

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche
scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

Pour l'obtention du diplôme de Master en Système et Télécommunications

Thème

IOT pour le monitoring énergétique

Présenté par

ARKAM ABDELKADER & LEGHRIBI ABDERRAHMANE

Proposé par : Yacine Kabir

Année Universitaire 2019-2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciement

Toute notre gratitude, grâce et remerciement à Dieu le tout puissant qui nous a donné sa foi, la patience et la sagesse.

Nous remercions les membres de jury qui nous font l'honneur de présider et d'examiner ce modeste travail.

A nos enseignant, docteur et encadreur : Mr : Yacine Kabir qui nous a donné l'éducation, la science et le savoir.

A nos mères qui nous porté et soutenu durant toute notre vie, à nos pères qu'ont veillé sur nous de jour comme de nuit.

A nos copains et copines, nos cousins et cousines, et surtout nos frères et sœurs pour nous avoir acoquiné et nous donner le courage à continuer et préserver.

Nous adressons de chaleureux remerciements à tous les enseignants qui on contribue à notre réussite dans nos études, depuis notre première année scolaire jusqu'à la dernière année universitaire.

Enfin, Nous remercions cette mémoire, qui sans leur aide, leur soutien et leurs conseils nous n'aurons pas arrivé à ce stade et peut-être peu d'avantages.

RESUME

La technologie de l'Internet des Objets (IoT) a attiré beaucoup d'attention ces dernières années dans tous les secteurs vitaux et industriels en particulier dans le domaine énergétique.

L'objectif de ce travail est de réaliser une plateforme IoT pour suivre et mieux gérer la consommation d'énergie dans des locaux pour habitation ou entreprise. Cette dernière, est l'un des grands défis dans le domaine de l'IoT.

Nous proposons une solution matérielle et logicielle intégrée sous forme d'une plateforme IoT. La partie matérielle est constituée principalement cartes à microcontrôleurs et de capteurs, tandis que la partie logicielle consiste en deux applications mobile pour contrôler et recevoir des alertes à distance, et une autre application desktop pour l'analyse, la visualisation des données en temps réel et l'archivage.

Mots clés : Internet des Objets, Microcontrôleurs, Platform, Capteurs, IoT

الملخص

حظيت تقنية إنترنت الأشياء (IoT) باهتمام كبير في السنوات الأخيرة في جميع القطاعات الحيوية والصناعية وخاصة في مجال الطاقة. الهدف من هذا العمل هو إنشاء منصة إنترنت الأشياء لمراقبة وإدارة استهلاك الطاقة بشكل أفضل في المباني السكنية أو التجارية. هذا الأخير هو أحد أكبر التحديات في مجال إنترنت الأشياء. نقدم حلاً متكاملاً للأجهزة والبرامج كنظام أساسي لإنترنت الأشياء. يتكون جزء الأجهزة بشكل أساسي من لوحات وأجهزة استشعار وحدات التحكم الدقيقة، بينما يتكون جزء البرنامج من تطبيقات متنقلين للتحكم في التنبيهات عن بُعد وتلقيها، بالإضافة إلى تطبيق سطح مكتب آخر للتحليل، وإظهار البيانات في الوقت الحقيقي وأرشفتها.

الكلمات المفتاحية : إنترنت الأشياء ، ميكروكنترولر ، منصة ، مجسات ، IoT.

Abstract

The Internet of Things (IoT) technology has attracted a lot of attention in recent years in all vital and industrial sectors especially in the energy field.

The objective of this work is to create an IoT platform to monitor and better manage energy consumption in residential or business premises. The latter is one of the great challenges in the field of IoT.

We offer an integrated hardware and software solution in the form of an IoT platform. The hardware part consists mainly of microcontroller boards and sensors, while the software part consists of two mobile applications to control and receive alerts remotely, and another desktop application for analysis, visualization of data in real time and archiving.

Keywords: Internet of Things, Microcontrollers, Platform, Sensors, IoT.

LISTES DES ACRONYMES ET ABRÉVIATIONS

6	GLOWPAN	IPv6 over Low -Power Wireless Personal Area Networks.
A	AC	Alternating Current
	AMQP	Advanced Message Queuing Protocol
	AR	Augmented reality
B	BLE	Bluetooth Low-Energy.
	BLE	Bluetooth à basse consommation
C	CoAP	Constrained Application Protocol
	CRM	Customer Relationship management
D	DC	Direct Current
	DCPS	Data-Centric Publish-Subscribe
	DDS	Data Distribution Service
	DLRL	Defence Electronics Research Laboratory
E	ERP	Enterprise Resource Planning
G	GND	Global Nearness Diagram
H	HRMS	Human Resource Management System
	HTML	HyperText Markup Language
	HTTP	HyperText Transfer Protocol
I	IBM	International Business Machines
	IBSG	Internet Business Solutions Group.
	IDE	Integrated Development Environment
	IOT	Internet of Things
	IP	Internet Protocol
	ISM	International Systems Marketing
	ISO	International Organization of Standardization
	ISO	International Organization for Standardization
L	LAN	Local Area Network
	LPWAN	Low Power Wide Area Network

M	M2M	Machine a machine
	MCU	Micro Controller Unit
	MQTT	Message Queue Telemetry Transport
N	NFC	Near Field Communication
	NGMN	National Groundwater Monitoring Network
O	OSI	Open Systems Interconnexion
Q	QoS	Quality of Service
R	RF	radio frequency
	RF	Radio-fréquence
	RFID	Radio-Frequency Identification
S	SDK	Software Development Kit
	SGBDR	système de Gestion de Base de Données
	SIG	système intégré de gestion
	SoC	Security Operating Center
	SSL	Secure Socket Layer
T	TCP	Transmission Control Protocol
U	UDP	User Datagram Protocol
	UNB	Ultra Narrow Band
	URI	stands for Uniform Resource Identifier
V	VR	Virtual reality

TABLE DES MATIERES

Introduction générale	1
CHAPITRE I : Généralité sur l'iot	3
Introduction	3
I-1- Définition de l'internet des objets.....	3
I-2- L'IoT aujourd'hui	4
I-3- Domaine d'application des systèmes d'IOT	6
I-3-1- Les Villes Intelligentes.....	6
I-3-2- La santé intelligente	7
I-3-3- L'énergie intelligente	8
I-3-4- L'environnement intelligent	8
I-3-5- L'industrie intelligente	9
I-3-6- Le Transport et La mobilité intelligente	10
I-4- Architecture de l'IOT	11
I-4-1- Les quatre couches de L'IOT	11
I-4-2- Les sept niveaux du modèle référence pour l'IoT	12
I-5- Les Défis de l'IOT.....	18
I-5-1- La Découverte Automatique	18
I-5-2- L'Interopérabilité	18
I-5-3- La Sécurité et La Confidentialité.....	18
I-5-4- La Tolérance Aux Pannes	19
I-5-5- L'Auto-Organisation	19
I-6- Les bénéfices de l'IoT	19
Conclusion	20
CHAPITRE II : Les différentes ProtocolES de l'IOT	21
Introduction.....	21
II-1- Connectivité IoT	21
II-1-1 Connexion des appareils IOT	21
II-1-2 Les passerelles IOT	22
II-1-3 Connections des appareils IOT au réseau.....	22
II-2- Types de réseaux IOT	22
II-2-1- Réseaux à faible puissance et à courte portée.....	22
II-2-2- Liaisons sans fil à faible consommation énergétique (LPWAN).....	25
II-3- Protocoles IoT	29
II-3-1- Couche Application	29

II-3-2-Couche Transport	31
II-3-4- Couche Liaison de données.....	33
II-3-5- Couche Physique	33
II-4-Le protocole MQTT	36
II-4-1- Définition.....	36
II-4-2- Mode de fonctionnement MQTT	36
II-4-3- Abonnements et Publications.....	37
II-4-4 Les topics	38
II-4-5- Qualité de service.....	38
CHAPITRE III : Implémentation et Réalisation	41
Introduction	41
III-1- L'environnement de travail.....	41
III-1-1- Logiciel Arduino	41
III-1-2- Esp8266 NodeMCU	43
III-1-3- Support de carte NodeMCU	45
III-1-4- Les capteurs	46
III-1-5- MQTT Dash	47
III-1-6- Le broker MQTT	47
III-1-7- Les technologies web.....	48
III-2- La Réalisation	49
III-2-1- Le montage de Projet Réalise :	49
III-2-2- La Programmation de ESP8266 NodeMCU :	51
III-2-3- Test de Mosquitto	56
III-2-4- l'Application avec dans MQTT Dash	58
III-2-4-Resulats de simulation	62
Conclusion	70
CONCLUSION GENERALE	71
REFERENCES BIBLIOGRAPHIQUES	

LISTE DES FIGURES

Figure I-1 Internet des objets.....	4
Figure I-2 L'évolution des objets connecté	5
Figure I-3 Les constituants d'une ville intelligente	6
Figure I-4 Le mécanisme de la surveillance distants des patients	7
Figure I-5 Les constituants d'une Smart Grid	8
Figure I-6 Les constituants de l'environnement intelligent.....	9
Figure I-7 Les constituants de l'industrie intelligente	10
Figure I-8 Quelques aspects du transport intelligent	10
Figure I-9 Le Modèle de référence de L'IoT [6].	12
Figure II-1 Modèle pour transférer des données entre deux client (publish / subscribe) à travers le broker	37
Figure II-2 Niveaux de Qualité de Service MQTT.....	39
Figure III-1 Interface du logiciel IDE.....	41
Figure III-2 Les boutons d'action.....	42
Figure III-3 La carte NodeMCU (ESP8266)	43
Figure III-4 Composition d'une carte NodeMCU (ESP8266).....	44
Figure III-5 Les différents composant d'esp8266	45
Figure III-6 Support de carte NodeMCU	45
Figure III-7 Le Support avec la carte NodeMCU	46
Figure III-8 Capteur DHT11.....	46
Figure III-9 Schéma de cablage de capture DHT11 et la LED avec la carte ESP8266.....	50
III-10 Le montage de projet réalisé	50
Figure III-11 Définition des paramatres (wifi,adress IP, les topics et les pin liés au capteur et la LED)	51
Figure III-12 Connecter a un reseaux wifi.....	51
Figure III-13 Reconnection a mqtt en cas de déconnexion	52
Figure III-14 Lécure des valeurs du capteur	52
Figure III-15 Verification du changement des valeurs de temperature et humidite	53
Figure III-16 Affichage d'une alerte en cas de sucronsommation (valeur <460wh).....	53
Figure III-17 Connexion del'esp8266 (vérification sur le moniteur serie)	54
Figure III-18 Affichage des valeurs (temp/hum/elec) sur le moniteur série	54

Figure III-19 Organigramme de programme Arduino.....	55
Figure III-20 Lancement de serveur mosquitto	56
Figure III-21 souscription au topic "sensor/temperature"	56
Figure III-22 souscription au topic "sensor/humidity"	57
Figure III-23 souscription au topic "sensor/electricity"	57
Figure III-24 souscription au trois topics (température, humidité et électricité) commande "sensor/#"	58
Figure III-25 Configuration App MQTT Dash N°1	59
Figure III-26 Configuration App MQTT Dash N°2	59
Figure III-27 Création de l'indicateur de température.....	60
Figure III-28 Création de l'indicateur de l'humidité.	60
Figure III-29 Création de l'indicateur de l'électricité.	61
Figure III-30 Interface de notre application sur MQTT Dash	61
Figure III-31 Interface sur MQTT Dash (exemple avec alerte sur la consommation d'électricité > 460w/h).....	62
Figure III-32 Led (allumé) pour alerte de surconsommation sur l'ESP8266	62
Figure III-33 La base de données MySQL"office"	63
Figure III-34 La table du capteur de consommation d'électricité	64
Figure III-35 La table du capteur Humidité.....	64
Figure III-36 La table du capteur Température	65
Figure III-37 Organigramme de connexion entre le Broker MQTT et la base de données MySQL.....	65
Figure III-38 Script de connexion python-MQTT.....	66
Figure III-39 Script de connexion python-My-SQL.....	66
Figure III-40 L'interface Web	67
Figure III-41 Organigramme d'affichage les dernier valeur capturer.	68
Figure III-42 Interface d'affichage de la consommation.....	69
Figure III-43 Organigramme qui afficher les valeur dans une intervalle precise du temps.	70

INTRODUCTION GENERALE

L'Internet est devenu en quelques années le vecteur principal de diffusion de l'information. Il s'est imposé dans de nombreux domaines comme une infrastructure essentielle pour les individus, les entreprises et les institutions. Toutefois, ses capacités d'extension, au-delà des seuls ordinateurs et terminaux mobiles, sont encore considérables car il devrait permettre l'interaction d'un nombre croissant d'objets entre eux ou avec nous-mêmes.

Internet transforme progressivement en un réseau étendu, appelé « Internet des Objets », reliant plusieurs milliards d'êtres humains mais aussi des dizaines de milliards d'objets [1].

Mais l'image que projette IoT de demain est beaucoup plus vaste que celle d'aujourd'hui. On parle plutôt d'objets en mode plug and Play, capables de se découvrir et de communiquer entre eux. Et de réseaux d'objets connectés qui embarquent de l'intelligence qui est capable de prendre des décisions et de faire des actions de façon autonome [2].

De toute évidence, Internet est l'un des plus importants et des plus puissantes créations dans toute l'histoire humaine.

Ce travail consiste à réaliser une plateforme IoT permettant de suivre la consommation d'énergie dans des locaux pour habitation ou entreprise. Le système, permet de suivre la consommation d'énergie électrique en temps réels, et permet à long terme d'avoir un profil de consommation personnalisé. Ce profil sert à prédire les consommations futures et détecter d'éventuelles surconsommations.

Le système est en mesure de renseigner l'utilisateur sur les horaires de pics de consommations, et ainsi l'aider à mieux optimiser ses consommations.

La plateforme est constituée de deux parties, la partie client (locaux avec les capteurs intelligents) et la partie serveur qui se charge de collecter les informations issues des capteurs du client.

Nous allons présenter notre travail dans trois chapitres :

Chapitre 1 : est consacré à définir l'Internet des Objets, ses différents domaines d'applications, nous expliquons l'architecture de l'IoT (les quatre couches et les sept niveaux du modèle référence), et nous mentionnons les défis et les bénéfices de l'IoT.

Chapitre 2 : dans ce chapitre, nous présentons les protocoles et connectivité IoT, les types de réseaux IoT, ainsi que le Protocol MQTT.

Chapitre 3 : et enfin nous présentons l'environnement de travail de notre projet, ainsi que les différentes étapes de la réalisation.

A la fin, nous terminons notre mémoire par une conclusion générale.

CHAPITRE I : GENERALITE SUR L'IOT

Introduction

Ce chapitre sera consacré à l'étude théorique d'un système de l'internet des objets, nous présentons l'historique de l'IoT et les différents domaines d'application, ensuite nous citerons l'architecture de l'IoT, ainsi que les défis et les bénéfices de l'IoT.

I-1- Définition de l'internet des objets

Historiquement le terme IoT a été utilisé pour la première fois en 1999 par Kevin Ashton, un ingénieur britannique, pour décrire un système où les objets physiques sont connectés à internet.

Au fil du temps, le terme a évolué et il englobe maintenant tout l'écosystème des objets connectés. L'IoT commence ainsi dans le monde physique avec les capteurs qui recueillent les informations, elles sont ensuite transmises grâce à la connexion et l'intégration des systèmes entre eux, les données sont enfin traitées et stockées pour être analysées et exploitées.

