilisateur n'a ni le contrôle ni la connaissance de l'emplacement exact des ressources allouées. Dans certains cas, il peut choisir l'emplacement géographique à un niveau haut (par pays, continent ou data-center).

- **Élasticité rapide**

Les ressources sont allouées et libérées d'une façon élastique, idéalement d'une façon automatique, pour s'adapter rapidement à la demande qu'elle soit croissante ou décroissante. Pour le consommateur, les ressources disponibles à l'allocation apparaissent comme illimitées et peuvent s'allouer à tout moment.

- **Services mesurés**

Toutes les ressources allouées peuvent être surveillées et contrôlées afin de mesurer leurs consommations avec un niveau d'abstraction approprié selon le type du service (stockage, temps de calcul, bande passante, etc.).

En plus des cinq caractéristiques définies par NIST, il y a d'autres caractéristiques dont nous citons les plus pertinentes :

- **Haute évolutivité**

Les services de type Cloud Computing devraient être évolutifs et doivent satisfaire toute demande de croissance de la part des utilisateurs selon le besoin des services et des ressources allouées. Cette évolutivité doit se faire d'une façon automatique et en cours d'exécution.

- **Haute disponibilité**

La haute disponibilité des ressources Cloud Computing est primordiale. Par exemple, l'accès réseau aux ressources doit être assuré à plein temps sans aucune interruption.

- **Haute fiabilité**

Toutes les ressources Cloud Computing doivent être d'une grande fiabilité. Par exemple, la probabilité de perte de donnée doit être quasi nulle.

- **Paiement à l'usage**

La philosophie de facturation des ressources Cloud Computing se base sur l'usage. L'idée est que l'utilisateur ne paye que ce qui a consommé.

- **Efficacité énergétique**

Malgré que cette caractéristique n'a pas été unanimement adoptée, mais la diminution de l'énergie consommée par les ressources Cloud Computing est d'une grande importance et sa prise en considération ne cesse de croître. En effet, l'efficacité énergétique contribue à diminuer les coûts.

- **Simplicité d'utilisation**

L'allocation, la gestion et l'utilisation des ressources Cloud Computing doivent être simples. Idéalement, elles doivent se faire à travers des interfaces (APIs) efficaces et génériques.

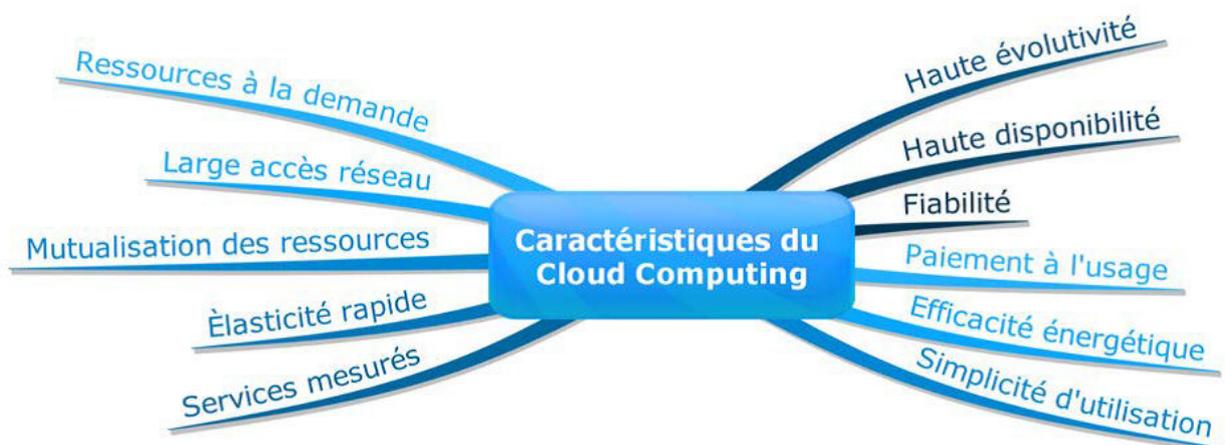


FIGURE II.14 – Caractéristiques du Cloud Computing.

Dans notre projet, on va utiliser le ThingSpeak Cloud, c'est une plateforme IoT qui permet de collecter et stocker les données des objets connectés en passant par le protocole HTTP via l'Internet ou via un réseau local.

Avec ThingSpeak, on peut créer des visualisations instantanées de données en direct, agir sur les données et envoyer des alertes à l'aide de services Web tels que Twitter, Twilio [B14]. Il offre la possibilité aux ingénieurs et aux scientifiques de prototyper et de construire des systèmes IoT sans installer des serveurs ni développer des logiciels Web.

Pour l'implémentation des notifications dans notre projet, on va se servir de Firebase Cloud Messaging. Ce dernier permet de notifier une application cliente qu'un nouvel e-mail ou d'autres données sont disponibles pour la synchronisation et d'envoyer des messages de notifications pour stimuler le réengagement et la rétention des utilisateurs [W7].

Chapitre **III**

## ÉTUDE DES PROTOCOLES DE COMMUNICATION

---

L'objectif de ce chapitre est de donner une brève explication des protocoles de communication choisis dans notre projet et de révéler l'intérêt de ce choix.

---

## III.1 Protocoles de communication utilisés

Notre projet est basé sur plusieurs protocoles de communication. Dans cette partie, on va décrire ces protocoles et citer les raisons majeures qui nous ont poussé de les utiliser.

### III.1.1 Wi-Fi

#### III.1.1.1 Définition

Le terme Wi-Fi désigne un ensemble de protocoles de communication sans fil dont le standard technique a été normalisé par le groupe IEEE 802.11. C'est une technologie de réseau informatique mise en place pour fonctionner en réseau interne et, depuis devenu un moyen d'accès haut débit à l'Internet. Le Wi-Fi permet de relier par ondes radio dans une bande de fréquence de 2.4 ou 5 GHz des équipements informatiques (ordinateur, Smartphone, tablette, point d'accès, etc.) à une liaison haut débit (de 11 Mbit/s en 802.11b à 54 Mbits/s en 802.11a/g) sur un rayon de plusieurs dizaines de mètres en intérieur. Dans un environnement ouvert, la portée peut atteindre plusieurs centaines de mètres.

#### III.1.1.2 Les modes d'opération

Le Wi-Fi peut fonctionner en deux modes : l'un appelé mode infrastructure (BSS) et l'autre mode ad hoc ou sans infrastructure (IBSS). Les deux modes sont présentés ci-dessous.

##### III.1.1.2.a Le mode ad hoc

Le mode ad hoc permet à plusieurs stations de communiquer directement entre elles sans avoir besoin d'un point d'accès (AP) central. En fonctionnant en mode ad hoc, les stations étant à portée les unes des autres peuvent se découvrir et communiquer directement, à condition qu'elles utilisent le même canal et le même SSID. Il n'y a pas de limite dans le nombre de stations qui peuvent participer au réseau ad hoc autre que les limites techniques qui font que le canal peut rapidement être saturé lorsque le nombre de stations augmente. Chaque station participant au réseau ad hoc est donc une partie de l'infrastructure portant le réseau.

Cependant, le mode ad hoc seul ne décrit pas les mécanismes de routage nécessaires au cas où deux stations appartenant au même réseau ad hoc ne sont pas à portée l'une de l'autre. Par exemple, sur la figure III.1, les stations 1 et 4 ne peuvent pas communiquer directement et doivent utiliser le relais 3 comme une station intermédiaire [B15].

##### III.1.1.2.b Le mode infrastructure

Dans le mode infrastructure, chaque station est reliée au réseau via un point d'accès (AP). On dit que la station est le client et l'AP le maître. Un réseau de ce type couvre un espace qu'on appelle une cellule. L'AP sert de relais entre les stations, mais il peut aussi servir de relais vers un réseau filaire, par exemple votre réseau d'entreprise [B16].

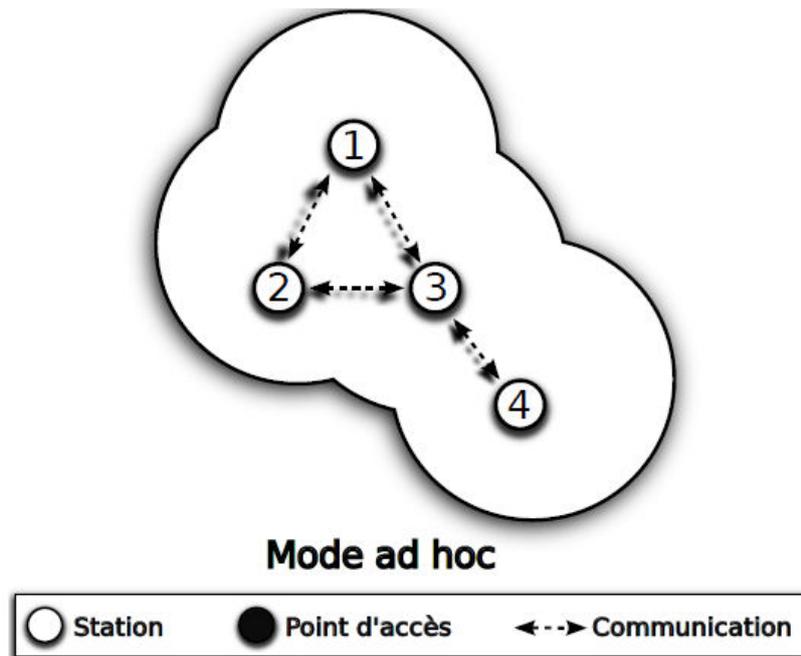


FIGURE III.1 – Mode de fonctionnement ad hoc du Wi-Fi.

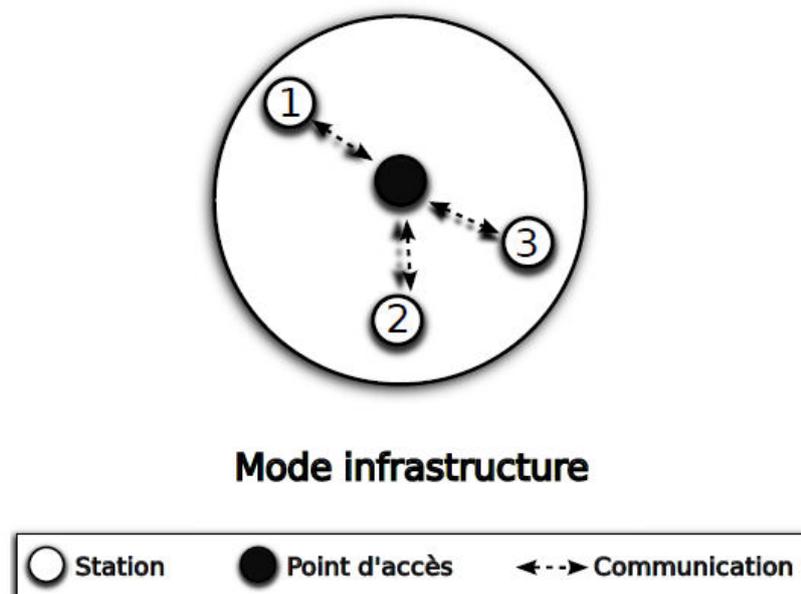


FIGURE III.2 – Mode de fonctionnement infrastructure du Wi-Fi.

### III.1.1.3 Pourquoi le Wi-Fi ?

Dans notre projet, les nœuds sont alimentés de façon permanente, donc pas de contrainte énergétique. De ce fait, on peut éviter l'utilisation d'autres protocoles à faible énergie comme le Zigbee pour connecter les nœuds à l'Internet (scénario 1) et d'utiliser directement le Wi-Fi (scénario 2) et bénéficier de ses points forts en termes de débit et de portée, en plus, le coût des modules Wi-Fi est réduit par rapport aux modules (Xbee) qui travaillent avec le ZigBee.

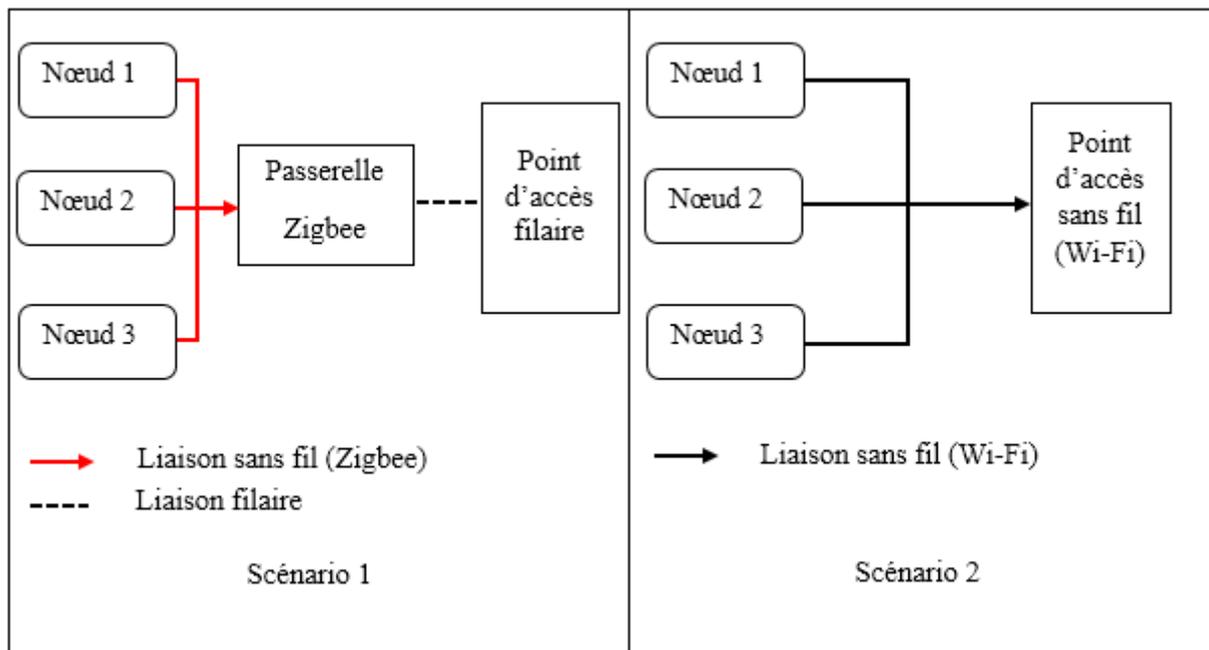


FIGURE III.3 – Schéma représentatif des deux scénarios.

## III.1.2 HTTP

### III.1.2.1 Définition

Le protocole de transfert hypertexte, plus connu sous l'abréviation HTTP est un protocole de communication client-serveur développé pour le World Wild Web, classé parmi les protocoles de haut niveau (couche application du modèle OSI) qui fonctionne sur le principe requête-réponse. Il est suffisamment léger et rapide pour la transmission des données localisées grâce à une chaîne de caractères appelée URL entre le client et le serveur.

Le HTTPS est une variante du HTTP, sécurisée par des protocoles SSL ou TLS qui est notamment utilisé pour le paiement sécurisé et les transactions financières en ligne.

### III.1.2.2 La requête HTTP

Une requête HTTP est un ensemble de lignes envoyées au serveur par le client [B17]. Elle comprend :

- 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 utilisé. La ligne comprend trois éléments devant être séparés par un espace :
  - La méthode.
  - L'URL.
  - La version du protocole utilisé par le client (généralement HTTP/1.0).
- 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.

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.

- Le corps de la requête : c'est un ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide permettant par exemple un envoi de données par une méthode POST lors de l'envoi de données au serveur par un formulaire.

Une requête HTTP a donc la syntaxe suivante (<crLf> signifie retour chariot) :

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

HTTP définit un ensemble de méthodes de requête qui indiquent l'action que l'on souhaite réaliser sur la ressource indiquée, les plus courantes sont GET, POST, HEAD, DELETE :

Méthode	Fonction
GET	Demander une ressource
POST	Envoyer des données en vue d'un traitement à une ressource
HEAD	Demander des informations sur la ressource, sans demander la ressource elle même
DELETE	Supprimer une ressource

TABLE III.1 – Des méthodes HTTP et leurs fonctions.

Voici donc un exemple de requête HTTP :

```
GET /page.html HTTP/1.0
Host : exemple.com
Referer : http ://exemple.com/
User-Agent : CERN-LineMode/2.15 libwww/2.17b3
```

### III.1.2.3 La réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées par le serveur au client [B17]. Elle contient :

- Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état de 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.
- Le corps de la réponse : il contient la ressource demandée, celle-ci est en fonction de la méthode utilisée dans la requête.

Une réponse HTTP a donc la syntaxe suivante (<crLf> signifie retour chariot) :

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

Les codes de statut de la réponse HTTP indiquent si une requête HTTP a été exécutée avec succès ou non. Les plus utilisés sont :

Code de statut	Signification
200	La requête a réussi
401	Une authentification est nécessaire pour obtenir la réponse
403	Accès refusé
404	Ressource introuvable par le serveur

TABLE III.2 – Signification des codes de statut des réponses HTTP.