On pense que l'IoT améliorera l'efficacité énergétique, la surveillance à distance et le contrôle des actifs physiques et la productivité grâce à des applications aussi diverses que la sécurité à domicile pour surveiller la condition sur le plancher de l'usine.

Maintenant, l'IoT a été utilisé sur les marchés dans le domaine des soins de santé à domicile appareils et bâtiments, marchés de détail, entreprises énergétiques et manufacturières, mobilité et le transport, les entreprises de logistique et les médias.

Les équipements sont de plus en plus numérisés et connectés, créant des réseaux entre les machines, les humains et Internet, conduisant à la création de nouveaux écosystèmes qui permettent une productivité plus élevée, une meilleure efficacité énergétique et une rentabilité plus élevée [3].



Figure I-1 Internet des objets

I-2- L'IoT aujourd'hui

Avant de parler de l'état actuel de l'IoT, il est important de s'entendre sur une définition. Selon Cisco Internet Business Solutions Group (IBSG), l'IoT correspond simplement au moment où il y a eu plus « de choses ou d'objets » connectés à Internet que de personnes.

En 2003, il y avait environ 6,3 milliards de personnes vivant sur la planète et 500 millions d'appareils connectés à Internet. En divisant le nombre d'appareils connectés par la population mondiale, nous constatons qu'il y avait moins d'un (0,08) appareil pour chaque personne. Selon la définition de Cisco IBSG, l'IoT n'existait pas encore en 2003 car le nombre d'objets connectés était relativement faible. En outre, les appareils les plus répandus actuellement, et notamment les smartphones, faisaient tout juste leur apparition sur le marché.

La croissance explosive des smartphones et des tablettes PC a amené le nombre d'appareils connectés à Internet à 12,5 milliards en 2010, tandis que la population humaine mondiale est passée à 6,8 milliards, ce qui rend le nombre d'appareils connectés par personne supérieur à 1 (1,84 pour être exact) pour la première fois de l'histoire [4].

En janvier 2009, une équipe de chercheurs basée en Chine a étudié les données qui transitaient sur Internet pendant plusieurs intervalles de six mois, entre décembre 2001 et décembre 2006. Les résultats présentaient des similitudes avec les propriétés de la loi de Moore et montraient que la taille d'Internet doublait tous les 5,32 ans. Après avoir combiné

ce résultat au nombre d'appareils connectés à Internet en 2003 (soit 500 millions d'après l'institut de recherche Forrester Research) et à la population mondiale estimée par le Bureau du recensement des États-Unis, Cisco IBSG a évalué le nombre d'appareils connectés par personne.

En affinant ces chiffres, Cisco IBSG a situé l'apparition de l'IoT entre 2008 et 2009 (voir figure I-2). Aujourd'hui l'IoT prend de l'ampleur, notamment en raison de la progression d'initiatives Cisco telles que la plate-forme Planetary Skin, les réseaux et les véhicules intelligents [4].

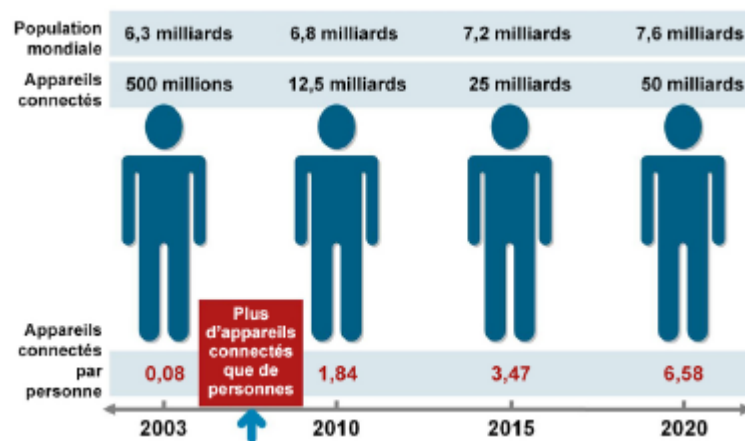


Figure I-2 L'évolution des objets connectés

En regardant vers l'avenir, Cisco IBSG avait déjà prédit qu'il y aura 50 milliards d'appareils connectés en 2020. Il est important de noter que ces estimations ne tiennent pas en compte les progrès rapides de la technologie Internet ou des appareils ; les chiffres présentés sont basés sur ce que l'on sait être vrai aujourd'hui.

De plus, le nombre d'appareils connectés par personne peut sembler faible. C'est parce que le calcul est basé sur l'ensemble de la population mondiale, dont une grande partie n'est pas encore connectée à Internet. En réduisant l'échantillon de population aux personnes réellement connectées à Internet, le nombre d'appareils connectés par personne augmente considérablement.

Par exemple, nous savons qu'environ 2 milliards de personnes utilisent Internet aujourd'hui. En utilisant ce chiffre, le nombre des appareils connectés par personne passe à 6,25 en 2010, au lieu de 1,84 [4].

Bien sûr, nous savons que rien ne reste statique, surtout en ce qui concerne Internet. Avec des initiatives et des avancées telles que la plate-forme Planetary Skin de Cisco, le projet CeNSE (Central Nervous System for the Earth) d'HP et le système smartdust (parfois traduit « poussière intelligente »), le nombre de capteurs connectés à Internet pourrait augmenter de plusieurs millions, voire de plusieurs milliards. En effet tout va être connectés, à mesure que les vaches, les conduites d'eau, les personnes et même les chaussures, les arbres et les animaux deviennent connectés à l'IoT, le monde a le potentiel de devenir un endroit meilleur [4].

I-3- Domaine d'application des systèmes d'IOT

L'internet offre de nombreuses applications à ses utilisateurs. Parmi ces applications nous citons :

I-3-1- Les Villes Intelligentes

Beaucoup de grandes villes ont été soutenues par des projets intelligents, comme Séoul, New York, Tokyo, Shanghai, Singapour, Amsterdam et Dubaï. Les villes intelligentes peuvent encore être considérées comme des villes de l'avenir et la vie intelligente, et par le taux d'innovation de la création de villes intelligentes d'aujourd'hui, il sera devenu très faisable pour entrer la technologie IoT dans le développement des villes.

La demande exige une planification minutieuse à chaque étape, avec l'appui de l'accord des gouvernements, citoyens à mettre en œuvre la technologie de l'Internet des objets dans tous les aspects. Par l'IoT, les villes peuvent être améliorées à plusieurs niveaux, en améliorant les infrastructures, en améliorant les transports [5].



Figure I-3 Les constituants d'une ville intelligente

I-3-2- La santé intelligente

L'internet des objets a rapidement transformé la prestation de soins. Les équipements et les capteurs sont de plus en plus « intelligents » et génèrent toujours plus de données nécessaires aux équipements médicaux, aux professionnels et profitant ainsi aux patients, en réduisant les coûts et en améliorant leur satisfaction. Les données ainsi collectées facilitent, adaptent, améliorent, anticipent ou réorganisent les soins des patients.

- **Surveillance des patients** : Surveillance des conditions des patients dans les hôpitaux et chez les personnes âgées.
- **Réfrigérateurs médicaux** : Contrôle des conditions à l'intérieur des congélateurs stockant des vaccins, médicaments et éléments organiques.
- **Détection Automne** : Assistance pour personnes âgées ou handicapées vivant indépendants.
- **Dentaire** : Bluetooth connecté avec une brosse à dents, avec une application Smartphone permettant l'analyse de brossage et donne des informations sur les habitudes de brossage sur le Smartphone. Ces informations peuvent être utiles pour les utilisateurs (vie privée) ou pour montrer des statistiques au dentiste.
- **Surveillance de l'activité physique** : Capteurs sans fil placés sur le matelas détectant de petits mouvements comme la respiration et la fréquence cardiaque. En lançant et en tournant pendant le sommeil, fournissant des données disponibles via une application sur le Smartphone [5].



Figure I-4 Le mécanisme de la surveillance distants des patients

I-3-3- L'énergie intelligente

L'Énergie intelligente peut apporter une amélioration à la qualité de vie des habitants et faire des économies.

-**Smart Grid** : Surveillance et gestion de la consommation d'énergie, Eoliennes / Centrale électrique : Surveillance et analyse du flux d'énergie des éoliennes et des centrales électriques, et communication bidirectionnelle avec des compteurs intelligents des consommateurs pour analyser les habitudes de consommation.

- **Contrôleurs d'alimentation** : Contrôleur pour Alimentations AC-DC qui déterminent l'énergie requise et améliorent l'efficacité énergétique avec moins de gaspillage d'énergie pour les alimentations liées aux ordinateurs, aux télécommunications et aux applications électroniques grand public.

- **Installations photovoltaïques** : surveillance et optimisation des performances des centrales Solaires [5].

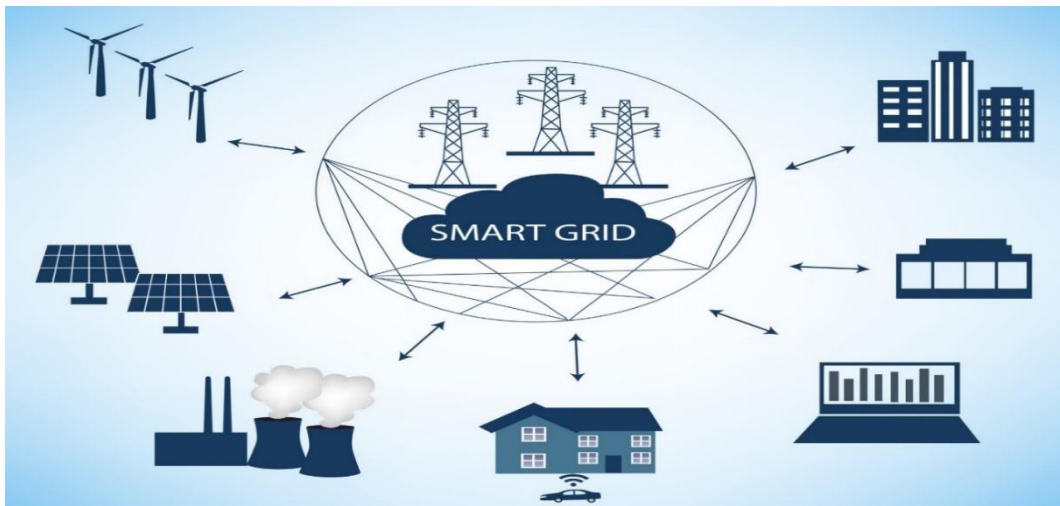


Figure I-5 Les constituants d'une Smart Grid

I-3-4- L'environnement intelligent

L'environnement intelligent s'appuie sur un ensemble de mesures :

- ✓ **Surveillance de la pollution de l'air.**
- ✓ **Contrôle des émissions de CO2** : des usines et des gaz toxiques générés dans les fermes,
- ✓ **Détection des incendies de forêt** : Surveillance des gaz de combustion et des conditions Préemptives d'incendie.

- ✓ **Surveillance météorologique.**
- ✓ **Surveillance de l'humidité, température, pression, vitesse du vent et pluie, séisme.**
- ✓ **Détection précoce, qualité de l'eau :** Étude de la qualité de l'eau dans les rivières et dans la mer pour l'admissibilité à l'usage potable [5].

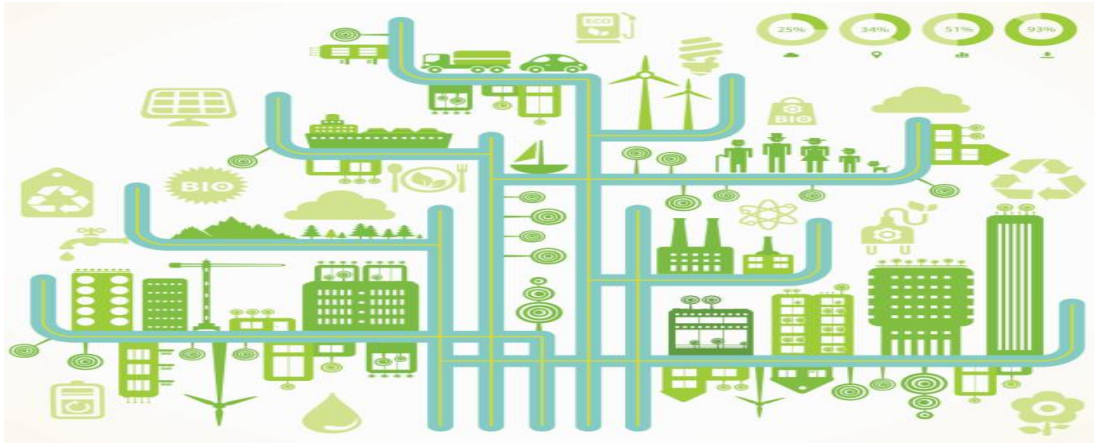


Figure I-6 Les constituants de l'environnement intelligent

I-3-5- L'industrie intelligente

L'industrie intelligente va changer fondamentalement, comment les produits sont inventés, fabriqués et expédiés. En même temps, cela améliorera la sécurité des travailleurs et protège l'environnement un faible incident de fabrication. Ces progrès dans la façon dont les machines et autres objets communiquent, et la manière dont la prise de décision passe des humains aux systèmes techniques signifient que la fabrication devient "plus intelligente".

- ✓ **Gaz explosifs et dangereux :** Détection des niveaux de gaz et des fuites dans les environnements industriels, environnants des usines chimiques et des mines,
- ✓ **Surveillance des niveaux de gaz toxique et d'oxygène dans les usines chimiques :** pour assurer la sécurité des travailleurs et des biens,
- ✓ **Surveillance des niveaux d'eau, de pétrole et de gaz :**
 - Réservoirs et Citerne.
 - Entretien et réparation.
- ✓ **Les prédictions précoces :** sur les défaillances d'équipement et la maintenance du service peuvent être automatiquement programmées avant une défaillance réelle de la pièce en installant des capteurs à l'intérieur de l'équipement pour surveiller et envoyer des rapports [5].

I-4- Architecture de l'IOT

Il n'y a pas de consensus unique sur l'architecture de l'IoT, qui est universellement accepté. Différentes architectures ont été proposées par différents chercheurs.

Comme nous l'avons vu, nous pouvons utiliser l'IoT dans plusieurs domaines et cette technologie porte un grand potentiel en tant que nouveau paradigme des télécommunications sans fil modernes. Nous considérons en particulier la visualisation de l'IoT dans un modèle de référence développé par Cisco. Ce modèle a été développé pour fournir des définitions et des descriptions claires qui peuvent être appliquées avec précision aux éléments et aux fonctions des systèmes et des applications IoT. Le modèle aide à décomposer les systèmes complexes discutés ci-dessus, afin que chaque partie soit plus compréhensible, et fournit également des informations supplémentaires pour identifier avec précision les niveaux IoT et établir une relation avec les systèmes de produits et services. Grâce à ce modèle, nous pouvons également identifier où les types spécifiques de traitement sont optimisés dans différentes parties du système. Cela permettra aux agents de créer des produits e/ou services IoT fonctionnant les uns avec les autres, ce qui rendra l'IoT réel et accessible, plutôt que simplement conceptuel.

Dans un système IoT, les données sont générées par plusieurs types de dispositifs, traitées de manières différentes et transmises dans différents emplacements par des applications. L'architecture orientée aux services a été appliquée comme technologie habilitante par (Xu, He, et Li, 2014) qui divisait l'IoT en quatre couches [6].