Voici donc un exemple de réponse HTTP :

```
HTTP/1.0 200 OK
Date : Fri, 31 May 2018 23 :59 :59 GMT
Server : Apache/2.3
Content-Type : text/html
Content-Length : 59

<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

### III.1.2.4 Pourquoi le HTTP ?

Le protocole HTTP permet aux nœuds d'écrire les données collectées sur le ThingSpeak Cloud et à l'application de lire ces données. Il existe également d'autres protocoles qui font la même tâche comme le MQTT [B18] et le CoAP, mais en raison de l'utilisation d'une application Web, le HTTP est le mieux adapté. En outre, le Firebase Cloud Messaging se base sur le protocole HTTP pour l'échange des messages de notifications.

# Chapitre **IV**

## RÉALISATION DU PROJET

---

Dans ce chapitre, on va présenter les étapes de réalisation du projet en commençant par les nœuds et ses schémas électriques puis, l'application Web et ses interfaces tout en exposant les organigrammes de fonctionnement.

---

## IV.1 La phase de réalisation du prototype

### IV.1.1 Les nœuds communicants

Les nœuds communicants constituent la grande partie matérielle de notre prototype. On peut distinguer deux types de nœuds, un nœud qui a comme fonction d'acquérir des informations sur l'état physique (température, humidité), et les nœuds détecteurs : détecteur de mouvement, détecteur de gaz/fumée. L'ensemble poste les informations vers ThingSpeak Cloud, de plus les nœuds détecteurs utilisent le Firebase Cloud Messaging pour délivrer les notifications. L'application Web s'occupe d'afficher ces informations et déclencher la notification suite à la détection de mouvement ou de gaz/fumée.

Dans ce qui suit, on va détailler la composition de chaque nœud, donner son schéma électrique ainsi que l'organigramme qui décrit le principe de son fonctionnement.

#### IV.1.1.1 Le nœud 1

##### IV.1.1.1.a Branchement du l'ESP-01 et du capteur DHT11

l'ESP-01 (voir chapitre II) dispose de 8 broches dont 5 broches sont utilisées pour la liaison avec l'Arduino Uno :

- Vcc : l'alimentation 3.3V du module.
- GND : la masse.
- TX : connectée à la broche 2 de l'Arduino Uno.
- RX : connectée à la broche 3 de l'Arduino Uno.
- CH\_PD : doit être mise à Vcc (3.3V) pour activer le mode fonctionnement du module.

Pour le capteur DHT11, le branchement est le suivant :

- Vcc : l'alimentation 5V du capteur.
- GND : la masse.
- S : connectée à la broche A0 de l'Arduino Uno.

La figure IV.1 représente le circuit électrique du branchement, conçu par le logiciel de conception des circuits imprimés Fritzing.