I-4-1- Les quatre couches de L'IOT

L'IoT comporte quatre couches distinctes à savoir :

- **Couche de détection** : Cette couche est intégrée avec le hardware existant (RFID, capteurs, actionneurs,.etc.) et sert à détecter/contrôler le monde physique et acquérir des données.
- **Couche de mise en réseau** : Cette couche fournit un soutien réseau de base et le transfert de données par un réseau sans ou avec fil.
- **Couche de service** : cette couche crée et gère les services et fournit des services pour satisfaire les besoins des utilisateurs.
- **Couche d'interface** : cette couche fournit des méthodes d'interaction aux utilisateurs et aux autres applications [6].

I-4-2- Les sept niveaux du modèle référence pour l'IoT

Le modèle de référence IoT que nous proposons comprend sept niveaux. Chaque niveau est défini avec une terminologie qui peut être standardisée pour créer un cadre de référence. Le modèle de référence IoT ne se limite pas à la portée ou à localisation de ses composants. Par exemple, d'un point de vue physique, chaque élément pourrait résider dans un seul rack d'équipement ou pourrait être distribué par ailleurs au monde.

Les réseaux traditionnels de communication de données ont de multiples fonctions, comme en témoigne le modèle de référence à 7 couches de l'Organisation internationale de normalisation (ISO). Cependant, un système IoT complet contient de nombreux niveaux en plus du réseau de communication. Le modèle de référence IoT permet également de varier les traitements effectués à chaque niveau de trivial à complexe, en fonction de la situation. Le modèle décrit comment les tâches à chaque niveau doit être traité pour maintenir la simplicité, permettre une grande évolutivité et assurer un soutien. Enfin, le modèle définit les fonctions nécessaires pour qu'un système IoT soit complet [6].

La figure 9 illustre le modèle de référence IoT et ses niveaux. Il est important de noter que dans l'IoT, les données circulent dans les deux sens. Dans un modèle de contrôle, les informations de contrôle circulent du bas du modèle (niveau 7) vers le haut (niveau 1).

Dans un modèle de surveillance, le flux d'informations est l'inverse. Sur la plupart des systèmes, le flux sera bidirectionnel [6].

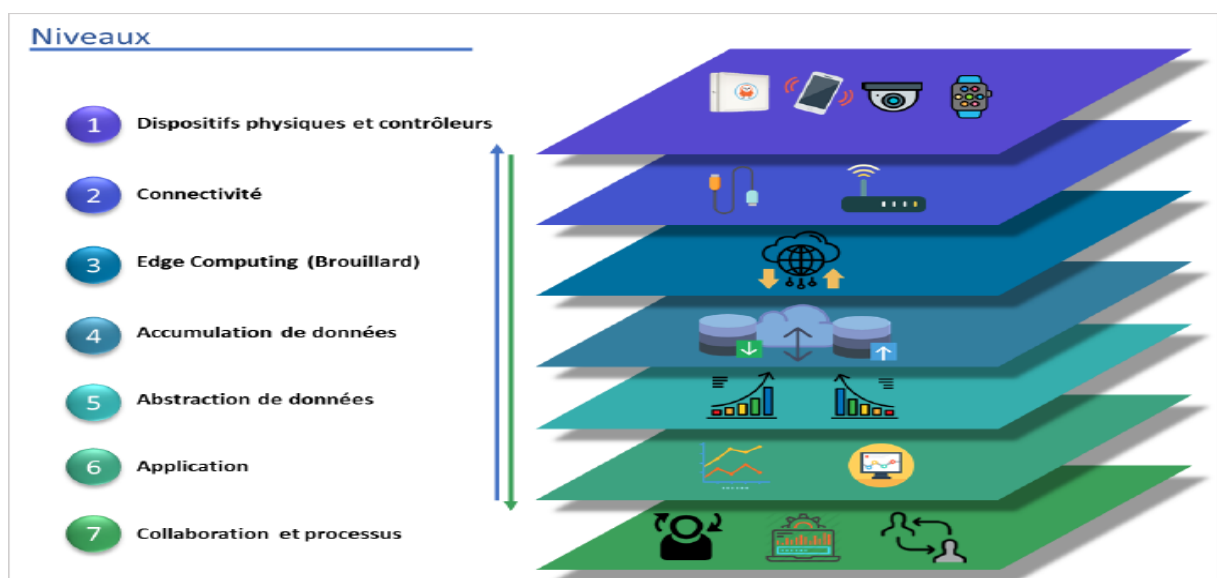


Figure I-9 Le Modèle de référence de L'IoT [6].

Le modèle de référence IoT en sept couches décrit deux flux :

- Dans un modèle de contrôle, un flux descendant, de la couche 7 à la couche 1
- Dans un modèle de surveillance, un flux montant, de la couche 1 à la couche 7

Niveau 1 : Dispositifs physiques et contrôleurs

Le modèle de référence IoT commence par la couche 1 : périphériques physiques et contrôleurs pouvant contrôler plusieurs périphériques. Ce sont les « objets » de l'Internet des objets, et ils comprennent une large gamme d'appareils de terminal qui envoient et reçoivent des informations. Aujourd'hui, la liste des appareils est déjà étendue. Il deviendra presque illimité, car plus d'équipements sont ajoutés à l'IoT au fil du temps [6].

Niveau 2 : Connectivité

Les communications et la connectivité sont concentrées au niveau 2. La fonction la plus importante du niveau 2 est la transmission fiable et opportune de l'information.

Cela comprend les transmissions entre les appareils (niveau 1) et le réseau, entre le réseau (niveau 2) et le traitement des informations de faible niveau se trouvant au niveau 3 et à travers les réseaux (est-ouest) [6].

Niveau 3 : Edge Computing (Brouillard)

Les fonctions de niveau 3 sont dictées par la nécessité de convertir les flux de données du réseau en informations appropriées pour le stockage et le traitement de niveau 4 (hiérarchisé). Cela signifie que les activités de niveau 3 se concentrent sur l'analyse sur place des données, et leur transformation à haut volume. Par exemple, un appareil de détection de niveau 1 peut générer des échantillons de données plusieurs fois par seconde, 24 heures sur 24, 365 jours par an. Un principe de base du modèle de référence IoT est que le système plus intelligent commence à traiter les informations le plus tôt possible à partir des limites du réseau. Ceci est parfois appelé cloud computing. Au niveau 3, c'est là que cela se produit. Étant donné que les données sont généralement soumises à un équipement de réseau de niveau de connectivité (niveau 2) par des dispositifs dans de petites unités, le traitement de niveau 3 est effectué sur une base par paquet. Ce traitement est limité, car il n'y a que la connaissance des unités de données et non pas des

« sessions » ou des « transactions ». Le traitement de niveau 3 peut couvrir de nombreux exemples, tels que [6] :

- **Évaluation** : évalue des données pour déterminer si elles doivent être traitées à un niveau supérieur.
- **Mise en forme** : reformate des données pour un traitement cohérent de haut niveau.
- **Expansion/décodage** : manipule de données cryptiques avec un contexte supplémentaire (comme l'origine).
- **Distillation/Réduction** : réduit et/ou synthétise des données afin de minimiser l'impact des données et du trafic sur le réseau et les systèmes de traitement de niveau supérieur.
- **Évaluation** : détermine si les données représentent un seuil ou une alerte ; cela peut inclure la redirection de données vers des destinations supplémentaires.

Niveau 4 : Accumulation de données

Les systèmes de réseau sont conçus pour déplacer les données en toute sécurité. Les données sont en mouvement. Avant le niveau 4, les données circulent sur le réseau et le taux et à l'organisation sont déterminées par les périphériques qui génèrent les données. Le modèle est piloté par les événements. Comme défini précédemment, les périphériques de niveau 1 n'incluent pas les ressources informatiques. Cependant, certaines activités de calcul peuvent se produire au niveau 2, telles que la traduction de protocole ou l'application de la politique de sécurité réseau. D'autres tâches informatiques peuvent être effectuées au niveau 3, telles que l'inspection d'un paquet. La réalisation de tâches de calcul aussi proches que possible de l'IoT, avec des systèmes hétérogènes répartis sur plusieurs domaines de gestion, représente un exemple de cloud computing. Les services de brouillard et de cloud computing sont une caractéristique distinctive de l'IoT [6].

La plupart des applications ne peuvent ou n'ont pas besoin de traiter des données à la vitesse du réseau. Les applications supposent généralement que les données sont « Au repos » ou immuables en mémoire ou sur disque. A ce niveau, les données en mouvement sont converties en données au repos. Le niveau 4 détermine :

- **Si les données présentent un intérêt pour les niveaux supérieurs** : si c'est le cas, le traitement de niveau 4 est le premier niveau configuré pour répondre aux besoins spécifiques d'un niveau supérieur.
- **Les données doivent-elles être persistantes ?** Les données doivent-elles être conservées sur disque dans un état non volatile ou accumulées en mémoire pour une utilisation à court terme ?
- **Le type de stockage requis** : La persistance nécessite-t-elle un système de fichiers, un système de données volumineux ou une base de données relationnelle ?
- **Si les données sont organisées correctement** : les données sont-elles organisées de manière appropriée pour le système de stockage requis ?
- **Si les données doivent être recombinaées ou recalculées** : les données peuvent être combinées, recalculées ou agrégées avec des informations précédemment stockées, dont certaines peuvent provenir de sources non IoT.

Au fur et à mesure que le niveau 4 capture les données et les met en veille, il devient possible d'utiliser ces données par des applications en temps non réel. Les applications accèdent aux données en cas de besoin. En bref, le niveau 4 convertit les données basées sur des événements en traitement basé sur une requête. C'est une étape cruciale pour surmonter les différences entre le monde des réseaux en temps réel et le monde des applications en temps non réel [6].

Niveau 5 : Abstraction de données

Les systèmes IoT devront évoluer au niveau de l'entreprise ou même au niveau mondial et nécessiteront plusieurs systèmes de stockage pour prendre en charge les données de données IoT provenant des systèmes ERP (« Entreprise Resource Planning »), HRMS (« Human Resource Management System »), CRM (« Customer Relationship management ») et d'autres systèmes d'entreprise traditionnels. Les fonctions d'abstraction des données de niveau 5 se concentrent sur le rendu des données et leur stockage de manière à permettre le développement d'applications plus simples et améliorées [6].

Avec plusieurs périphériques générant des données, il existe de nombreuses raisons qui expliquent pourquoi ces données ne peuvent pas se retrouver dans le même magasin de données:

- Il peut y avoir trop de données à mettre dans un endroit.
- Le transfert de données vers une base de données peut consommer beaucoup de puissance de traitement, de sorte que sa récupération doit être séparée du processus de génération de données.
- Les bases de données et les entrepôts de données.
- Les appareils peuvent être géographiquement séparés et le traitement est optimisé localement.
- Les niveaux 3 et 4 peuvent séparer les « flux de données brutes continues » des « données représentant un événement ».
- Le stockage de données pour le streaming de données peut être un système de données volumineux. Le stockage des données d'événement peut être un système de gestion de base de données relationnelle avec des temps de requête plus rapides.
- Différents types de traitement de données peuvent être requis. Par exemple, le traitement en magasin se concentrera sur des choses autres que le traitement sommaire dans tous les magasins.

Pour ces raisons, le niveau d'abstraction des données doit traiter beaucoup de choses différentes. Ceux-ci incluent :

- Concilier différents formats de données provenant de différentes sources.
- Assurer une sémantique des données cohérente entre les sources.
- Vérifiez que les données sont complètes pour l'application de niveau supérieur [6].

Niveau 6 : Application

Le niveau 6 est l'endroit où l'interprétation de l'information se produit. Le logiciel à ce niveau interagit avec le niveau 5 et les données au repos, il n'a donc pas besoin de fonctionner à des vitesses réseau. Le modèle de référence IoT ne définit pas strictement une application. Les applications varient en fonction des marchés verticaux, de la nature des données de l'appareil et des besoins de l'entreprise. Par exemple, certaines applications se concentreront sur la surveillance des données de l'appareil. Certains se concentreront sur le contrôle de l'appareil. Certains vont combiner l'appareil et non les données de l'appareil. Les applications de surveillance et de contrôle représentent de

nombreux modèles d'application, modèles de programmation et différentes piles logicielles, conduisant à des discussions sur les systèmes d'exploitation, la mobilité, les serveurs d'applications, les hyperviseurs, le « multithreading », etc. Ces sujets dépassent la portée de notre discussion sur le modèle de référence IoT. Il suffit de dire que la complexité de l'application varie considérablement [6].

Les exemples incluent :

- Des applications entrepreneuriales critiques telles que des solutions ERP (« Enterprise Resource Planning ») généralisées ou des solutions sectorielles spécialisées ;
- Des applications mobiles qui traitent d'interactions simples ;
- Des rapports Business Intelligence, dans lesquels l'application est le serveur BI ;
- Des applications analytiques qui interprètent les données pour les décisions d'affaires
- Des applications de gestion de systèmes/centres de contrôle qui contrôlent le système IoT lui-même et n'agissent pas sur les données produites par celui-ci.

Si les niveaux 1-5 sont correctement architecturés, la quantité de travail requise par le niveau 6 sera réduite. Si le niveau 6 est correctement conçu, les utilisateurs pourront mieux faire leur travail [6].

Niveau 7 : Collaboration et processus

L'une des principales distinctions entre l'IoT et le système IoT est que l'IoT inclut les personnes et les processus. Cette différence devient particulièrement claire au niveau 7 (Collaboration et processus). Le système IoT et les informations qu'il crée ont peu de valeur à moins de produire une action qui nécessite souvent des personnes et des processus.

Les applications exécutent une logique d'affaires pour former les personnes. Les gens utilisent des applications et des données associées pour leurs besoins spécifiques. Souvent, plusieurs personnes utilisent la même application pour différentes raisons.

Par conséquent, l'objectif n'est plus l'application, mais de permettre aux gens de mieux faire leur travail. Les applications (niveau 6) donnent aux entrepreneurs les bonnes données, au bon moment, afin qu'ils puissent prendre la bonne décision.

Mais souvent, l'action requise nécessite plus d'une personne. Les gens devraient être capables de communiquer et de collaborer, parfois en utilisant l'Internet traditionnel, pour

rendre l'IoT utile. La communication et la collaboration nécessitent souvent plusieurs étapes. Et cela transcende généralement plusieurs applications. C'est pourquoi le niveau 7 représente un niveau supérieur à la place d'une seule application [6].

Les modèles de distribution, les standards et les études sur l'IoT et ses applications, y compris les villes et communautés intelligentes, fournissent la plateforme de normalisation spécialisée de l'IoT nécessaire à cette convergence vers un ensemble cohérent de normes internationales. Avec la participation des participants dans le domaine des technologies de l'information et de la communication, ce type d'étude aura une incidence sur la promotion du développement de « systèmes de systèmes » très efficaces qui contribueront à réduire la fracture numérique et à rendre le monde plus connecté [6].

I-5- Les Défis de l'IOT

Il y a des défis à l'application de l'Internet des objets de point de vue coût de la mise en œuvre du fait que l'attente de la technologie doit être disponible à faible coût avec un grand nombre d'objets. De plus, l'IOT est également face à de nombreux autres défis qui se résument dans cinq sous sections suivantes [7].

I-5-1- La Découverte Automatique

Dans les environnements dynamiques les services appropriés pour les objets doivent être identifiés automatiquement. Ceci nécessite des moyens sémantiques appropriés pour décrire leur fonctionnalité afin de les exploiter d'une manière efficace.