Après avoir validé le schéma électrique, on va basculer vers l'interface du logiciel Fritzing utilisé pour la conception du PCB. Le circuit imprimé final utilisé pour le shield du l'ESP-01 et le capteur DHT11 est illustré à la figure IV.2.

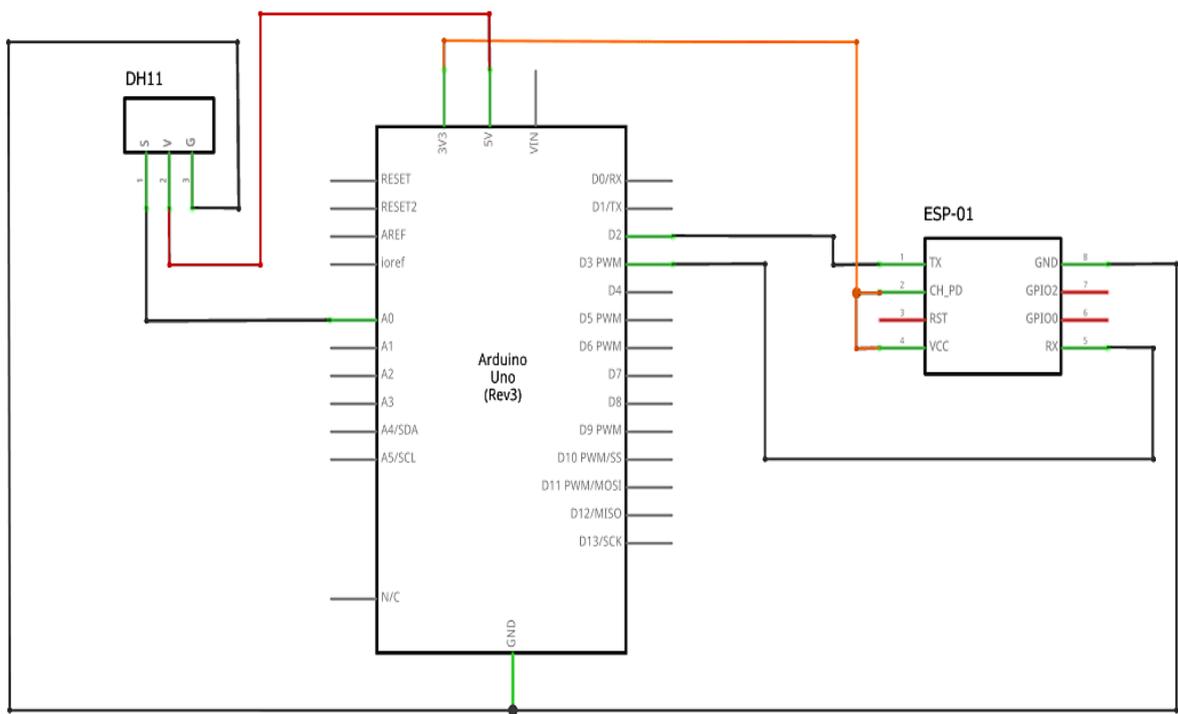


FIGURE IV.1 – Schéma électrique du branchement du capteur DHT11 et l'ESP-01 avec l'Uno.

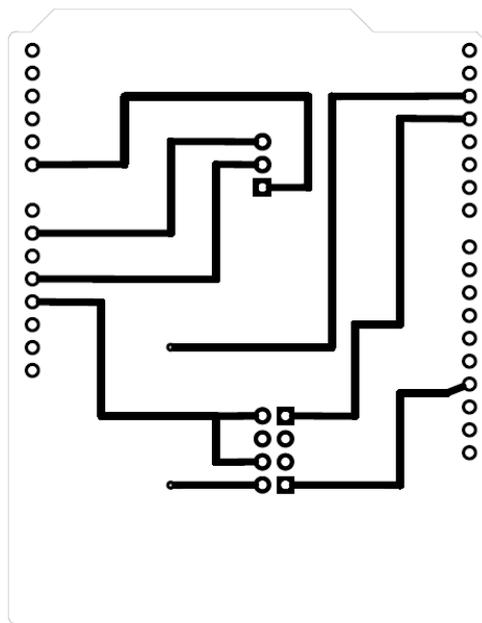


FIGURE IV.2 – PCB du shield de l'ESP-01 et le capteur DHT11.

#### IV.1.1.1.b Algorithme d'acquisition de la température et d'humidité

L'acquisition de la température et de l'humidité est réalisée via le capteur DHT11, dont l'algorithme de fonctionnement est donné par l'organigramme ci-dessous. Les valeurs mesurées seront par la suite affichées par l'application Web.

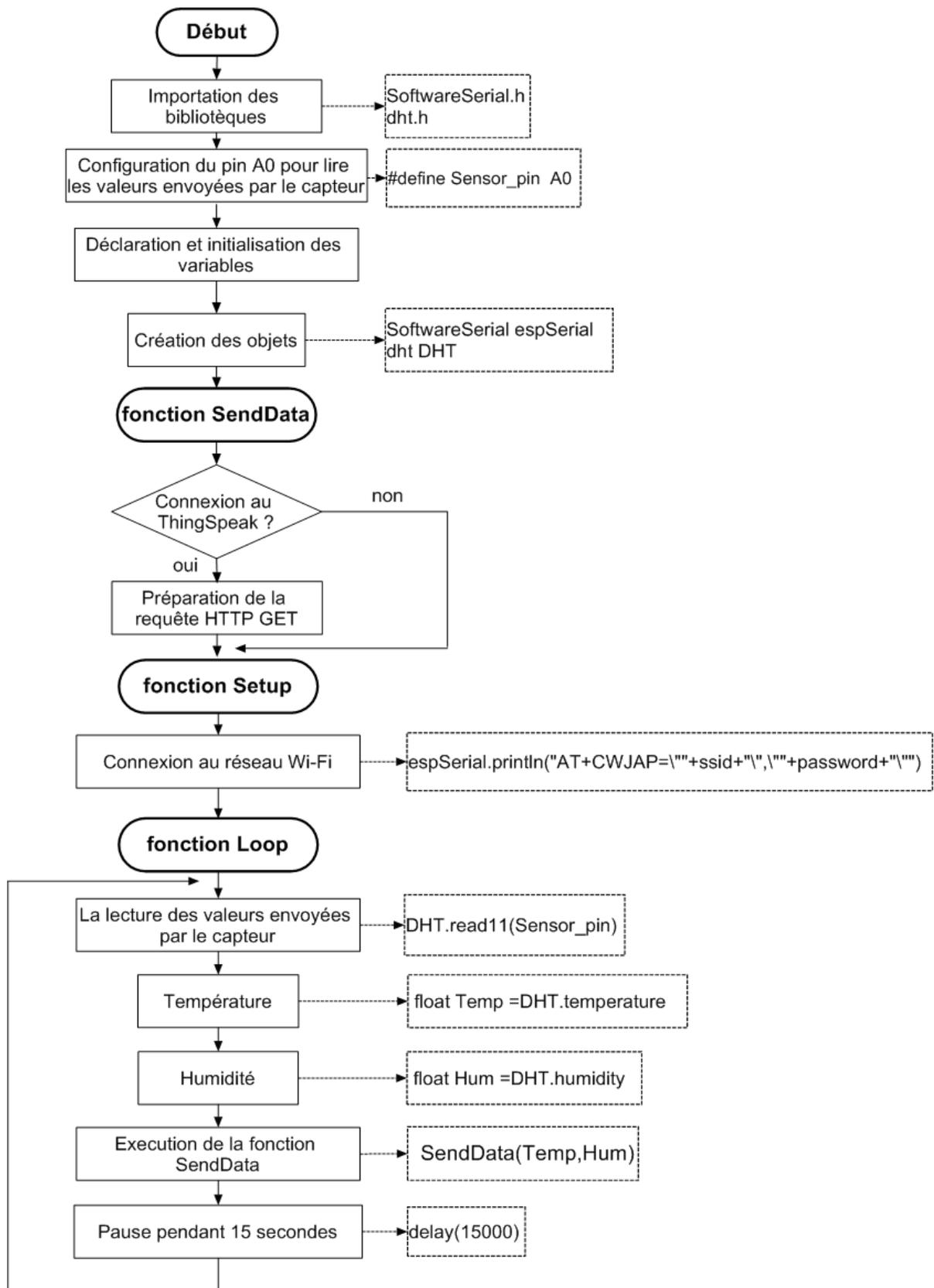


FIGURE IV.3 – Organigramme d’acquisition de la température et d’humidité.

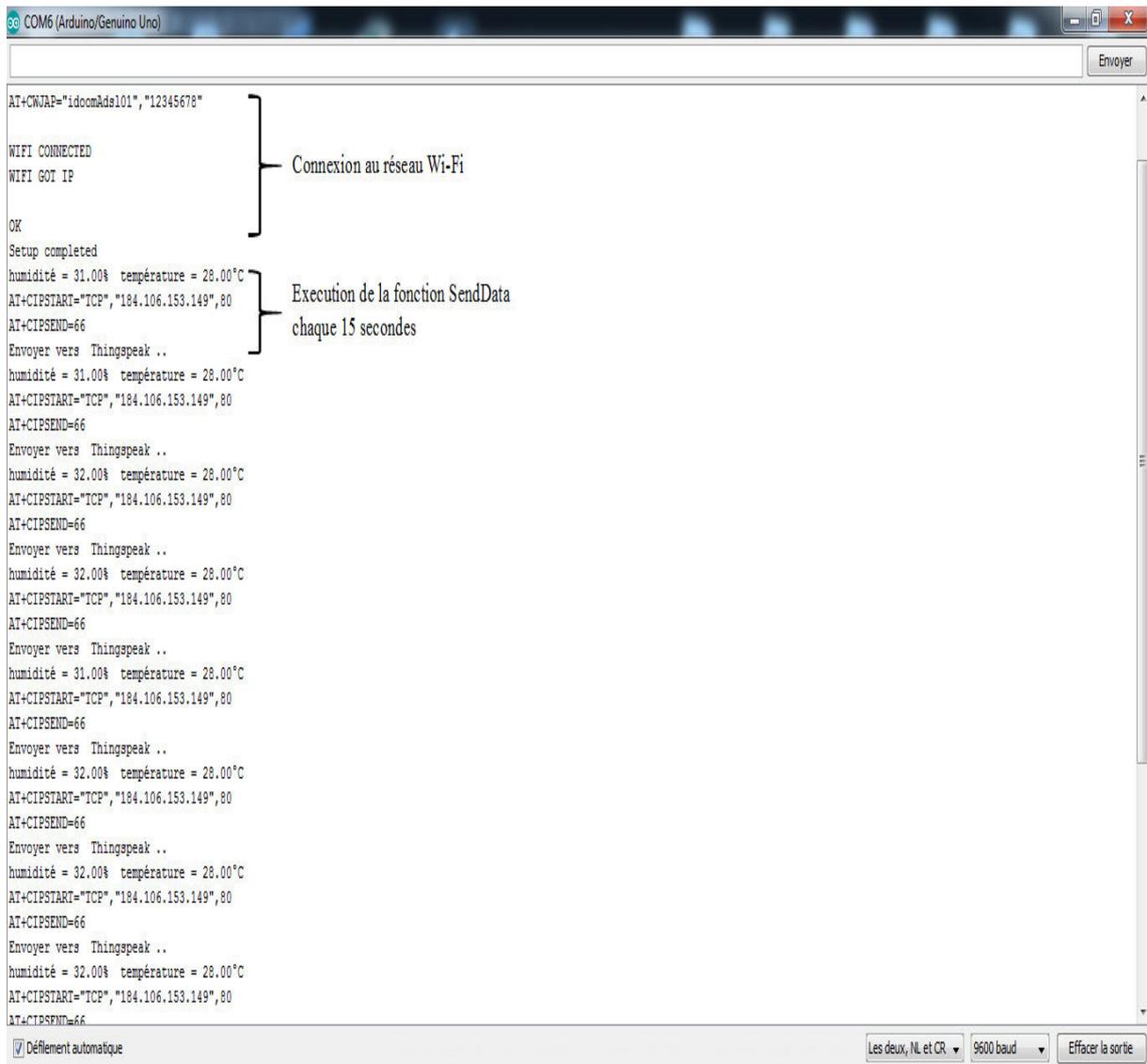


FIGURE IV.4 – Capture du moniteur série.

### IV.1.1.2 Le nœud 2

#### IV.1.1.2.a Branchement du capteur HC-SR501

Le branchement du capteur HC-SR501 est très simple, il suffit en effet de relier l'alimentation 5V et on aura une lecture numérique de la WeMos D1 Mini à la broche OUT du capteur, comme mentionné ci-dessous :

- Vcc : l'alimentation 5V du capteur.
- GND : la masse.
- OUT : connectée à la broche D0 de la WeMos D1 Mini.

Le schéma électrique du branchement est présenté ci-après :

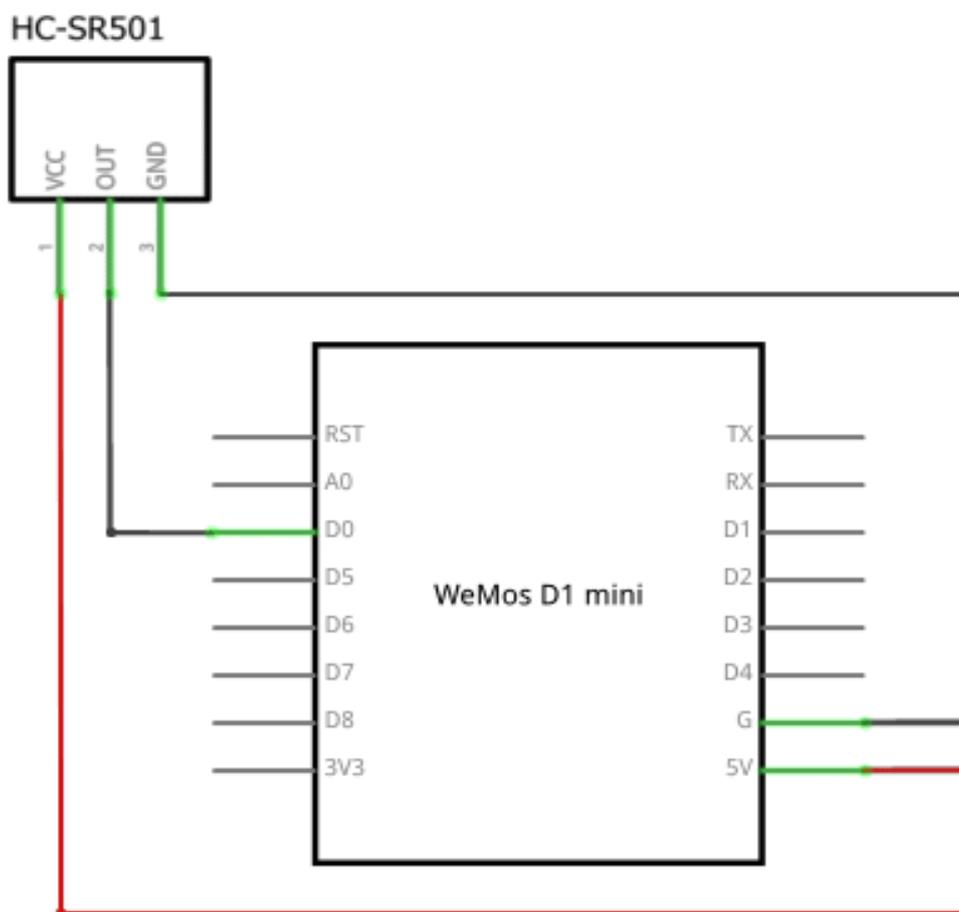


FIGURE IV.5 – Schéma électrique du branchement de la D1 Mini avec le capteur HC-SR501.

#### IV.1.1.2.b Algorithme de détection de mouvement

Le rôle principal de cet algorithme est d'informer l'utilisateur en temps réel lorsqu'un mouvement se produit. Ceci est assuré par le capteur HC-SR501. L'organigramme proposé pour cet algorithme est présenté dans la figure suivante :

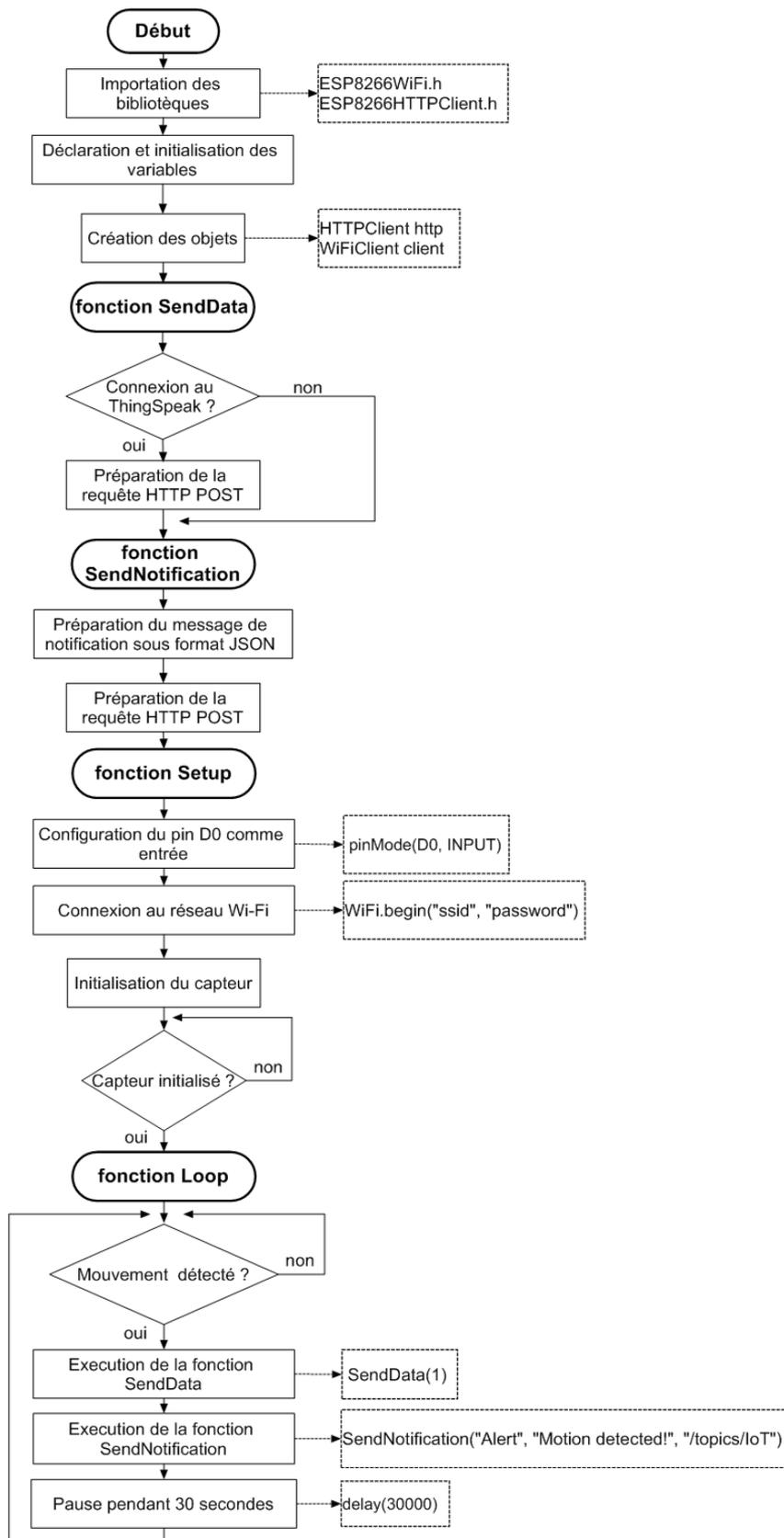


FIGURE IV.6 – Organigramme de détection de mouvement.

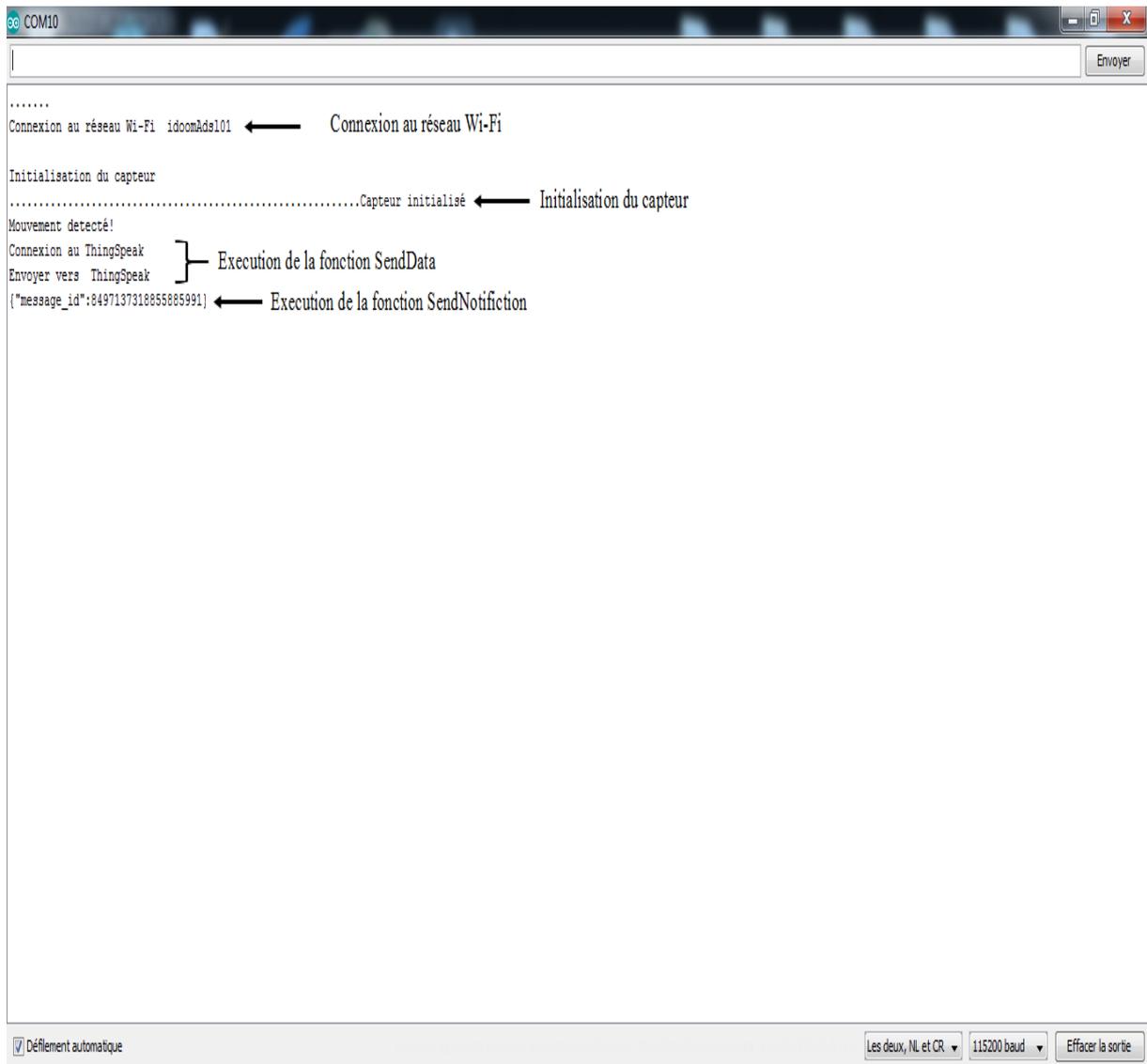


FIGURE IV.7 – Capture du moniteur série.

**IV.1.1.3 Le nœud 3**

**IV.1.1.3.a Branchement du capteur MQ-2**

Le branchement du capteur MQ-2 consiste à relier les broches Vcc et GND du capteur à l'alimentation 5V et à la masse de la WeMos D1 Mini. La sortie analogique du capteur A0 ne doit pas être connectée directement à la broche A0 de la WeMos D1 Mini, car cette dernière ne tolère pas au delà de 3.2V comme tension d'entrée pour la broche A0. Pour cela, on est obligé d'utiliser un diviseur de tension qui permet de diminuer la tension d'entrée 5V vers une tension de sortie de 3.2V.