I-5-2- L'Interopérabilité

Chaque type d'objets intelligents dans l'Internet des objets possède différentes capacités de stockage d'information, de traitement et de communication. Différents objets intelligents seraient également soumis à des conditions différentes telles que la disponibilité de l'énergie et les besoins en bande passante de communications. A cet effet, pour faciliter la communication et coopération de ces objets, des normes communes sont requises.

I-5-3- La Sécurité et La Confidentialité

Les aspects de sécurité et de protection d'Internet tel que la confidentialité des communications, l'intégrité des messages ainsi que l'authenticité et la fiabilité de

partenaires de communication doivent être assurés dans un environnement IoT. A titre d'exemple dans certain cas il y a un besoin d'accéder à certains services pour accomplir une tâche ou d'empêcher de communiquer avec d'autres objets du système IoT pour des raisons préventives de sécurité. De plus, d'autres exigences selon la particularité du domaine d'application, doivent être aussi considérées comme par exemple dans les transactions commerciales des objets intelligents doivent être protégés des yeux des concurrents [7].

I-5-4- La Tolérance Aux Pannes

Les objets dans IOT sont beaucoup plus dynamiques et mobiles que les ordinateurs Internet, et dans un changement rapide et de manière inattendue. Pour structurer Internet des objets d'une manière robuste et digne de confiance on utilise la redondance en plusieurs niveaux pour l'adaptation automatique au changement de conditions.

I-5-5- L'Auto-Organisation

Les objets intelligents ne devraient pas être gérés comme des ordinateurs qui nécessitent leurs utilisateurs pour les configurer et les adapter à des situations particulières. Les objets mobiles, qui sont souvent utilisées de façon sporadique, doivent établir spontanément des liens et pouvoir être organisé et configurer par eux-mêmes en fonction de leur environnement d'exécution [7].

I-6- Les bénéfices de l'IoT

L'Internet des objets promet ainsi une forte plus-value à chaque organisme. En connectant objets, personnes et environnements, il devient possible de développer des améliorations qui ne pourront être que bénéfiques.

Les principaux bénéfices que peut avoir une entreprise dotée d'un système IoT sont [8]:

- **Amélioration de la productivité** : l'IoT permet la surveillance, le monitoring et le contrôle des différents process, ce qui optimise les différentes opérations qui augmentent la productivité et l'efficacité.
- **Analyses prédictives** : grâce à la collecte de nombreuses données, les nouvelles technologies de l'IoT permettent d'examiner les patrons récurrents et contribuent à

l'analyse prédictive qui peut être principalement utilisée en maintenance. Ces informations précises vont servir à améliorer les process et les services existants.

- **Rapidité d'action** : les données permettent de suivre en temps réel et même à distance les systèmes mis en place. Elles facilitent l'optimisation des interventions de maintenance, mais aussi donnent un avantage stratégique à l'entreprise dans le suivi de l'évolution des marchés.
- **Diminution des erreurs humaines** : grâce à la complémentarité des technologies comme l'intelligence artificielle, l'IoT permet de minorer les erreurs humaines dues à des tâches mondaines ou répétitives.

L'IoT constitue ainsi un atout dans l'aide à la prise de décision et au rendement de l'entreprise. Il représente un avantage stratégique face au marché concurrentiel. C'est une tendance qui ne cessera de croître, puisqu'on estime qu'il y aura plus de 20 milliards d'objets connectés dans le monde d'ici 2020, Face à leur futur prometteur, il serait temps d'en profiter et commencer à transformer son business [8].

Conclusion

Dans ce chapitre, nous avons présenté une étude détaillée sur l'internet des objets, sa définition, leurs domaine d'application, l'architecture, les défis, et enfin les bénéfices de l'IoT.

L'IoT incarne la prochaine évolution de l'Internet. Sachant que l'être humain progresse et évolue en transformant les données en informations, en connaissances et en savoir, l'IoT à le potentiel d'améliorer le monde tel que nous le connaissons. La rapidité à laquelle nous y parviendrons ne dépend que de nous.

Dans le chapitre suivant nous allons détailler les différents Protocol de l'IoT

CHAPITRE II : LES DIFFERENTES PROTOCOLES DE L'IOT

Introduction

L'internet des objets et ses protocoles sont parmi les sujets les mieux financés dans l'industrie et étudiés dans le milieu universitaire. L'évolution rapide de l'Internet mobile, la fabrication de matériel mini, micro-informatique, et la machine à machine (M2M) ont permis aux technologies IoT d'être au sommet de sujets médiatiques, ce qui implique que de grands investissements sont réalisés par l'industrie et la recherche actuellement et dans les prochaines années.

Les technologies IoT permettent à des choses ou des appareils qui ne sont pas des ordinateurs, d'agir intelligemment et de prendre des décisions de collaboration qui sont bénéfiques pour certaines applications [28].

Nous présentons dans ce chapitre des protocoles IoT fonctionnant à différentes couches de la pile réseau offerts par des organismes de normalisation. Ces normes ont été proposées au cours de la dernière demi-décennie pour répondre aux besoins actuels et aux besoins futurs de l'IoT.

II-1- Connectivité IoT

Lors de la planification d'un projet IoT, il est important de tenir compte de la manière dont l'appareil se connectera et communiquera. Cela déterminera les protocoles IoT qui s'appliqueront à celui-ci [9].

II-1-1 Connexion des appareils IOT

Dans la pile de technologies IoT, les appareils se connectent via des passerelles ou des fonctionnalités intégrées.

II-1-2 Les passerelles IOT

Les passerelles connectent les appareils IoT au cloud. Les données collectées à partir des appareils IoT transitent par une passerelle, sont prétraitées en périphérie, puis envoyées au cloud.

L'utilisation de passerelles IoT permet d'allonger l'autonomie de la batterie, de réduire la latence et de raccourcir les transmissions. Les passerelles vous permettent également de connecter des appareils sans accès direct à Internet et offrent une couche de sécurité supplémentaire en protégeant les données qui circulent dans les deux sens.

II-1-3 Connexions des appareils IOT au réseau

Le type de connectivité requis dépend de l'appareil, de sa fonction et de ses utilisateurs. En général, celui-ci est déterminé par la distance que les données doivent parcourir (courte ou longue).

II-2- Types de réseaux IOT

II-2-1- Réseaux à faible puissance et à courte portée

Ces réseaux conviennent aux maisons, bureaux et autres environnements de petite taille. Ils se prêtent à des batteries de petite taille (et dans certains cas, à des configurations sans batterie) et leur utilisation est souvent peu coûteuse.

Voici quelques exemples courants [10]:

II-2-1-1- Bluetooth

S'étant imposée dans le secteur de l'informatique et dans différents marchés de produits de consommation,



la technologie Bluetooth est un acteur incontournable

pour les télécoms de courte portée. Elle devrait apporter une contribution majeure pour les produits portables notamment, facilitant encore une fois la connexion à l'IOT même par l'intermédiaire d'un smartphone. La nouvelle technologie Bluetooth Low-Energy (BLE), qui répond désormais au nom de Bluetooth Smart, est un protocole important pour les applications IOT. Offrant une portée similaire à celle du Bluetooth, elle a été conçue pour réduire considérablement la consommation électrique.

Cependant, la technologie Smart/BLE n'a pas été vraiment conçue pour le transfert de fichiers et elle est plus adaptée aux petits blocs de données. Son intégration poussée dans les smartphones et autres appareils mobiles lui confère un indéniable avantage sur bien des technologies concurrentes dans le contexte des périphériques plus personnels. Bluetooth SIG estime que plus de 90 % des smartphones compatibles Bluetooth, notamment les modèles fonctionnant sur iOS, Android et Windows, devraient être « Smart Ready » d'ici 2018 [29].

Les circuits faisant appel aux fonctionnalités Bluetooth Smart incorporent la spécification fondamentale de Bluetooth version 4.0 ou supérieure, associent une vitesse de transmission de base à une configuration du cœur faible consommation pour un transceiver RF, une bande de base et une pile de protocole. La version 4.2 via son profil de prise en charge des protocoles Internet permettra aux capteurs Bluetooth Smart d'accéder directement à Internet via la connectivité 6LoWPAN. Cette connectivité IP permet d'utiliser l'infrastructure IP existante pour gérer les périphériques en bordure de réseau Bluetooth Smart. [10]

- Norme : la spécification fondamentale de Bluetooth 4.2
- Fréquence : 2,4 GHz (ISM)
- Portée : 50-150 m (Smart/BLE)
- Vitesses de transmission : 1 Mbit/s (Smart/BLE).

II-2-1-2- Wi-Fi

Compte tenu de l'omniprésence du Wi-Fi dans l'environnement domotique au sein de réseaux locaux, la connectivité Wi-Fi s'impose souvent comme le choix évident pour beaucoup de développeurs. Elle ne nécessite pas de longues explications,



sauf pour rappeler l'évidence, à savoir la vaste infrastructure existante, le transfert de données rapide et la possibilité de gérer de grandes quantités de données.

À l'heure actuelle, la 802.11n s'impose comme la norme Wi-Fi la plus utilisée dans le contexte privé et professionnel. Cette norme offre un débit élevé, de l'ordre de centaines

de mégabits par seconde, idéal pour les transferts de fichiers, mais peut-être trop énergivore pour la plupart des applications IoT [10].

- Norme : basée sur 802.11n (actuellement la norme la plus utilisée pour un usage privé)
- Fréquences : bandes de 2,4 GHz et 5 GHz
- Portée : environ 50 m
- Vitesses de transmission : 600 Mbit/s maximum, mais les vitesses habituelles sont plus proches de 150 Mbit/s, en fonction de la fréquence de canal utilisée et du nombre d'antennes (la dernière norme 802.11-ac devrait permettre des vitesses pouvant atteindre 500 Mbit/s à 1 Gbit/s) [10].

II-2-1-3- Z-Wave

Z-Wave est une technologie télécoms RF à faible consommation, principalement conçue pour la domotique et les produits tels que les contrôleurs de



lampe ou les capteurs. Optimisée pour une communication fiable et à faible latence de petits paquets de données avec des vitesses de transmission pouvant atteindre 100 Kbit/s, elle fonctionne dans la bande Sub-GHz et offre une totale résistance aux interférences causées par le Wi-Fi et les autres technologies sans fil dans la plage 2,4 GHz, telles que le Bluetooth ou ZigBee. Permettant de contrôler jusqu'à 232 circuits, elle est très évolutive et prend en charge les réseaux full mesh sans recours à un nœud coordinateur. Z-wave utilise un protocole plus simple que d'autres technologies, offrant un développement plus rapide et plus facile. Cela étant, elle ne peut compter que sur un seul fabricant de circuits, Sigma Designs, contre plusieurs sources pour d'autres technologies sans fil telles que ZigBee, notamment [10].

- Norme : Z-Wave Alliance ZAD12837/ITU-T G.9959
- Fréquence : 900MHz (ISM)
- Portée : 30 m
- Vitesses de transmission : 9,6 / 40 / 100 Kbit/s.

II-2-1-4- ZigBee

ZigBee dispose d'une importante base d'exploitation installée, surtout en milieu industriel. Parmi les profils ZigBee disponibles, ZigBee PRO et ZigBee Remote Control (RF4CE) reposent sur le protocole IEEE802.15.4 ;



fonctionnant à 2,4 GHz, cette technologie de réseau sans fil standard de l'industrie cible les applications nécessitant des échanges de données relativement peu fréquents à de faibles vitesses de transmission sur un espace restreint et dans une portée de 100 m (résidence ou bâtiment, par exemple).

Associant fonctionnement à faible consommation, haute sécurité, robustesse et grande évolutivité avec un nombre important de nœuds, ZigBee/RF4CE présente d'importants avantages dans les systèmes complexes. Ce protocole est par ailleurs en position idéale pour tirer le meilleur parti du contrôle et des réseaux de capteurs sans fil dans les applications M2M et IoT. La dernière version de ZigBee est la version 3.0, qui consiste essentiellement à unifier les différentes normes sans fil ZigBee en une norme unique [10].

- Norme : ZigBee 3.0 basé sur IEEE802.15.4
- Fréquence : 2,4 GHz
- Portée : 10-100 m
- Vitesses de transmission : 250 Kbit/s.

II-2-2- Liaisons sans fil à faible consommation énergétique (LPWAN)

Permet la communication sur un minimum de 500 mètres, requiert une puissance minimale et est utilisé pour la majorité des appareils IoT. Par exemple, les réseaux étendus à longue portée (LoRaWAN) relient des appareils mobiles, sécurisés et bidirectionnels fonctionnant sur batterie.

Voici quelques exemples courants [11] :

II-2-2-1- IoT 4G LTE

Compte tenu de leur capacité élevée et de leur faible latence, ces réseaux sont un excellent choix pour les scénarios IoT qui requièrent des informations ou des mises à jour en temps réel [11].

II-2-2-2- IoT 5G

La 5G est la prochaine génération Mobile Network Alliance (NGMN) qui vise des spécifications 40 fois plus rapides que la 4G tout en prenant en charge jusqu'à un million de connexions par kilomètre carré. La 5G permet déjà des applications haut débit et haut débit pour le streaming Ultra-HD (4k), la connectivité automobile sans conducteur ou les applications VR/AR].

II-2-2-3 Cat-0

Pour que les réseaux IoT basés sur le LTE réussissent, ils doivent présenter les caractéristiques suivantes :

- 1) Longue durée de vie des piles.
- 2) Faible coût.
- 3) Prise en charge d'un grand nombre d'appareils.
- 4) Couverture améliorée (meilleure pénétration du signal à travers les murs par exemple).
- 5) Spectre longue portée/large.

Le Cat-0 optimise les coûts en éliminant les fonctionnalités qui supportaient les exigences de débit de données élevé pour le Cat-1 (chaîne de réception double, filtre duplex). Alors que le Cat-1 remplace le 3G, le Cat-0 est le protocole qui prépare le terrain pour le remplacement du 2G par le Cat-M comme option la moins chère [11].

II-2-2-4 CAT-1

Le Cat-1 est la seule option d'IOT cellulaire entièrement disponible pour le moment et représente un premier pas vers la connexion des dispositifs IoT à l'aide des réseaux LTE existants. Bien que les performances soient inférieures à celles des réseaux 3G, c'est une excellente option pour les applications IoT qui nécessitent une interface de navigateur ou la voix. Le principal avantage est qu'il est déjà normalisé et, surtout, il est facile de passer au réseau Cat-1. Les experts prédisent qu'à mesure que les technologies 3G - et éventuellement 4G - prendront fin, les réseaux Cat-1 (et Cat-M1) prendront leur place [11].

II-2-2-5 Bande étroite

Cette norme radio fonctionne sur un sous-ensemble de la norme LTE. Elle se concentre sur la couverture intérieure, et offre des coûts réduits ainsi qu'une longue autonomie de la batterie.

II-2-2-6 Cat-M1/Cat-M/LTE-M

Cat-M (officiellement connu sous le nom de LTE Cat-M1) est souvent considéré comme la deuxième génération de puces LTE construites pour les applications de l'IoT. Elle complète la réduction des coûts et de la consommation d'énergie pour laquelle Cat-0 avait ouvert la voie à l'origine. En plafonnant la largeur de bande maximale du système à 1,4 MHz (par opposition aux 20 MHz de Cat-0), Cat-M a des cas d'utilisation spécifiques pour les applications LPWAN comme les compteurs intelligents, dans lesquels seule une petite quantité de transfert de données est nécessaire.