Le circuit électrique du branchement se présente dans la figure suivante :

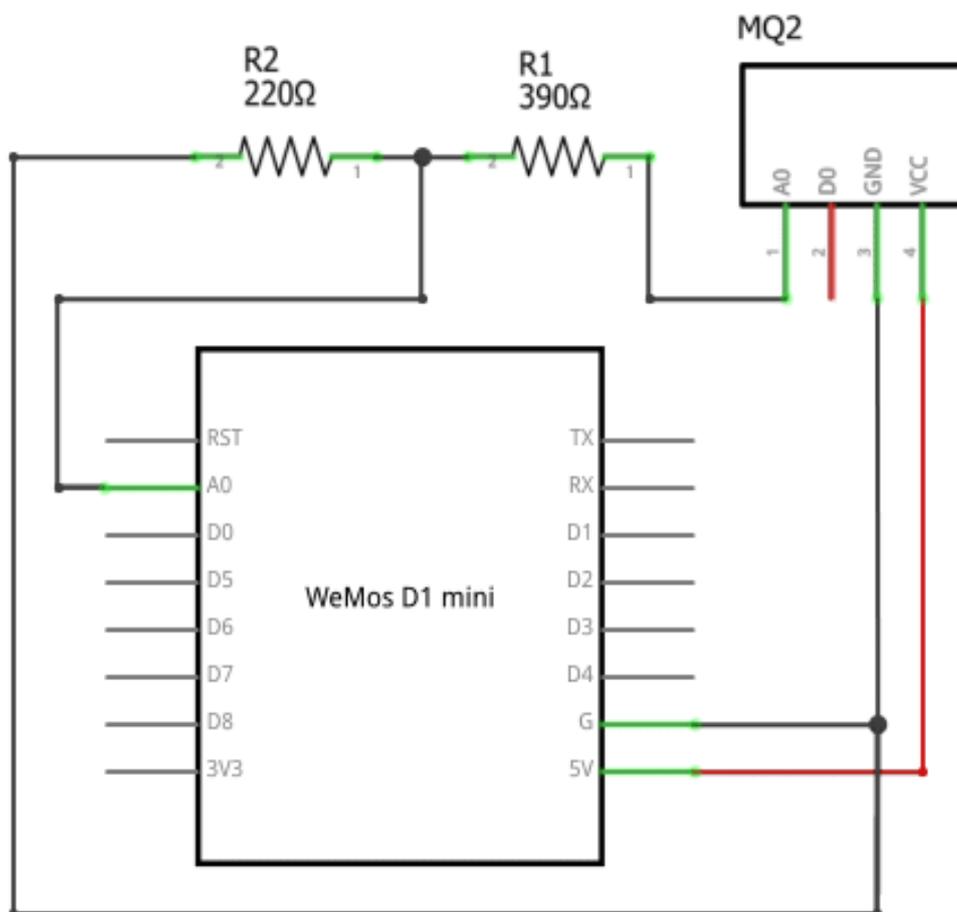


FIGURE IV.8 – Schéma électrique du branchement de la D1 Mini avec le capteur MQ2.

**IV.1.1.3.b Algorithme de détection de gaz/fumée**

Cet algorithme permet de détecter un gaz combustible ou une fumée à l'aide du capteur MQ2, dont le but est de prévenir l'utilisateur en cas de danger. Le fonctionnement de cet algorithme est décrit par l'organigramme de la figure IV.9.

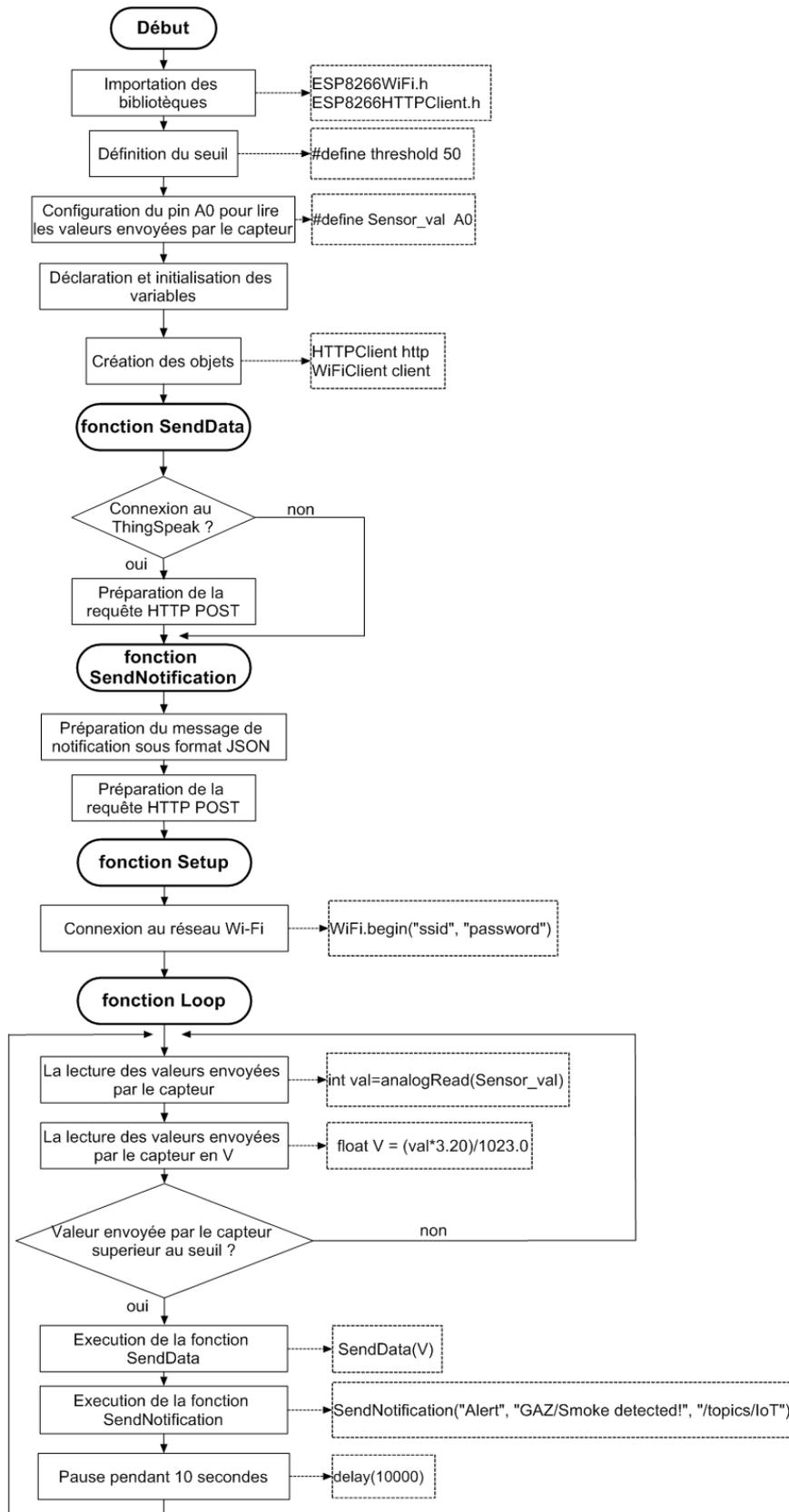


FIGURE IV.9 – Organigramme de détection de gaz/fumée.

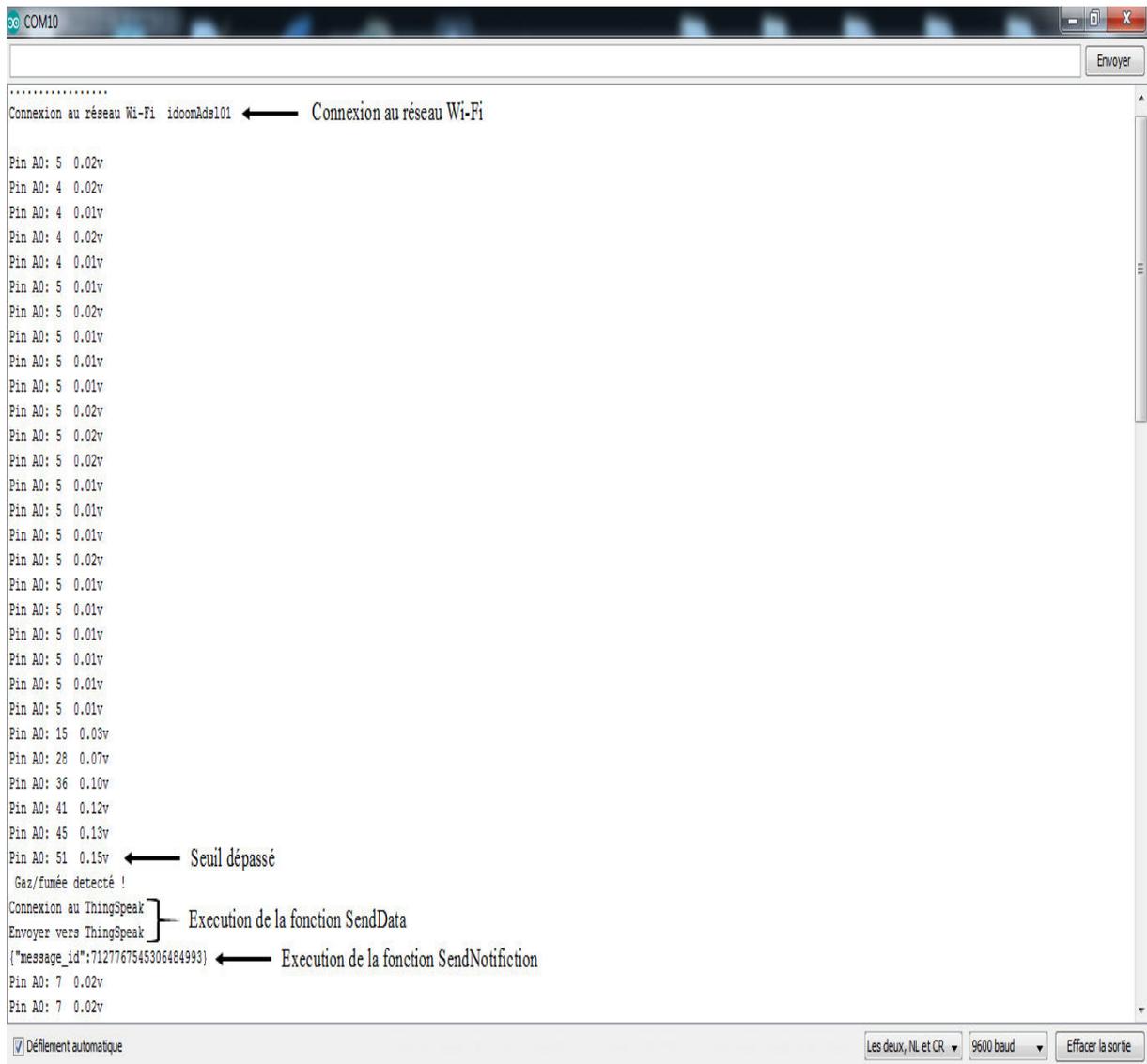


FIGURE IV.10 – Capture du moniteur série.

### IV.1.2 Application Web

L'application Web permet de visualiser les informations collectées par le premier nœud (température, humidité) et d'alerter l'utilisateur par une notification lors de la détection d'un mouvement par le deuxième nœud ou d'un gaz/fumée par le troisième nœud. Elle est basée sur le responsive Web design ce qui donne la possibilité à l'utilisateur de bénéficier de l'application quelle que soit sa gamme d'appareils (téléphones mobiles, tablettes, liseuses, moniteurs d'ordinateur de bureau).

Plusieurs langages de programmation sont utilisés pour le développement de notre application. Ils sont présentés ci-dessous :

- HTML5 : pour la création et la structuration.
- CSS3 : pour la mise en forme.
- JavaScript : pour créer le système de notification et importer les données à partir du ThingSpeak Cloud en utilisant AJAX.
- PHP : pour gérer le mécanisme d'authentification.
- SQL : pour la gestion de la base de données.

Dans ce qui suit, on va présenter les différentes interfaces de l'application.

#### IV.1.2.1 Présentation des interfaces

Sur les deux figures (Figure IV.11, Figure IV.12), nous trouvons l'interface d'accueil de notre application invitant l'utilisateur à s'authentifier en saisissant son propre nom d'utilisateur et mot de passe.

Une fois l'authentification est réussie, l'utilisateur sera redirigé vers la page de contrôle (Figure IV.13, Figure IV.14) sur laquelle il peut : consulter les valeurs de la température et de l'humidité, s'informer de la dernière date/heure du mouvement produit ou du gaz/fumée détecté, comme il peut aussi cliquer sur : le bouton "visualize" pour afficher les dix dernières valeurs de la température ou de l'humidité via une courbe représentative (Figure IV.15, Figure IV.16), et sur le bouton "reset" pour réinitialiser les fonctions de détection de mouvement et de gaz/fumée.

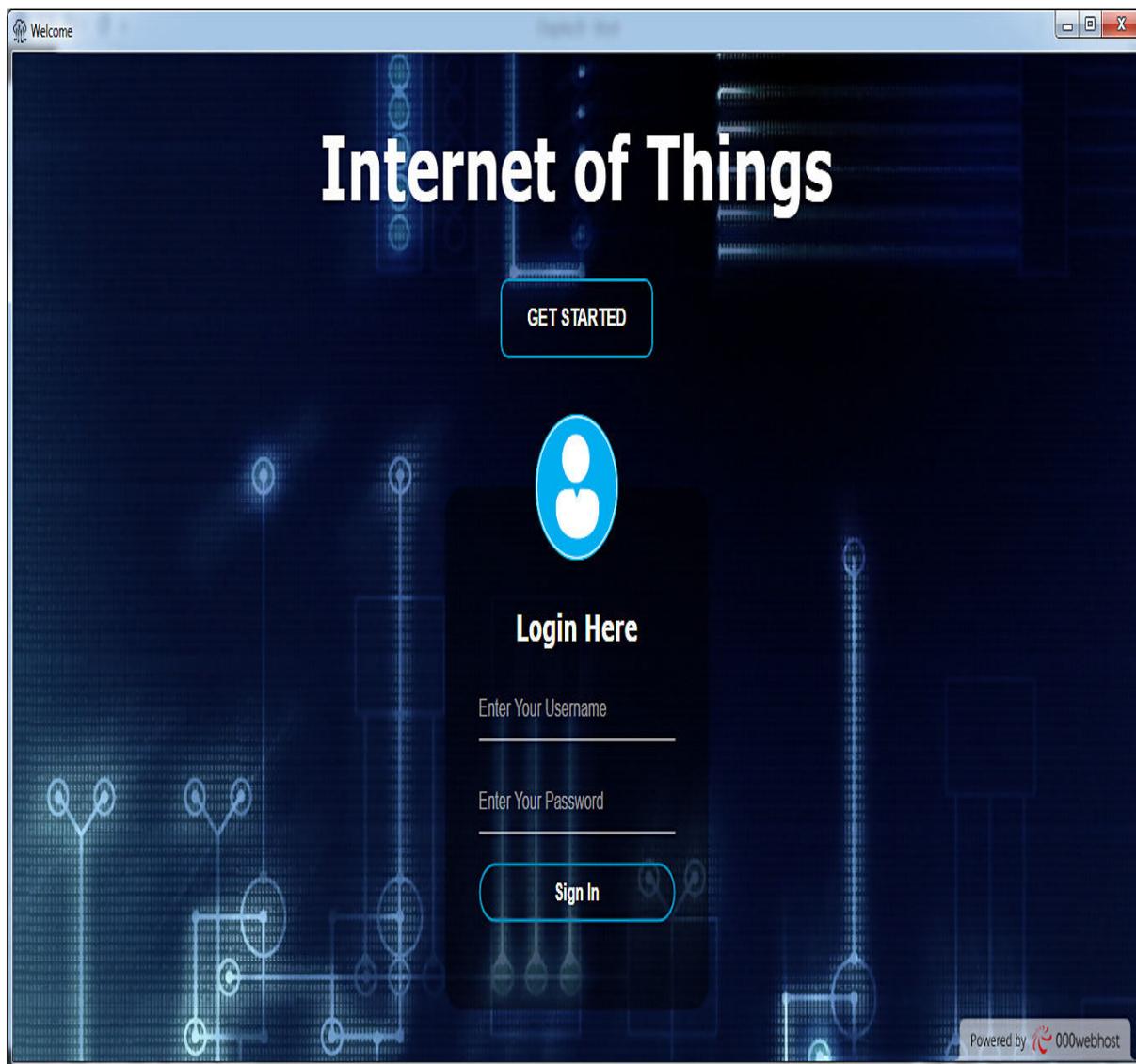


FIGURE IV.11 – Interface d'authentification de la version ordinateur de l'application Web.

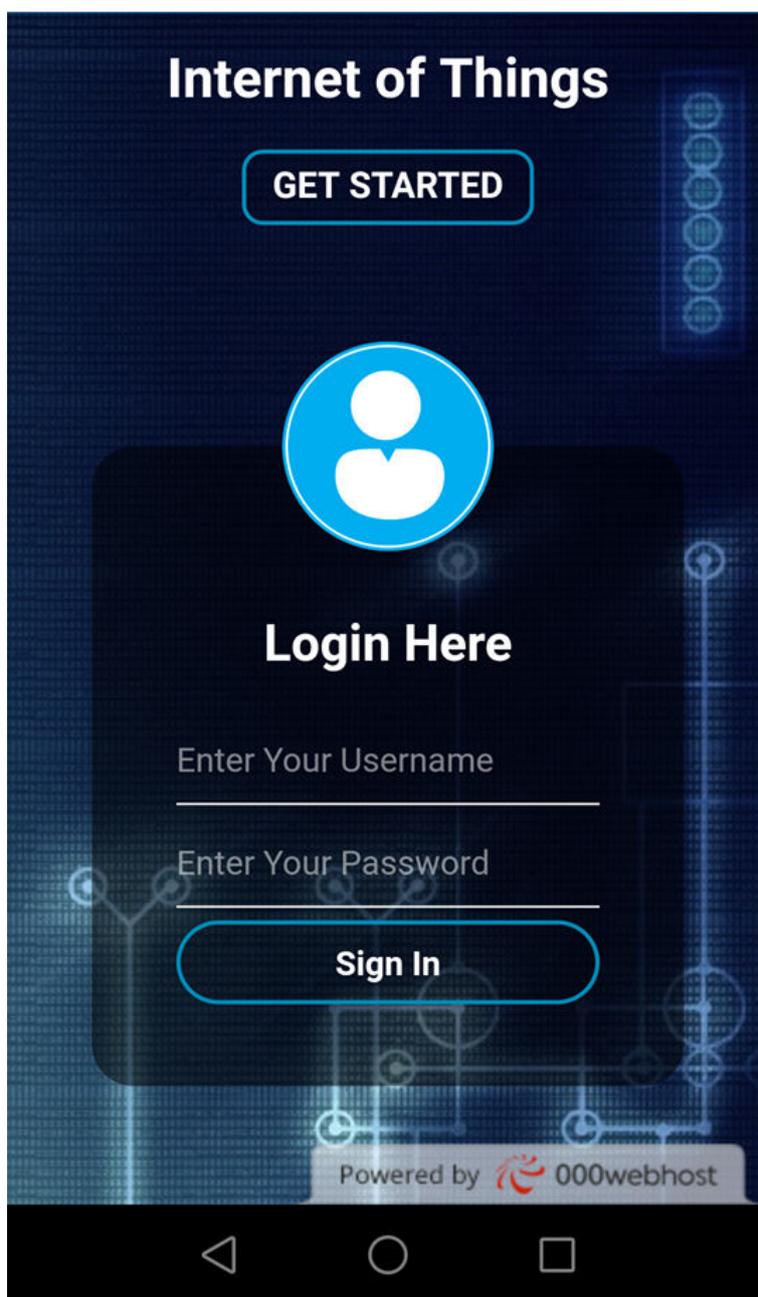


FIGURE IV.12 – Interface d’authentification de la version mobile de l’application Web.

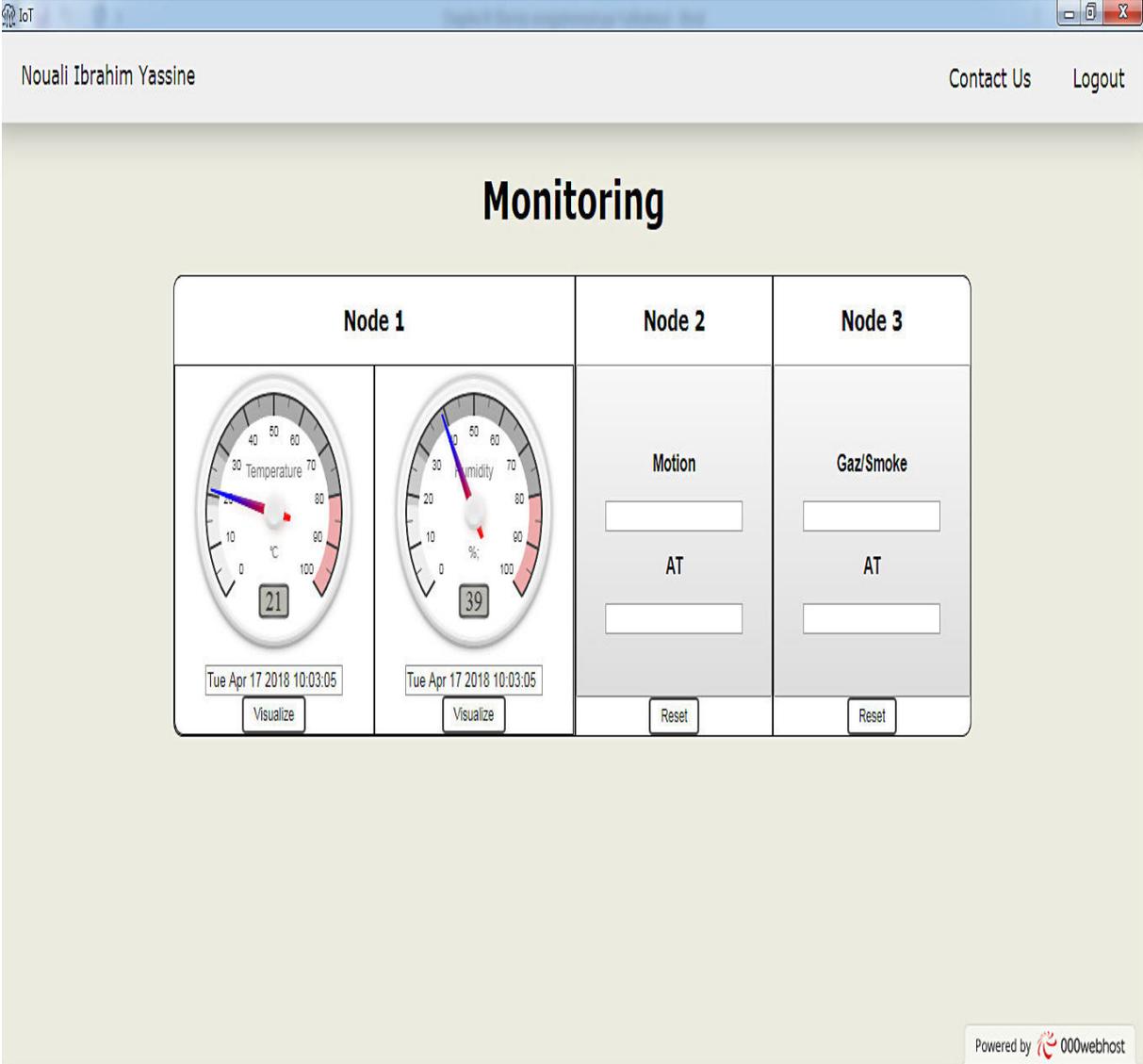


FIGURE IV.13 – Interface de contrôle de la version ordinateur de l’application Web.

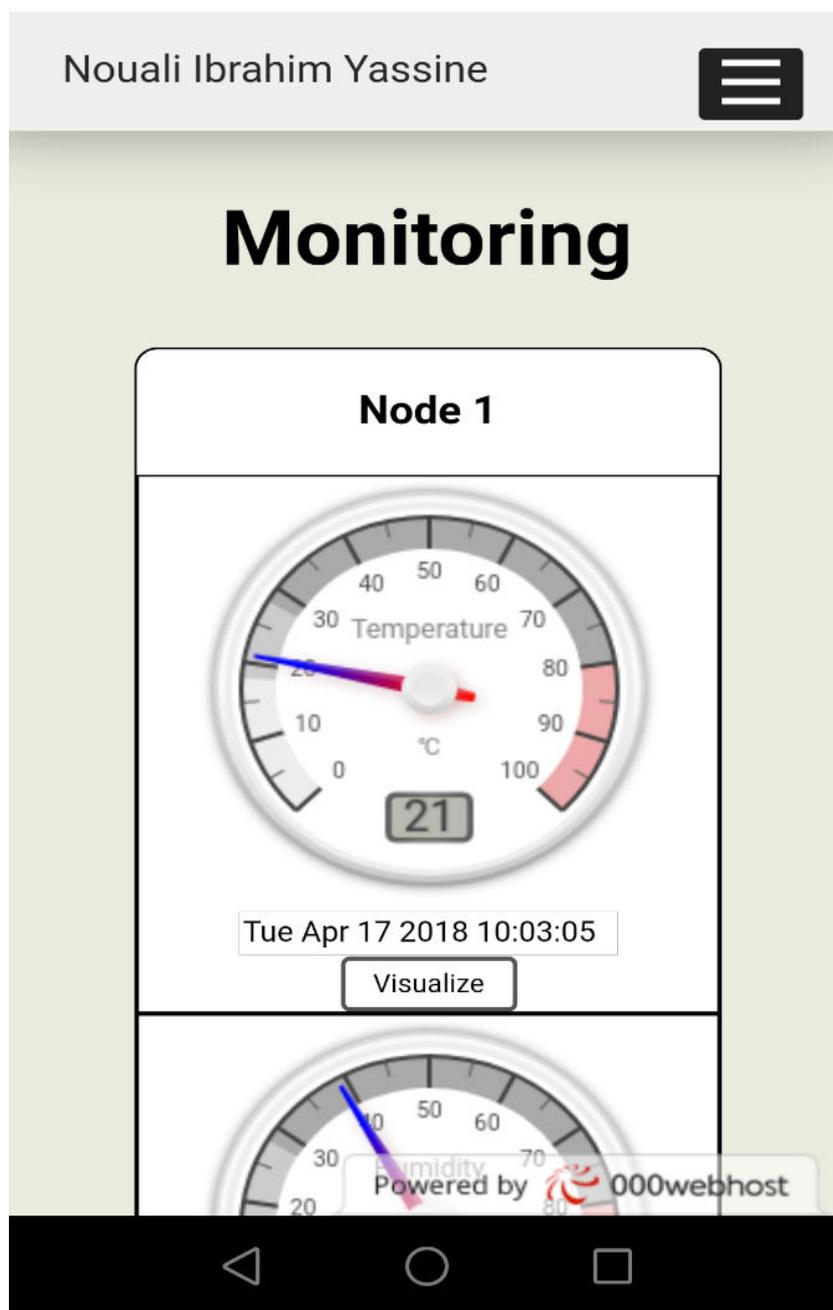


FIGURE IV.14 – Interface de contrôle de la version mobile de l'application Web.

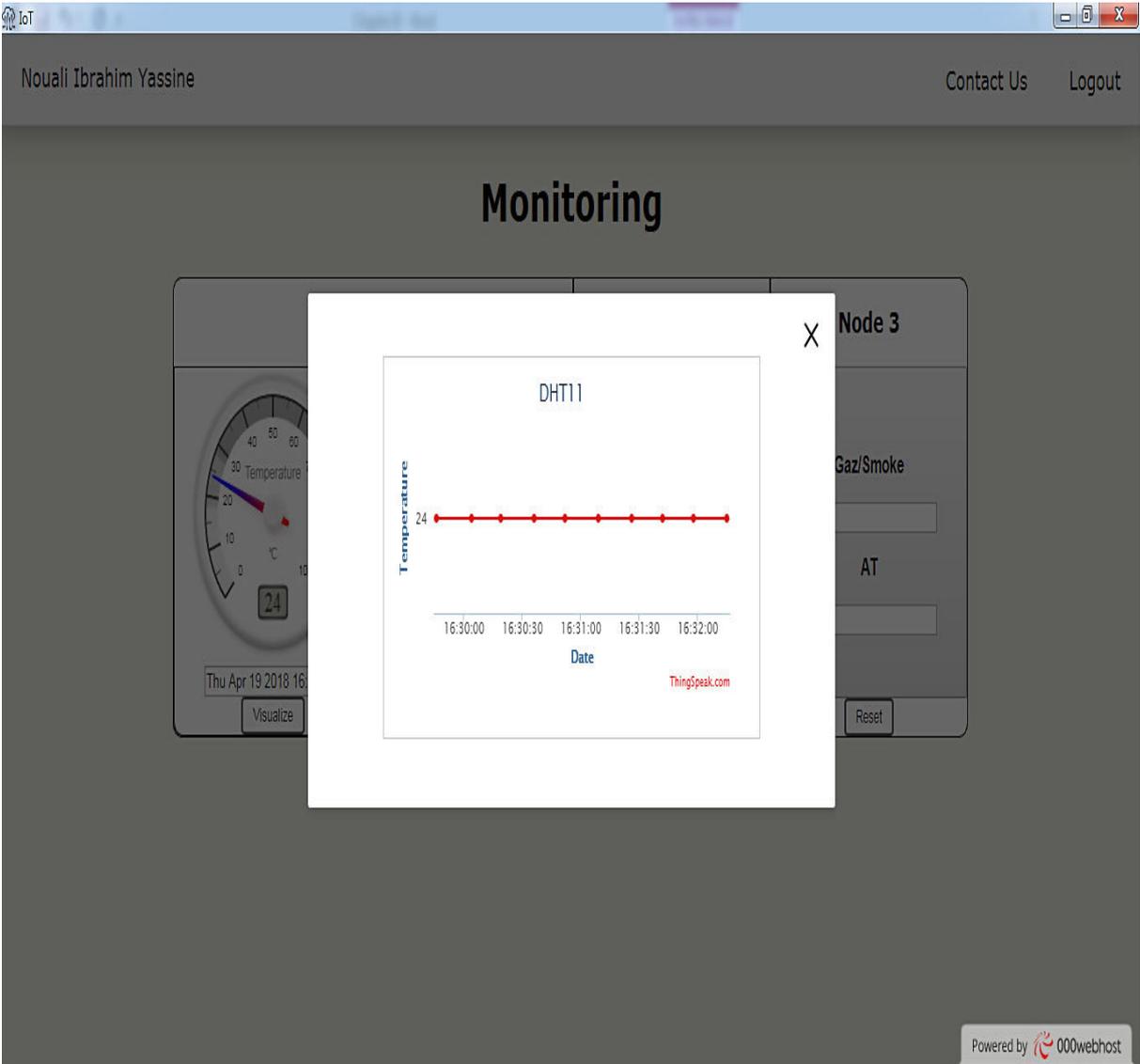


FIGURE IV.15 – Courbe représentative des dix dernières valeurs de la température.

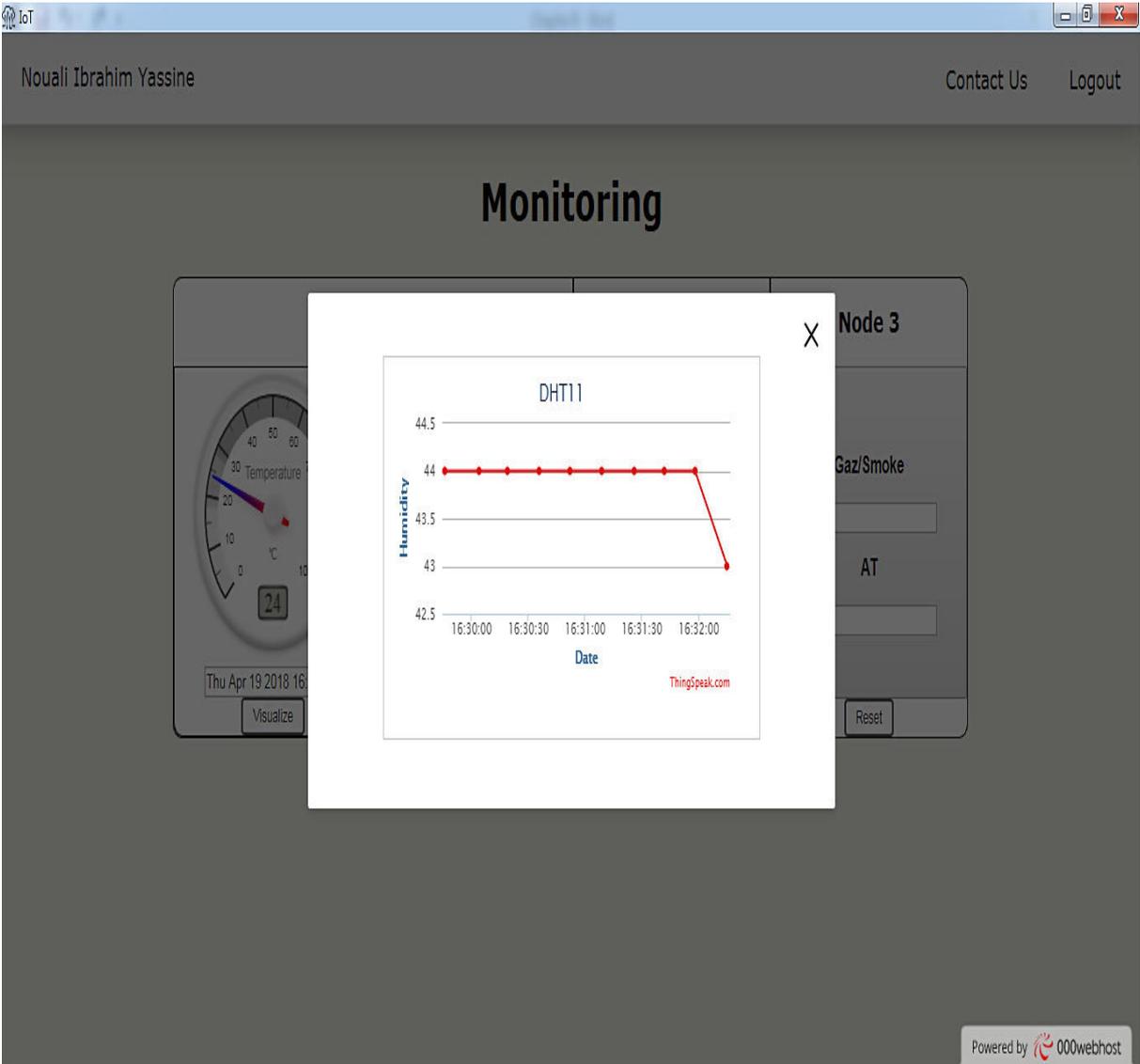


FIGURE IV.16 – Courbe représentative des dix dernières valeurs de l’humidité.

### IV.1.2.2 Principe de fonctionnement

Le principe de fonctionnement est expliqué par des organigrammes associés au code utilisé pour le développement de l'application.

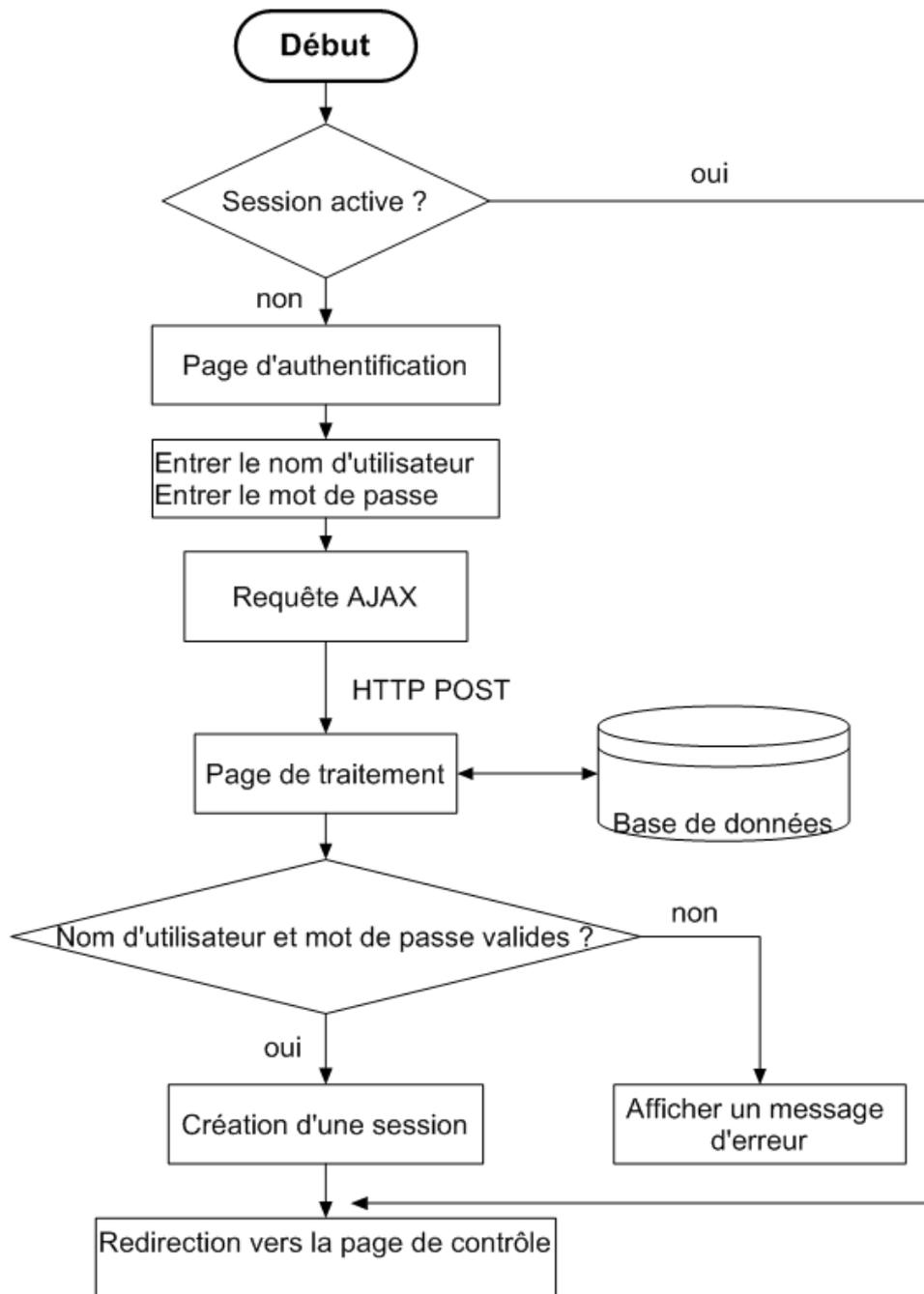


FIGURE IV.17 – Organigramme d'authentification.

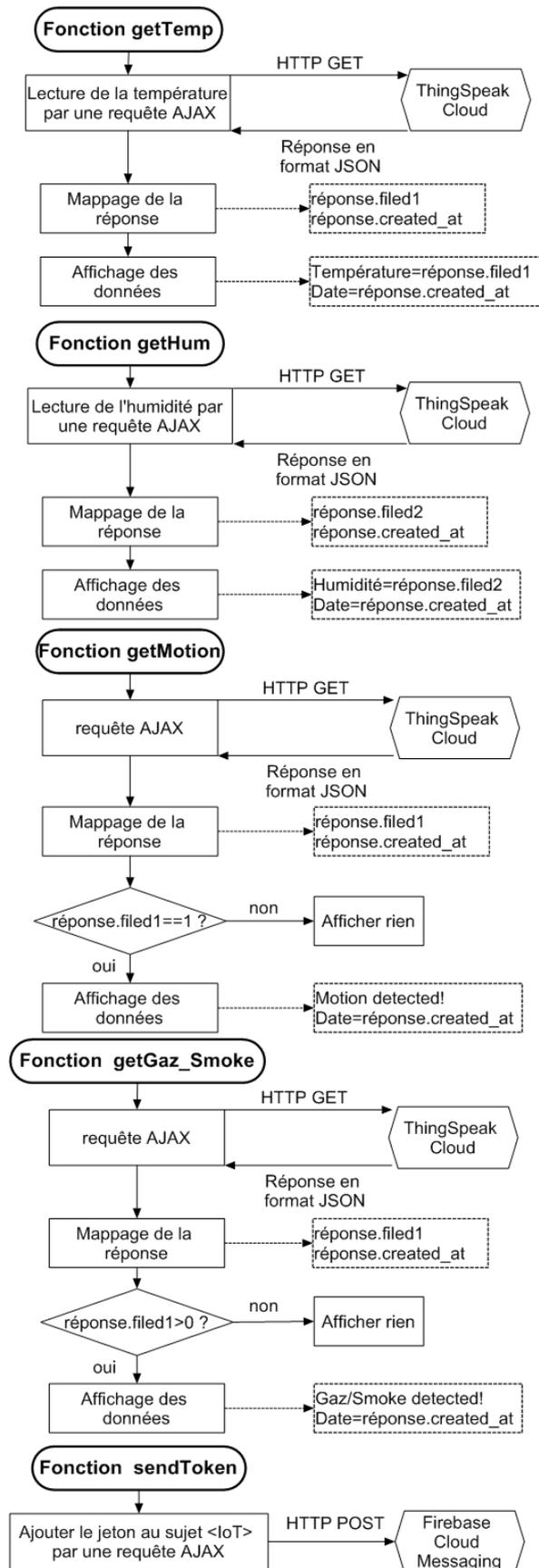


FIGURE IV.18 – Organigrammes des fonctions de contrôle.

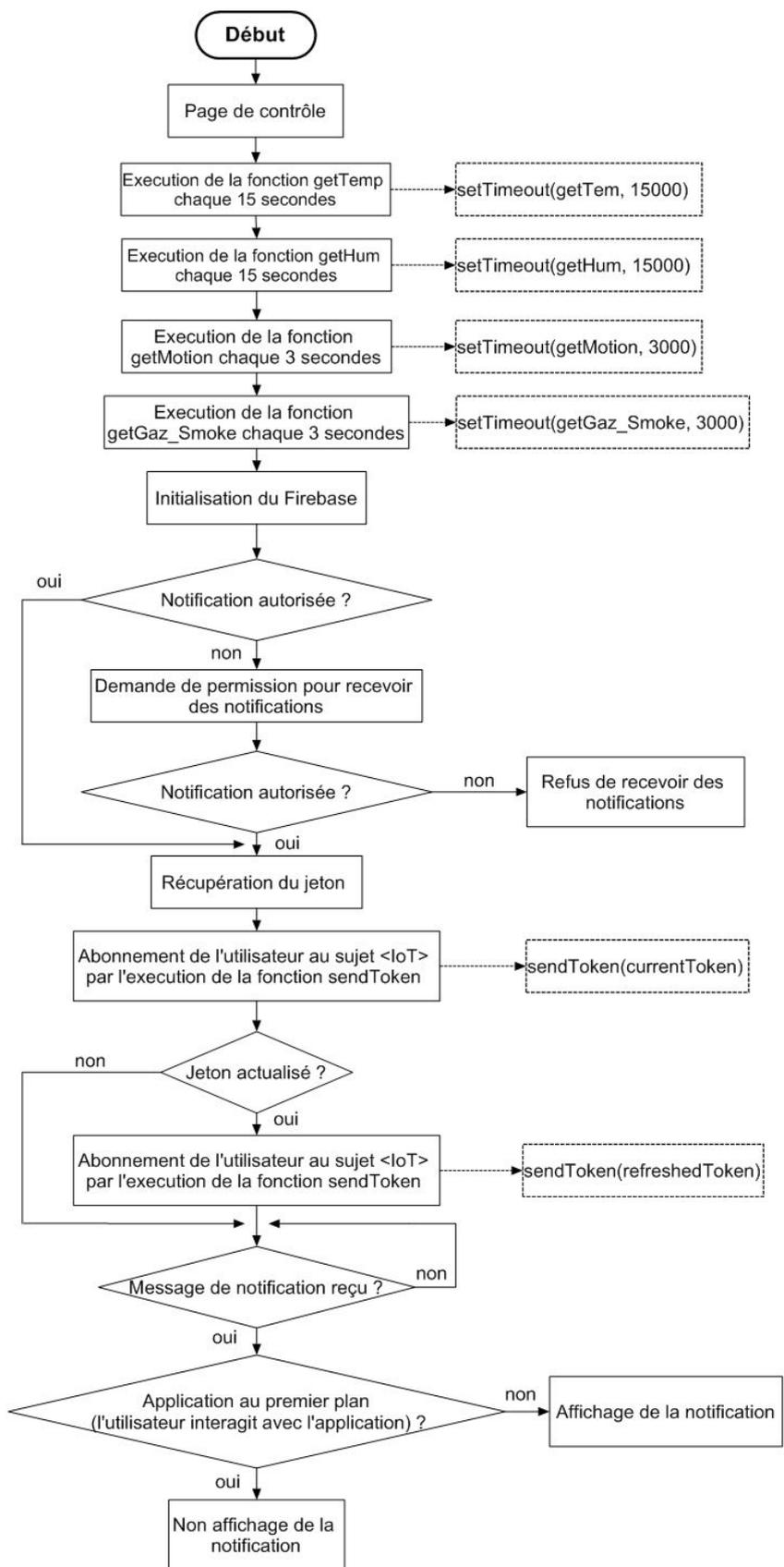


FIGURE IV.19 – Organigramme de contrôle.

Les fonctions `getTemp`, `getHum`, `getMotion` et `getGaz_Smoke` nous permettent de calculer respectivement la température, l'humidité, le mouvement et les quantités de gaz/fumée. Ces fonctions envoient des requêtes grâce au protocole HTTP via la méthode GET au ThingSpeak Cloud. À l'issue de ces requêtes, une réponse est reçue depuis le serveur sous format JSON. Dans le cas de la température ou de l'humidité, on obtient directement la réponse, mais dans le cas du mouvement, un test doit être fait sur la réponse. Si cette dernière est égale à un, cela signifie qu'il y a un mouvement détecté. Idem pour le cas du gaz et de la fumée. La fonction `sendToken` permet un abonnement de l'utilisateur au sujet IoT afin de recevoir des messages de notifications envoyés par les nœuds communicants (nœud 2, nœud 3). Ceci est réalisé en envoyant son jeton récupéré par une requête HTTP POST au Firebase Cloud Messaging.