Mais le véritable avantage de Cat-M par rapport aux autres options réside dans ce fait : Cat-M est compatible avec le réseau LTE existant. Pour les opérateurs tels que Verizon et AT&T, c'est une excellente nouvelle car ils n'ont pas à dépenser d'argent pour construire de nouvelles antennes, bien que le maillage de Cat-M dans les réseaux LTE nécessite un correctif logiciel. Les clients actuels de Verizon et d'AT&T concluront très probablement que Cat-M est de loin l'option la plus avantageuse. Enfin, il est presque certain que les technologies 5G et LTE coexisteront pendant une bonne partie des années 2020, de sorte que la rétrocompatibilité de Cat-M est un atout [11].

II-2-2-7 NB-IOT/Cat-M2

Le NB-IoT (également appelé Cat-M2) a un objectif similaire à celui du Cat-M ; cependant, il utilise la modulation DSSS au lieu des radios LTE. Par conséquent, le NB-IoT ne fonctionne pas dans la bande LTE, ce qui signifie que les fournisseurs ont un coût initial plus élevé pour déployer le NB-IoT.

Néanmoins, l'IoT-NB est présenté comme l'option potentiellement la moins coûteuse, car il élimine le besoin d'une passerelle. D'autres infrastructures ont généralement des passerelles qui regroupent les données des capteurs, qui communiquent ensuite avec le serveur principal. (Voici un exemple plus détaillé des passerelles). Avec le NB-IoT, cependant, les données des capteurs sont envoyées directement au serveur primaire. C'est pourquoi Huawei, Ericsson, Qualcomm et Vodafone investissent activement dans les applications commerciales de NB-IoT [11].

II-2-2-8 Sigfox

Autre technologie à longue portée, Sigfox s'insère entre les technologies Wi-Fi et cellulaire en termes de portée. Elle utilise les bandes ISM, qui peuvent être utilisées gratuitement sans licences, pour transmettre des données sur un spectre très étroit, à partir et à destination d'objets connectés. Sigfox repose sur l'idée suivante : pour de nombreuses applications M2M fonctionnant sur une petite batterie et se contentant de transferts de données très limités, la portée du Wi-Fi est trop courte, tandis que celle de la technologie cellulaire est trop coûteuse et énergivore. Faisant appel à la technologie UNB (Ultra Narrow Band - À bande très étroite), Sigfox est uniquement conçu pour gérer des vitesses de transfert de données assez basses, de 10 à 1 000 bits par seconde. Sa consommation est limitée à 50 microwatts seulement, contre 5 000 microwatts pour la communication cellulaire ; ce protocole peut aussi offrir une durée de veille type de 20 ans avec une batterie 2,5 Ah, contre 0,2 an seulement pour la communication cellulaire.



Déjà activé pour des dizaines de milliers d'objets connectés, le réseau est en cours de déploiement dans les grandes villes d'Europe, dont dix au Royaume-Uni. Associant robustesse, efficacité et évolutivité, le réseau permet de connecter des millions de circuits alimentés par batterie sur des zones de plusieurs kilomètres carrés. Cette distance est idéale pour différentes applications M2M, qui devraient comprendre les objets suivants : compteurs intelligents, moniteurs de patients, dispositifs de sécurité, éclairage de rue et capteurs d'environnement. Le système Sigfox utilise du silicium, avec notamment les transceivers sans fil EZRadioPro de Silicon Labs, qui offrent des résultats excellents en sans-fil, une portée étendue et une consommation ultra-faible pour les applications réseau sans fil fonctionnant dans la bande Sub-GHZ [10].

- Norme : Sigfox
- Fréquence : 900 MHz
- Portée : 30-50 km (environnements ruraux), 3-10 km (environnements urbains)
- Vitesses de transmission : 10-1 000 bit/s.

II-2-2-9 LoRaWAN

Également similaire à certains égards à Sigfox et à Neul, LoRaWAN cible les applications de réseau étendu (WAN) et est conçu pour fournir des réseaux étendus à faible consommation dont les fonctionnalités sont



indispensables à la communication bidirectionnelle sécurisée mobile à faible coût dans les applications urbaines et industrielles intelligentes, ainsi que IoT et M2M. Optimisées pour une faible consommation et prenant en charge de grands réseaux comprenant plusieurs millions de circuits, les vitesses de transmission vont de 0,3 Kbit/s à 50 Kbit/s. [10]

- Norme : LoRaWAN
- Fréquence : variable
- Portée : 2-5 km (environnement urbain), 15 km (environnement suburbain)
- Vitesses de transmission : 0,3-50 Kbit/s.

II-3- Protocoles IoT

Les appareils IoT communiquent à l'aide des protocoles IoT. Le protocole IP (Internet Protocol) est un ensemble de règles qui détermine la façon dont les données sont envoyées à Internet. Les protocoles IoT garantissent que les informations d'un appareil ou d'un capteur seront lisibles et compréhensibles par un autre. Compte tenu de la diversité des appareils IoT disponibles, il est important d'utiliser le bon protocole dans le bon contexte.

Le type de protocole IoT à utiliser dépend de la couche de l'architecture système sur laquelle nos données doivent circuler. Le modèle OSI (Open Systèmes Interconnexion) fournit une carte des différentes couches qui envoient et reçoivent des données. Chaque protocole de l'architecture du système IoT permet une communication d'appareil à appareil, d'appareil à passerelle, de passerelle à centre de données ou de passerelle à cloud, ainsi qu'une communication entre centres de données [9].

II-3-1- Couche Application

La couche Application sert d'interface entre l'utilisateur et l'appareil.

II-3-1-1 AMQP (Advanced Message Queuing Protocol)

AMQP Est un Protocole de couche d'application standard ouvert pour l'IOT se concentrant sur des environnements axés sur les messages. Il requiert un protocole de transport sécurisé comme



TCP pour échanger des messages. Il prend en charge une communication fiable via des primitives de garantie de livraison de messages, en définissant un protocole au niveau du fil, les implémentations AMQP peuvent inter opérer entre elles. Les communications sont traitées par deux composants principaux :

- Échanges et files d'attente de messages. Les échanges sont utilisés pour acheminer les messages vers les files d'attente appropriées.
- Le routage entre les échanges et les files d'attente des messages repose sur certaines règles et conditions prédéfinies. Les messages peuvent être stockés dans les files d'attente, puis envoyés au récepteur par la suite. AMQP prend également en charge le modèle de communication publié [12].

II-3-1-2- CoAP (Constrained Application Protocol)

C'est un protocole de la couche application pour les applications IOT. Il définit un protocole de transfert Web Basé sur les fonctionnalités HTTP, est lié à UDP (et non TCP) par défaut qui le Rend plus approprié pour les applications IOT.



En outre, CoAP modifie certaines fonctionnalités HTTP pour répondre aux exigences de l'IOT telles que la faible consommation d'énergie et le fonctionnement en présence de liens à perte et bruyants.

CoAP a été conçu sur la base de REST qui représente un moyen plus simple d'échanger des données entre les clients et les serveurs via HTTP. REST Peut être considéré comme un protocole de connexion qui repose sur une architecture sans serveur apatriide, un protocole est dit « apatriide » quand aucune information de session n'est conservée par le récepteur qui est généralement un serveur. REST est utilisé dans les applications des réseaux sociaux et mobiles et élimine l'ambiguïté en utilisant les méthodes HTTP get, post, put et delete. Il permet aux clients et aux serveurs d'exposer et de consommer des services Web comme le protocole d'accès aux objets simples (SOAP), mais de manière plus simple en utilisant les

Chapitre II : Les différents protocoles de l'IOT

identificateurs de ressources uniformes (URI). CoAP vise à permettre à de minuscules appareils à faible puissance, le calcul et les capacités de communication à utiliser les interactions RESTful. Avec CoAP, les interactions entre services web de l'Internet des PC et de l'Internet des Objets deviennent bien plus simples à réaliser, une passerelle applicative assez légère (correspondance entre les commandes REST et CoAP) se charge de l'adaptation d'un monde à l'autre [12].

II-3-1-3- DDS (Data Distribution Service)

Parmi les protocoles de l'Internet des objets, les protocoles de messagerie IoT - DDS ou service de distribution de données sont un standard pour une communication machine à machine haute performance,



extensible et en temps réel. Le service de distribution de données DDA est développé et conçu par OMG ou Object Management Group. Avec l'aide de DDS, vous pouvez transférer des données à la fois dans les appareils à faible encombrement et avec les plates-formes Cloud.

Le service de distribution de données comprend deux couches importantes. Ce sont le DCPS et le DLRL. Le DCPS ou Data-Centric Publish-Subscribe fonctionne en fournissant des informations aux abonnés. Le DLRL ou Data-Local Reconstruction Layer fait son travail en fournissant une interface aux fonctionnalités Data-Centric Public-Subscribe [12].

II-3-2-Couche Transport

La couche Transport permet la communication et protège les données lorsqu'elles circulent entre les couches.

II-3-2-1- TCP (Transmission Control Protocol)

Le TCP, pour Transmission Control Protocol, désigne un protocole de transmission utilisé sur les réseaux IP. Il est, dans le modèle Internet, rarement cité seul, mais bien souvent en compagnie de l'acronyme IP pour former la suite de protocoles TCP/IP. Développé au début des années 1970, le protocole TCP fait l'objet d'une description détaillée dans la RFC 793 de l'IETF (un document décrivant officiellement les aspects techniques d'Internet). Il est basé sur un fonctionnement en trois temps. Il établit d'abord une connexion, transfère ensuite les données, et met enfin un terme à la connexion. Toutes ces opérations sont

réalisées dans un environnement sécurisé, fiable et robuste, notamment grâce à l'utilisation de systèmes de séquençage [13].

II-3-2- UDP (User Datagram Protocol)

L'UDP, pour User Datagram Protocol, désigne un protocole de communication utilisé sur Internet, défini par la RFC 768 de l'IETF, l'UDP permet la transmission de données entre deux entités avec une grande facilité, chacune d'entre elles possédant une adresse IP propre et un numéro de port.

Différent du protocole TCP, l'UDP n'est toutefois pas en mesure de garantir que les données transmises (les datagrammes) sont bel et bien arrivées à destination. C'est la raison pour laquelle on l'utilise principalement pour la transmission de faibles volumes de données, ou bien lorsqu'on juge que la perte de données représente un risque très limité. Typiquement, les jeux en ligne sont basés sur ce genre de protocole dans leur mode de transmission des données [13].

II-3-3 Couche Réseau

La couche Réseau permet à des appareils individuels de communiquer avec le routeur.

II-3-3-1- 6LoWPAN

6LoWPAN (IPv6 Low-power wireless Personal Area Network) est un protocole standard pour réaliser une communication IPv6 sur des réseaux sans fil composés de modules sans fil de



faible puissance. La spécification 6LoWPAN inclut la compression de paquets et d'autres mécanismes d'optimisation, de sorte que les paquets IPv6 puissent être efficacement transmis sur des réseaux à faible puissance et avec perte, ce qui rend la communication IPv6 pratique sur ces réseaux contraints.

De nombreux protocoles pour ces réseaux sans fil de faible puissance ont été proposés avant la naissance du 6LoWPAN, 6LoWPAN étant considéré comme le meilleur protocole pour réaliser l'IoT (Internet of Things). En effet, la communication 6LoWPAN est basée sur IPv6, ce qui permet de créer des services intelligents qui ne pouvaient pas être réalisés auparavant, lorsque les réseaux basse consommation étaient fermés et déconnectés à Internet [10].

II-3-3-2-IPv6

le protocole IPv6 apporte plus de liberté pour la gestion des adresses, et permet d'offrir plus de flexibilité en termes de gestion de trafic et de sécurité des échanges sur Internet, le protocole IPv6 offre un espace d'adressage presque infini avec une adresse IPv6 codée sur 128 bits



face à 32 bits pour l'IPv4, aussi la simplification du format des messages constitue un apport majeur par rapport à l'IPv4 : un message, communément appelé paquet ou datagramme, est composé de 2 parties : l'entête et le contenu (les données utiles), dans sa version 4, l'entête du message a une taille fixe (20 octets mais la longueur variable des champs optionnels y ajoute du poids), alors que dans sa version 6, celle-ci a été scindée en deux parties: une fixe et une variable, par le biais de son entête variable, IPv6 offre un support étendu à toutes extensions ou options pouvant être nécessaires, en d'autres termes, pour ne pas avoir un entête trop long et trop lourd à traiter, une série de champs des messages IPv4 ont été mis en option dans des entêtes d'extension et c'est l'émetteur qui choisit quelles extensions à inclure dans le message, d'autre part, l'IPv6 se base sur l'optimisation des échanges des données avec des messages plus courts, ce qui permet un gain appréciable en bande passante, dans un réseau IPv6, l'émetteur n'utilise que les extensions qu'il estime utiles, ce qui donne un coût réduit de traitement au niveau des routeurs, en plus les entêtes d'extension ne sont pas examinés par les nœuds intermédiaires le long du chemin vers la destination offrant ainsi une meilleure gestion de la qualité du service [10].

II-3-4- Couche Liaison de données

La couche liaison de données transfère les données au sein de l'architecture système, en identifiant et en corrigeant les erreurs trouvées dans la couche physique.

II 3-4-1- IEEE 802.15.4

Norme radio relative aux connexions sans fil à faible consommation. Elle est utilisée avec ZigBee, 6LoWPAN et d'autres normes pour créer des réseaux sans fil incorporés.

II-3-5- Couche Physique

La couche Physique établit un canal de communication, permettant aux périphériques de se connecter dans un environnement spécifié.

II-3-5-1- Bluetooth à basse consommation (BLE)

Bluetooth Low Energy, également connu sous le nom de « Bluetooth Smart », a été développé par le groupe d'intérêt spécial Bluetooth. Il a une portée relativement plus courte et consomme moins d'énergie que les protocoles concurrents. La pile de protocoles BLE est similaire à la pile utilisée dans la technologie Bluetooth classique. BLE n'est pas compatible avec le Bluetooth classique [10].

II-3-5-2- Ethernet

Ethernet désigne un protocole de réseau local (LAN). Celui-ci se base sur des commutations de paquets et sur des câbles en paires torsadées pour permettre de relier plusieurs machines entre elles. Pour les particuliers, Ethernet se résume à ce câble branché entre un ordinateur et une box Internet, qui permet d'offrir un réseau et un débit Internet de grande qualité. Il se rapproche dans ses fonctions du réseau Wi-Fi classique. Il existe aujourd'hui une grande variété d'Ethernet allant de l'Ethernet 10 mégabits/s à l'Ethernet 10 gigabits/s. À l'échelle internationale, Ethernet est aussi connu sous la norme IEEE 802.3 [10].

II-3-5-3- NFC (Near Field Communication)

NFC (Near Field Communication) est une technologie favorisant des interactions bidirectionnelles simples et sûres entre deux dispositifs électroniques (les smartphones en particulier), pour permettre aux consommateurs d'effectuer des transactions par paiement sans contact, d'accéder à des contenus numériques et de se connecter à des dispositifs électroniques. Son action principale consiste à étendre les fonctionnalités de la technologie de carte sans contact, pour permettre aux dispositifs de partager des informations à une distance inférieure à 4 cm. Des informations complémentaires sont disponibles ici [10].



- Norme : ISO/CEI18000-3
- Fréquence : 13,56MHz (ISM)
- Portée : 10 cm
- Vitesses de transmission : 100–420 Kbit/s

II-3-5-4- Radio-identification (RFID)

La RFID est une technologie d'identification dans laquelle une étiquette RFID (une petite puce avec une antenne) transporte des données, qui sont lues par un lecteur RFID. L'étiquette transmet les données qui y sont stockées via des ondes radio.