# CONCLUSION

Le développement de ce mémoire m'a permis d'explorer différents sujets, en se terminant par la réalisation d'un prototype de surveillance à distance basé sur la technologie d'Internet des objets où on a formé trois nœuds chacune contrôle un paramètre bien définis (présence non autorisée, un gaz combustible ou fumée, température et humidité). De plus, on a développé une application Web en créant une interface de surveillance à l'utilisateur avec des visualisations et des notifications en temps réel. L'interconnexion entre ces deux se fait via des plateformes IoT Cloud.

Ce prototype bien qu'il soit fonctionnel, il présente certaines limites qu'il convient de les énumérer si on veut l'exploiter dans une situation réelle. La première considération à faire, c'est l'utilisation d'un seul nœud pour chaque paramètre, alors qu'une situation réelle exige l'implantation de plusieurs nœuds.

Le deuxième point digne de remarque est le manque de cryptage des informations envoyées par les nœuds communicants ce qui rend le système vulnérable aux attaques comme l'homme au milieu, où l'information est interceptée ou modifiée avant d'atteindre sa destination. Il est donc approprié d'intégrer des protocoles cryptographiques tels que SSL ou TLS pour protéger ces informations.

En outre, pour que le prototype fonctionne correctement, il est nécessaire que les nœuds de contrôle soient constamment en ligne. Dans une mise en œuvre réelle, il est donc opportun de prendre toutes les précautions afin de préserver cette condition, comme l'activation de l'alimentation par batterie lorsque l'électricité n'est pas disponible. Au niveau de l'application, l'utilisateur pourrait être averti du passage à une alimentation par batterie, afin qu'il puisse intervenir plus tard pour résoudre le problème.

Finalement, en ce qui concerne les plateformes IoT Cloud, celles-ci réduisent le rendement de notre prototype à cause des limitations imposées sur l'utilisation gratuite. Dans ce cas, le nombre des requêtes depuis et vers ces plateformes est limité. La solution possible consiste à déployer notre propre serveur ou passer à un service payant de ces plateformes.

# BIBLIOGRAPHIE

- [B1] Benjamin Billet, *Système de gestion de flux pour l'Internet des objets intelligents*. Université de Versailles Saint-Quentin En Yvelines, 2015.
- [B2] François Baccelli, "Les réseaux de communication du futur.", 2010.
- [B3] Pierre-Jean Benghozi, Sylvain Bureau, Françoise Massit-Folléa, *L'internet des objets Quels enjeux pour l'Europe*. Maison des sciences de l'homme, 2015.
- [B4] Tanwar Sudeep, Sudhanshu Tyagi et Sachin Kumar, "The Role of Internet of Things and Smart Grid for the Development of a Smart City.", 2018.
- [B5] Cédric Locqueneux, *Le guide de la maison et des objets connectés : Domotique, smart home et maison connectée*. Eyrolles, 2016.
- [B6] R.Raut, H.Varma, C.Mulla, et al, "Soil Monitoring, Fertigation, and Irrigation System Using IoT for Agricultural Application.", 2018.
- [B7] Baig, Mirza Mansoor et Gholamhosseini, Hamid, "Smart health monitoring systems : an overview of design and modeling.", 2013.
- [B8] Somia Sahraoui, *Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things)*. Université de Batna 2, 2016.
- [B9] Numa de Montmollin, *Implémentation de plusieurs cas d'utilisation, utilisation et création d'outils pour développer des applications RESTful du Web des Objets*. Université de Fribourg, 2014.
- [B10] Gérôme Bovet, Hennebert Jean, "Communicating With Things - An Energy Consumption Analysis.", 2012.

- [B11] Erik Bartmann, *Le grand livre d'Arduino*. Eyrolles, 2015.
- [B12] Massimo Banzi, *Getting Started with Arduino*. O'Reilly Media, 2011.
- [B13] Housseem Medhioub, *Architectures et mécanismes de fédération dans les environnements cloud computing et cloud networking*. Institut National des Télécommunications, 2015.
- [B14] Pasha, Sharmad, "ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis.", 2016.
- [B15] Lucien Loiseau, *De l'exploitation des réceptions opportunistes dans les mécanismes de relayage pour les réseaux sans-fil*. Télécom Bretagne, Université de Rennes, 2013.
- [B16] Aurélien Géron, *WiFi Professionnel- 3e édition - : La norme 802.11, le déploiement, la sécurité*. Dunod, 2009.
- [B17] David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy, *HTTP : The Definitive Guide*. O'Reilly Media, 2002.
- [B18] Gastón C. Hillar, *MQTT Essentials - A Lightweight IoT Protocol*. Packt, 2017.

# WEBOGRAPHIE

- [W1] [https://www.cisco.com/c/dam/global/en\\_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf](https://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf) (accédé le 03/06/2018).
- [W2] <http://www.fao.org/climatechange/epic/notre-action/definition-de-lagriculture-intelligente-face-au-climat/fr/> (accédé le 03/06/2018).
- [W3] <https://www.iso.org/fr/news/2016/09/Ref2112.html> (accédé le 03/06/2018).
- [W4] <http://arduino.cc/en/Main/ArduinoBoardUno> (accédé le 03/06/2018).
- [W5] <https://www.mpja.com/download/31227sc.pdf> (accédé le 03/06/2018).
- [W6] <https://www.futura-sciences.com/tech/definitions/informatique-cloud-computing-11573/> (accédé le 03/06/2018).
- [W7] <https://firebase.google.com/docs/cloud-messaging/> (accédé le 03/06/2018).