Elle est similaire à la technologie des codes à barres. Mais contrairement à un code à barres traditionnel, il ne nécessite pas de communication en visibilité directe entre l'étiquette et le lecteur et peut s'identifier à distance même sans opérateur humain. La plage de RFID varie avec la fréquence. Il peut aller jusqu'à des centaines de mètres.

Les étiquettes RFID sont de deux types : actives et passives. Les balises actives ont une source d'alimentation et les balises passives n'ont aucune source d'alimentation. Les étiquettes passives tirent leur énergie des ondes électromagnétiques émises par le lecteur et sont donc bon marché et ont une longue durée de vie.

La technologie RFID est utilisée dans diverses applications telles que la gestion de la chaîne d'approvisionnement, le contrôle d'accès, l'authentification d'identité et le suivi d'objets. L'étiquette RFID est attachée à l'objet à suivre et le lecteur détecte et enregistre sa présence au passage de l'objet. De cette manière, le mouvement des objets peut être suivi et la RFID peut servir de moteur de recherche pour des choses intelligentes.

Les données de bas niveau collectées à partir des étiquettes RFID peuvent être transformées en informations de plus haut niveau dans les applications IoT. Il existe de nombreux outils de niveau utilisateur disponibles, dans lesquels toutes les données collectées par des lecteurs RFID particuliers et les données associées aux étiquettes RFID peuvent être gérées. Les données de haut niveau peuvent être utilisées pour tirer des inférences et prendre d'autres mesures [14].

Quel est le meilleur protocole ?

Certains protocoles sont mieux adaptés que d'autres pour remplir ce besoin, le protocole MQTT est actuellement considéré comme l'un des candidats les plus sérieux pour assurer le transport des données au sein des architectures IoT.

Donc nous avons considéré ce protocole comme étant à la fois suffisamment simple et intéressant pour constituer l'objet de notre étude.

II-4-Le protocole MQTT

II-4-1- Définition

MQTT (Message Queue Telemetry Transport) est un protocole M2M (Machine-to-Machine). C'est un protocole de messagerie basé sur le publish/subscribe à la fois extrêmement simple et léger idéal pour l'Internet des objets. Il a été inventé par Andy Stanford-Clark (IBM) et Arlen Nipper (Arcom, maintenant Cirrus Link) en 1999, lorsque leur cas d'utilisation était de créer un protocole avec les objectifs suivants :

- 1- Simple à mettre en œuvre.
- 2- Fournir une qualité de service de livraison de données.
- 3- Léger et faible consommation de la bande passante.

Par la suite, la version 3.1 a été libérée de droits en 2010, et le protocole est standardisé par l'OASIS en 2014 dans sa version 3.1.1. Il devient un standard ISO (ISO/IEC 20922) en 2016 [15].

Aujourd'hui, avec l'essor de l'IOT, MQTT suscite un intérêt grandissant, dont on peut se rendre compte en observant l'évolution du nombre de recherches pour MQTT.

Les qualités de MQTT font de lui un excellent candidat pour les communications au sein de l'Internet des Objets, mais il a également été utilisé avec succès par Facebook pour être au cœur de leur système de messagerie à cause de sa performance, comme il est utilisé par Amazon web services dans ses passerelles pour permettre la diffusion des données à plusieurs abonnés sur un sujet donné.

La norme MQTT décrit le comportement attendu des brokers et clients MQTT, ainsi que les détails du format binaire employé pour les échanges [15].

II-4-2- Mode de fonctionnement MQTT

MQTT est un protocole de publication et de souscription de message. Les clients ne communiquent pas directement entre eux, ils publient des messages sur un broker (appelé aussi courtier), les messages sont composés d'un contenu et d'un sujet (topic). Le broker garde en mémoire le dernier message pour chaque sujet. Les clients qui sont intéressés par les messages d'un sujet peuvent les récupérer en se connectant au broker. L'avantage de cette solution est qu'elle permet à plusieurs clients de communiquer même s'ils ne sont jamais connectés en même temps au broker. Ce modèle consiste à organiser les messages

par sujets (topics) et à gérer leur distribution selon le principe d'abonnement. Ainsi, toute application peut publier ses messages sur les topics de son choix, tandis que les applications intéressées par les messages d'un topic donné peuvent s'abonner pour recevoir tous les nouveaux messages publiés sur ce topic [16].

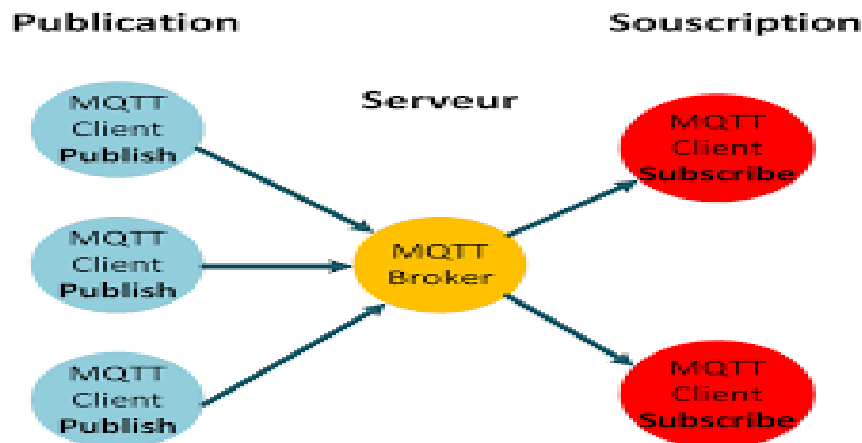


Figure II-1 Modèle pour transférer des données entre deux client (publish / subscribe) à travers le broker

II-4-3- Abonnements et Publications

Chaque message publié est nécessairement associé à un sujet, ce qui permet sa distribution aux abonnés. Les sujets peuvent être organisés en hiérarchie arborescente, ainsi les abonnements peuvent porter sur des motifs de filtrage.

La gestion des abonnements est très simple et consiste en trois commandes essentielles :

- 1. SUBSCRIBE** : Permet à un client de s'abonner à un topic, une fois abonné il recevra par la suite toutes les publications concernant ce sujet. Un abonnement définit également un niveau de qualité de service. La bonne réception de cette commande est confirmée par le broker par un SUBACK portant le même identifiant de paquet.
- 2. UNSUBSCRIBE** : Donne la possibilité d'annuler un abonnement, et ainsi ne plus recevoir les publications ultérieures. La bonne réception de cette commande est confirmée par le broker par un UNSUBACK portant le même identifiant de paquet.
- 3. PUBLISH** : Initié par un client, permet de publier un message qui sera transmis par le broker aux abonnés éventuels. La même commande sera envoyée par le broker aux abonnés pour délivrer le message. Si la qualité de service requise est supérieure à zéro, des messages seront échangés pour confirmer la prise en charge de la publication [17].

II-4-4 Les topics

Les clients de souscription MQTT s'enregistrent auprès du broker sur des topics, des sortes de chemins d'accès à une ressource. Ils demandent ainsi à être notifiés lorsque quelqu'un publie sur ces topics [18].

Cela peut être un topic de température par exemple : `/sensor/1/temperature`.

On peut souscrire à un ensemble de topics en utilisant des wildcards `#` ou `+`.

Par exemple, si un client publie sur les topics `/sensor/1/temperature` et `/sensor/1/humidity`, un autre client peut écouter ces deux topics à la fois : `/sensor/1/#`.

Si plusieurs clients publient leurs températures et humidités en intercalant leur numéro de client sur leur topic, un autre client peut écouter toutes les températures ainsi : `/sensor+/temperature`. Il recevra alors les températures du client 1 (`/sensor/1/temperature`), et du client 2 (`/sensor/2/temperature`), etc [18].

II-4-5- Qualité de service

Trois niveaux de qualité de service (QoS) sont définis pour la publication des messages :

1. QoS 0 « Livraison une fois maximum ».

Les messages sont envoyés en fonction des capacités maximum du réseau TCP/IP sous-jacent. Aucune réponse n'est attendue. Aucune sémantique de relance n'est définie dans le protocole. Par conséquent, le message ne parvient pas du tout ou une seule fois au broker de destination. Le niveau QoS 0 est également connu comme `fire and forget`.

2- QoS 1 « Livraison au moins une fois ».

L'arrivée d'un message QoS 1 au broker est reconnue. En cas d'échec identifiable de la liaison ou de l'unité d'envoi, ou bien après une certaine période de temps sans réception du message d'accusé de réception, l'expéditeur envoie à nouveau une copie du message. Par conséquent, le message est sûr d'arriver, mais il peut le faire plusieurs fois.

3/-QoS 2 « Livraison exactement une fois ».

Pour le niveau QoS 2, d'autres flux de protocoles sont utilisés au-delà de QoS 1 pour que des messages ne soient pas envoyés en double à l'application de réception. Il s'agit du niveau de service le plus élevé qui sert lorsque des messages en double ne sont pas

appropriés. Il existe évidemment des conséquences en matière de trafic réseau, mais cet impact est souvent acceptable sachant l'importance du contenu du message.

La qualité de service peut être sélectionnée message par message, ce qui permet la publication des messages d'importance mineure avec le niveau QoS 0 et la distribution des messages plus importants avec QoS 2.

Ces niveaux sont mis en œuvre par des échanges supplémentaires entre l'expéditeur et le récepteur, et plus la qualité demandée est élevée, plus il faudra d'échanges pour valider une publication comme illustré dans la figure. Pour tous les niveaux supérieurs à zéro, un identifiant est associé au message pour permettre son suivi, MQTT prévoyant la possibilité d'avoir au plus 65535 messages en attente (l'identifiant de message tenant sur 16 bits) [19].

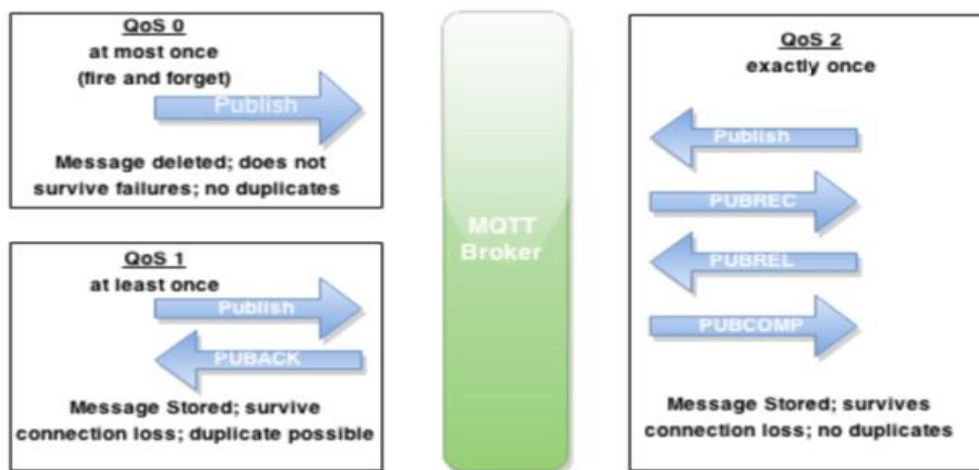


Figure II-2 Niveaux de Qualité de Service MQTT

II-4-6- Sécurité

Trois concepts sont fondamentaux pour la sécurité MQTT : identification, authentification et autorisation.

- **L'identification** : consiste à nommer le broker et le client auquel on donne des droits d'accès.
- **L'authentification** : cherche à prouver mutuellement l'identité du client et du broker.
- **L'autorisation** : consiste à gérer les droits du client. Le broker MQTT s'identifie auprès du client avec son adresse IP et son certificat numérique. Le client MQTT utilise le protocole SSL pour authentifier le certificat envoyé par le broker. Un client authentifie un broker à l'aide du protocole SSL. Un broker MQTT authentifie un client à l'aide du

Chapitre II : Les différents protocoles de l'IOT

protocole SSL, d'un mot de passe, ou des deux. L'autorisation ne fait pas partie du protocole MQTT. Elle est fournie par les brokers MQTT. Ce qui est autorisé ou non dépend de ce que fait le broker [20].

Conclusion

Dans ce chapitre nous avons présenté les types de réseaux pour l'IOT, et les différentes familles de protocole est que chacun d'elles réponde à un besoin spécifique dans l'IOT, cela permet de sélectionner le bon protocole en fonction des objets connectés utilisés dans notre écosystème IOT.

Nous avons vu que MQTT est un protocole tout à la fois minimaliste et très simple à mettre en œuvre (en particulier du côté client). Malgré tout, il reste suffisamment souple et extensible pour suffire aux besoins d'un bon nombre d'applications, et est donc particulièrement adapté au contexte de l'IoT.

C'est pour ces raisons que nous choisissons ce protocole dans la conception de notre plateforme.

CHAPITRE III : IMPLEMENTATION ET REALISATION

Introduction

Dans ce chapitre, nous allons présenter deux parties principales : la première partie concerne l'environnement de travail et la deuxième partie présente les différentes étapes pour la réalisation de notre projet.

III-1- L'environnement de travail

III-1-1- Logiciel Arduino

Un environnement de développement intégré fonctionnant sur divers systèmes d'exploitation (Windows, Mac Os, Gnu/Linux) qui permet d'éditer le programme sur un ordinateur et de le transférer via le port USB.

Le langage de programmation est le C++, et lié à la bibliothèque de développement Arduino.

Ce logiciel a une simple interface composée de quatre blocs principaux (figure III-1) [21].

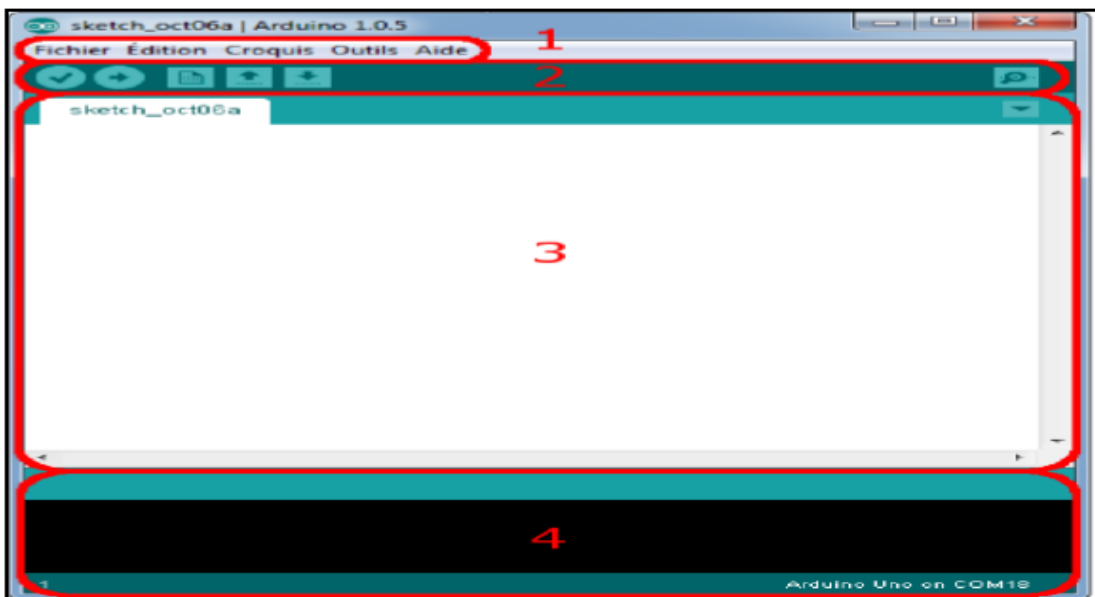


Figure III-1 Interface du logiciel IDE

1. **Menu** Les différents éléments du menu permettent de créer de nouveaux sketches (programmes), de les sauvegarder, et de gérer les préférences du logiciel et les paramètres de communication avec votre carte esp8266 Node MCU.

Le menu comprend :

- **Fichier** : pour créer, sauvegarder en spécifiant la destination, et d'appeler un programme.
- **Edition** : Pour couper, copier, coller, supprimer, sélectionner..., etc.
- **Croquis** : regroupe les fichiers réalisés.
- **Outils** : pour spécifier le type de la carte, le port série, formater, recharger et réparer l'encodage, graver la séquence d'initialisation, de la carte branchée sur l'ordinateur [22].

2- Les boutons

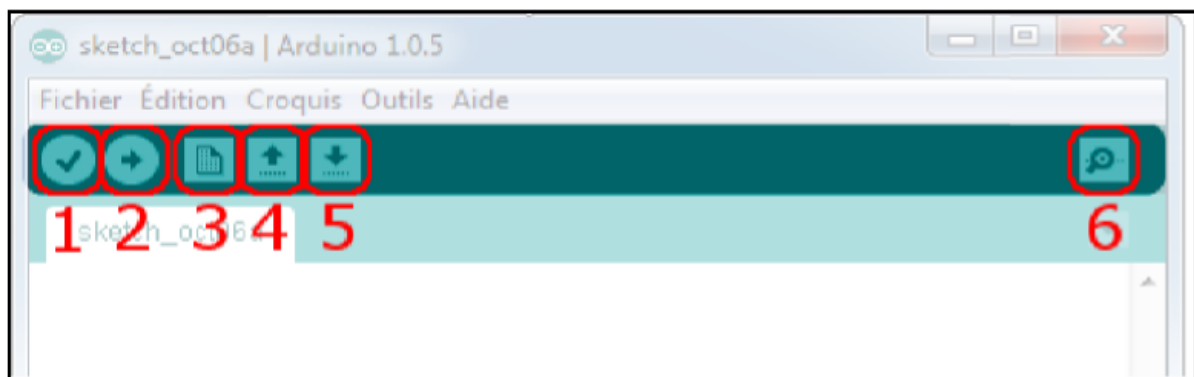


Figure III-2 Les boutons d'action

- **Bouton 1** : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme
- **Bouton 2** : Charge (téléverse) le programme dans la carte esp8266.
- **Bouton 3** : Cree un nouveau fichier.
- **Bouton 4** : Ouvre un fichier.
- **Bouton 5** : Enregistre le fichier.
- **Bouton 6** : Ouvre le moniteur série [23].

NOTE : Le moniteur série est un outil de communication entre le pc et la carte esp8266 qui sont connecté via le port USB, utilisé généralement pour tester le fonctionnement des programmes.

III-1-2- Esp8266 NodeMCU

III-1-2-1- Définition

NodeMCU est une plate-forme open source, matérielle et logicielle basée sur un SoC Wi-Fi ESP8266 fabriqué par Espressif Systems. Le terme 'NodeMCU' se réfère par défaut au firmware plutôt qu'aux kits de développement. Le firmware, permettant nativement l'exécution de scripts écrits en Lua.

L'ESP 8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi de taille réduite, permet de connecter un microcontrôleur à un réseau Wi-Fi et d'établir des connexions TCP/IP avec des commandes AT.

Il existe à ce jour plus de 12 versions de modules qui ont été construits à partir de ce composant. Chaque version est identifiée par une nomenclature sous la forme : ESP-01, ESP-02 ...ESP-12 [24].



Figure III-3 La carte NodeMCU (ESP8266)

L'ESP8266 peut se programmer de plusieurs façons

- Avec des scripts Lua, interprétés ou compilés, avec le firmware NodeMCU.
- En C++, avec l'IDE Arduino,
- En JavaScript, avec le firmware Espruino,
- En Micro Python, avec le firmware MicroPython5 ou Circuit Python.
- En C, avec le SDK d'Espressif.
- En C, avec le SDK esp-open-sdk7 basé sur la chaîne de compilation GCC.
- En Go, avec le Framework Gobot.

III-1-2-2-Description technique d'ESP8266

Les spécifications principales de l'ESP8266 sont :

- WIFI direct 802.11 b/g/n,
- Une pile de protocoles TCP/IP intégrés.

- Processeur 32 bits intégré de faible puissance.
- Il fonctionne à 3,3 V et ne possède pas de régulateur de tension, il faudra donc bien veiller à toujours l'alimenter en 3,3 V et non en 5 V ou depuis un pack de piles.
- Le processeur est cadencé à 80 MHz (contre 16 MHz pour carte Arduino UNO) et possède 80 KB de RAM (contre 2,5 KB pour une carte Arduino UNO).
- Il intègre une mémoire flash externe d'environ 400 KB (contre 32 KB pour une carte Arduino UNO) [24].

III-1-2-3- Les avantages d'ESP8266

- Pas cher.
- Environnement de programmation clair et simple (Arduino).
- Outils de développement Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, sites persos, etc.).
- Existence de « Shield » (boucliers en français) : ce sont des cartes supplémentaires [24].

III-1-2-4- Schéma interne d'ESP8266

L'ESP8266 possède plusieurs entrées-sorties comme le montre le schéma ci-dessous :

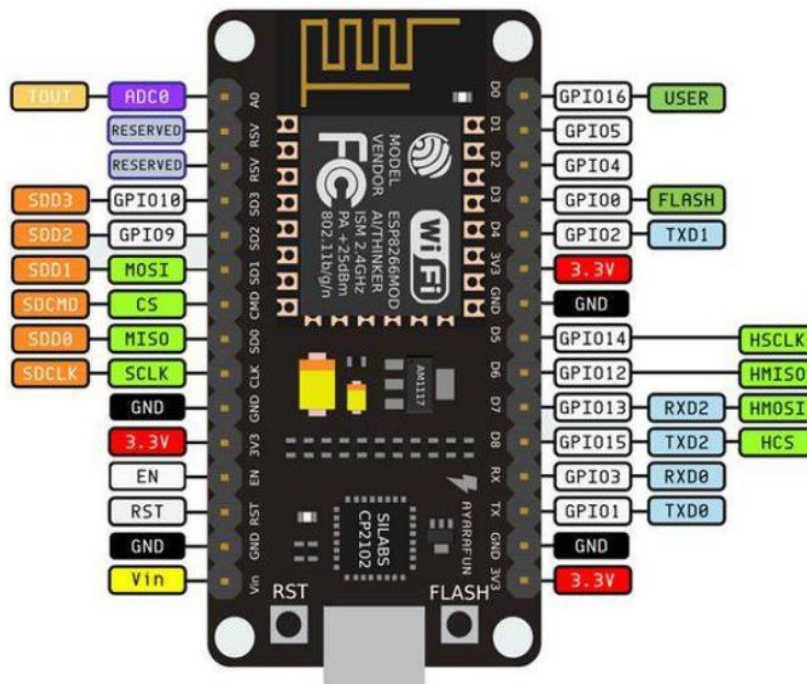


Figure III-4 Composition d'une carte NodeMCU (ESP8266)

III-1-2-5- Les différents composant d'ESP8266

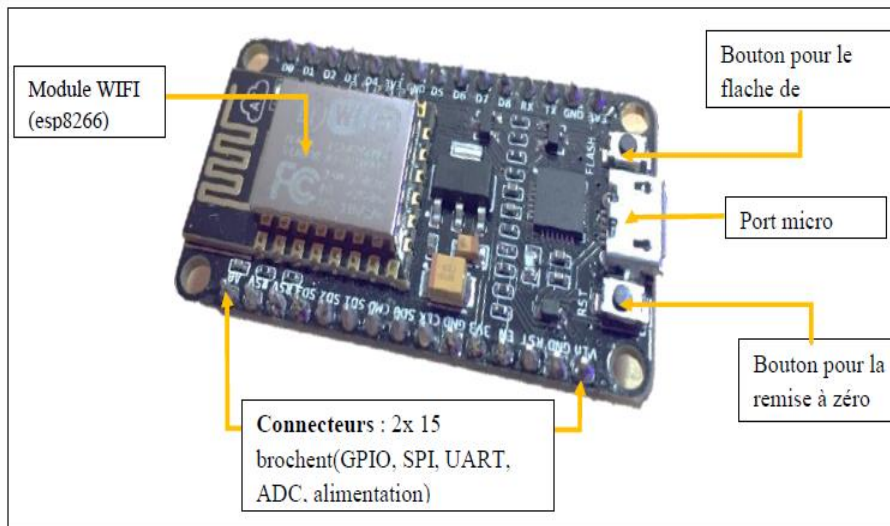


Figure III-5 Les différents composant d'esp8266

III-1-3- Support de carte NodeMCU

C'est une base sur laquelle le NodeMCU est installé et présente plusieurs avantages :

- Résolution du problème des dimensions de la carte et se passer de la plaque de fixation des parties électroniques.
- Nous nous permettons d'utiliser une source d'alimentation externe (batteries) avec une valeur de tension allant Entre 6 et 24 volts.
- Il dispose d'une prise dédiée pour connecter des sources d'alimentation (batteries).

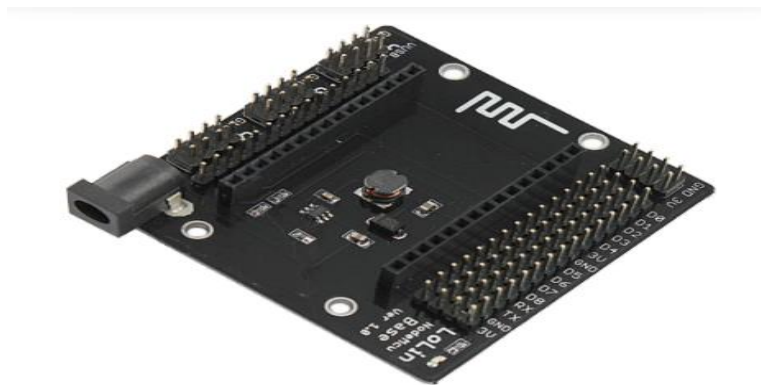


Figure III-6 Support de carte NodeMCU



Figure III-7 Le Support avec la carte NodeMCU

III-1-4- Les capteurs

III-1-4-1- Capteur Température/humidité DHT11

Dans notre système nous avons utilisé un capteur DHT11, dans lequel est intégré une thermistance pour capter la température et un capteur capacitif pour capter l'humidité.

[25]

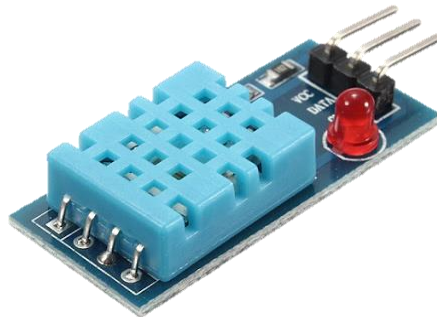


Figure III-8 Capteur DHT11

Les caractéristiques de DHT11

- Coût faible
- Alimentation 3 à 5 V
- 2.5mA utilisation maximale actuelle lors de la conversion (lors de la demande de données)
- Bon pour 20-80% de lectures d'humidité avec une précision de 5%
- Bon pour des variations de température de 0-50 ° C Précision de ± 2 ° C
- Pas plus de 1 Hz de fréquence d'échantillonnage (une fois par seconde)
- Taille du corps 15.5mm x 12mm x 5.5mm
- 3 broches avec espacement de 0,1

Le brochage du capteur est le suivant

- La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).
- La broche n°2 est la broche de communication.
- La broche n°3 est la masse du capteur (GND).

III-1-5- MQTT Dash

MQTT Dash est l'une des meilleures applications gratuites pour utiliser MQTT sur votre smartphone. Il a une interface ergonomique avec de nombreuses ressources graphiques et est très simple à utiliser.



MQTT Dash

Control the Things via MQTT protocol

III-1-6- Le broker MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie qui fonctionne sur le principe de souscription / publication qui a été développé à la base pour simplifier la communication entre les machines. Pour fonctionner il faut installer un serveur, qui est désigné par Broker.

Mosquitto est un très bon choix. Il est disponible sur toutes les plateformes et le plus utilisé dans les projets DIY [26].

1- Publier (envoyer) un message MQTT depuis le terminal :

La commande **mosquitto_pub** permet de publier simplement un message MQTT.

Voici quelques options utiles à connaître :

- h nom du broker (ou IP) sur lequel se connecter
- p port, par défaut 1883
- u nom d'utilisateur (optionnel)
- P mot de passe (optionnel)
- t topic sur lequel publié le message
- m message (payload) à envoyer. Le payload peut être une valeur ou un JSON.
- r indique au broker de retenir (ou pas) le message

2-Recevoir (souscrire) des messages :

La commande **mosquitto_sub** permet de souscrire, c'est à dire de recevoir des messages publiés sur un broker MQTT. Les options sont presque identiques à **mosquitto_pub**.

III-1-7- Les technologies web

Les différentes technologies web utilisés pour la création de l'interface web sont :

III-1-7-1 HTML (Hyper Text Markup Language)

L'HTML est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du Standard Generalized Markup Language (SGML) [27].



III-1-7-2 PHP (Personal Home Pages)

PHP est un langage de programmation web côté serveur, ce qui veut dire que c'est le serveur qui va interpréter le code PHP et générer du code qui pourra être interprété par votre navigateur.



Pour décrire une page PHP, on pourrait dire que c'est un fichier avec l'extension **PHP**, lequel contient une combinaison de balises HTML et de scripts qui tournent sur un serveur web [27].

III-1-7-3 MYSQL

MySQL est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels [27].



III-1-7-4 Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le



domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [27].

III-2- La Réalisation

III-2-1- Le montage de Projet Réalise :

Le schéma suivant représenté le branchement du capteur de température et d'humidité DHT11 et une LED avec la carte esp8266.

Le capture température et humidité se compose de 3 broches : la file verte est pour l'échange de donne, le rouge et le noire sont pour l'alimentation. La LED se compose de 2 broches : le fil rouge est pour l'échange de donnée, et la noire pour l'alimentation.

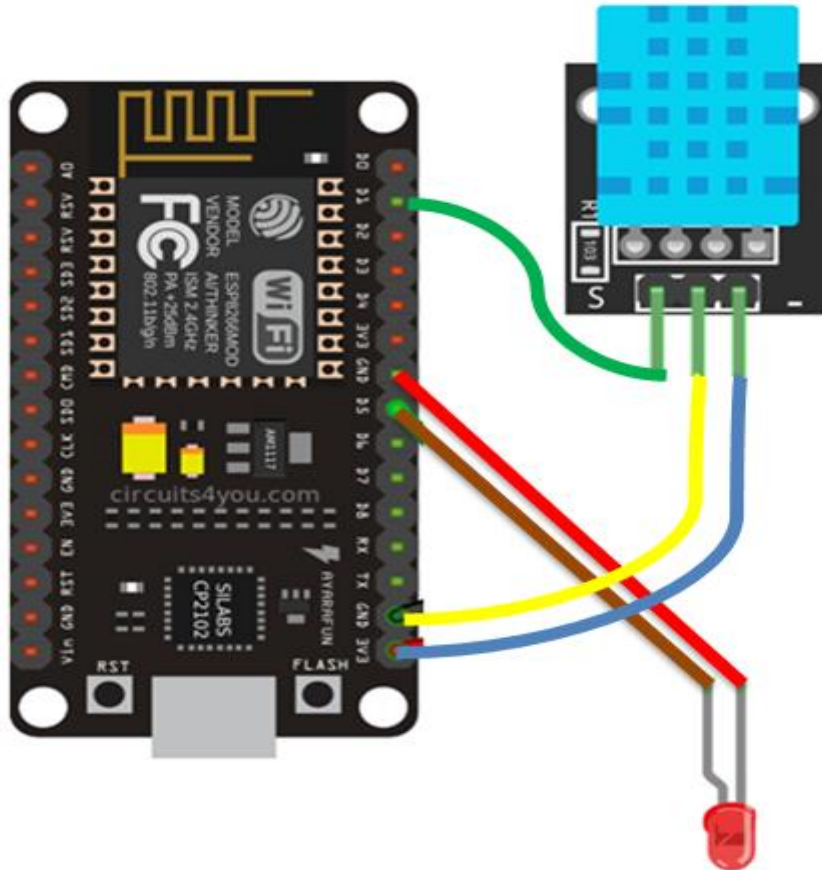
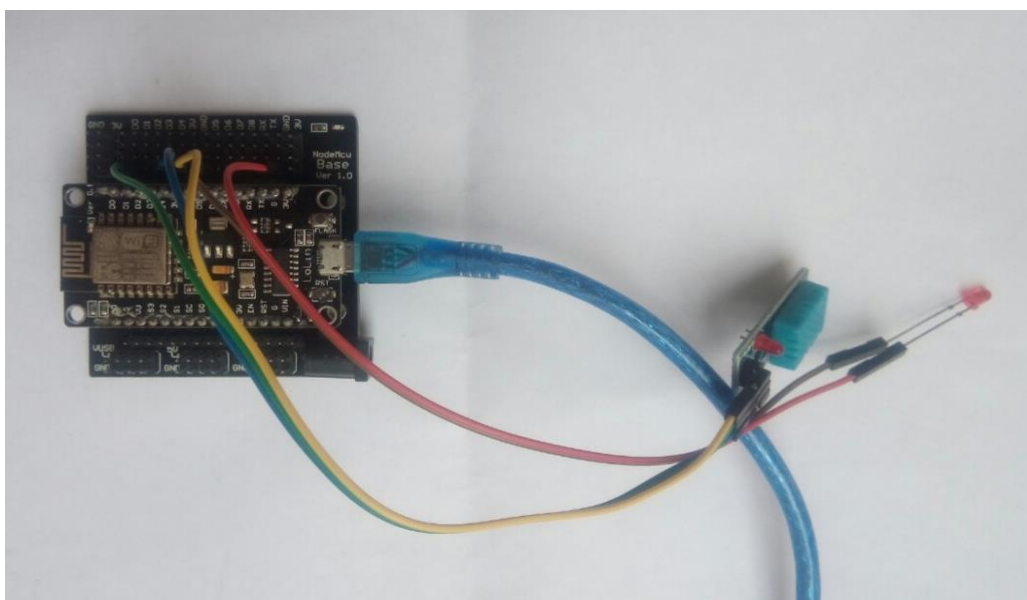


Figure III-9 Schéma de câblage de capture DHT11 et la LED avec la carte ESP8266.

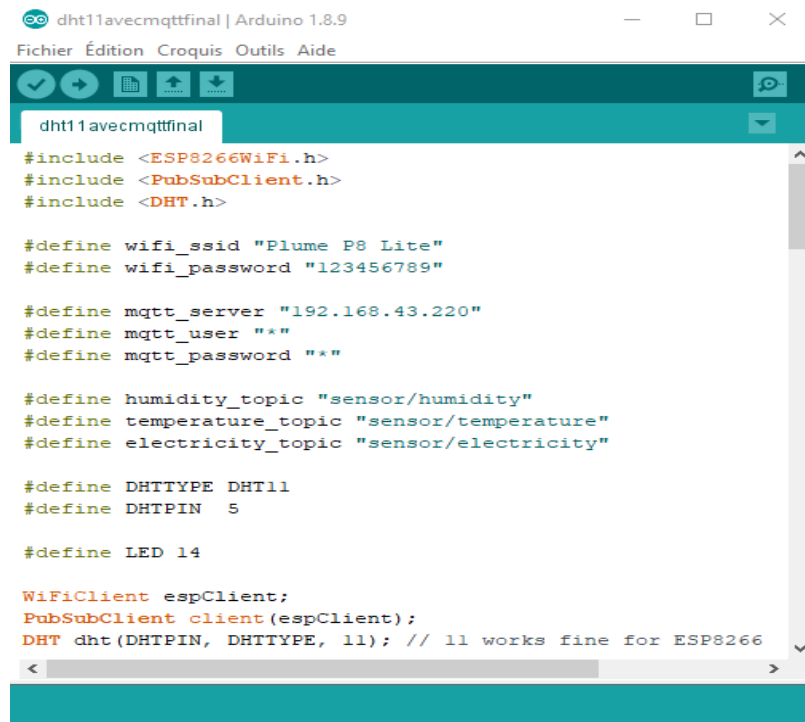
Le schéma suivant représenté le branchement de différent matériel utilise (le capteur DHT11 et une LED) avec la carte esp8266.



III-10 Le montage de projet réalisé

III-2-2- La Programmation de ESP8266 NodeMCU :

Après la création de l'application on va programmer l'esp8266 NodeMCU via l'Arduino et en utilise le programme suivant détaillé



```
dht11avecmqttfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

dht11avecmqttfinal
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>

#define wifi_ssid "Plume P8 Lite"
#define wifi_password "123456789"

#define mqtt_server "192.168.43.220"
#define mqtt_user "*"
#define mqtt_password "*"

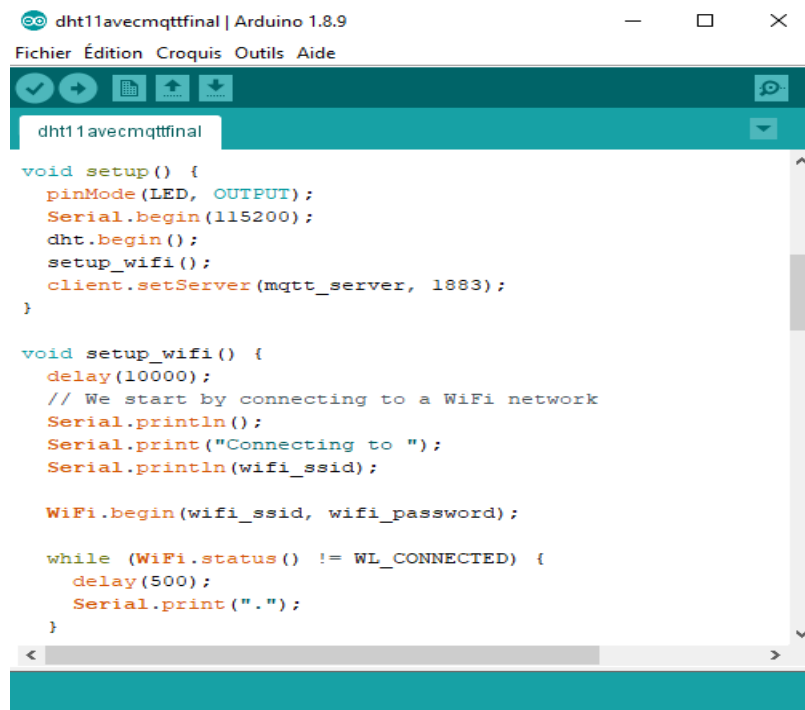
#define humidity_topic "sensor/humidity"
#define temperature_topic "sensor/temperature"
#define electricity_topic "sensor/electricity"

#define DHTTYPE DHT11
#define DHTPIN 5

#define LED 14

WiFiClient espClient;
PubSubClient client(espClient);
DHT dht(DHTPIN, DHTTYPE, 11); // 11 works fine for ESP8266
```

Figure III-11 Définition des paramètres (wifi, adresse IP, les topics et les pins liés au capteur et la LED)



```
dht11avecmqttfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

dht11avecmqttfinal

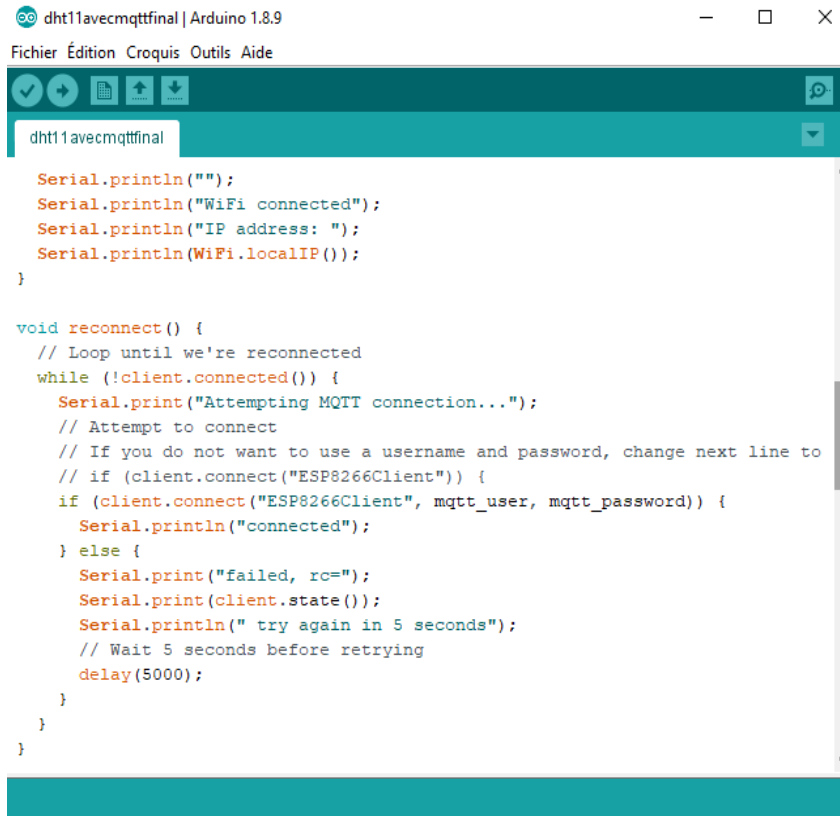
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(115200);
  dht.begin();
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void setup_wifi() {
  delay(10000);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(wifi_ssid);

  WiFi.begin(wifi_ssid, wifi_password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}
```

Figure III-12 Connecter à un réseau wifi

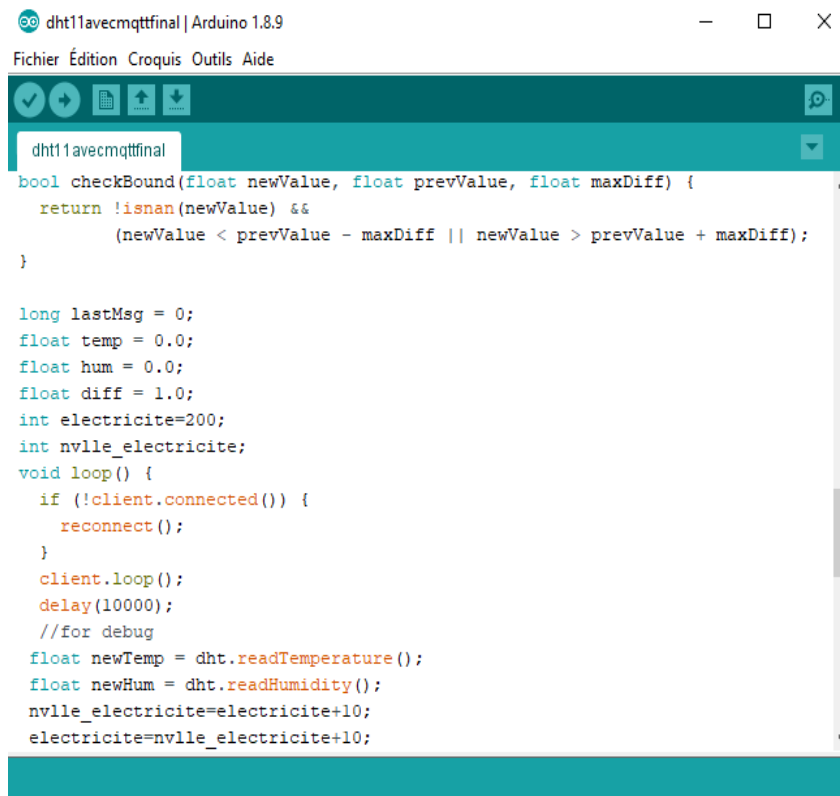


```
dht11avecmqttfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    // If you do not want to use a username and password, change next line to
    // if (client.connect("ESP8266Client")) {
    if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

Figure III-13 Reconnection a mqtt en cas de déconnexion

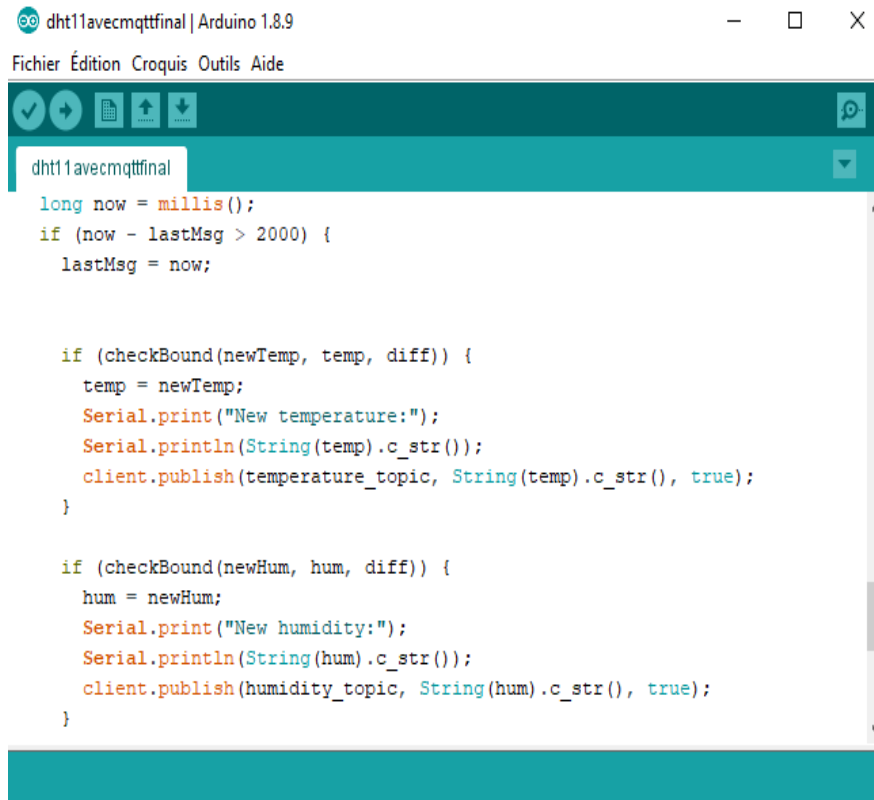


```
dht11avecmqttfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

bool checkBound(float newValue, float prevValue, float maxDiff) {
  return !isnan(newValue) &&
    (newValue < prevValue - maxDiff || newValue > prevValue + maxDiff);
}

long lastMsg = 0;
float temp = 0.0;
float hum = 0.0;
float diff = 1.0;
int electricite=200;
int nvlle_electricite;
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  delay(10000);
  //for debug
  float newTemp = dht.readTemperature();
  float newHum = dht.readHumidity();
  nvlle_electricite=electricite+10;
  electricite=nvlle_electricite+10;
}
```

Figure III-14 Lecture des valeurs du capteur



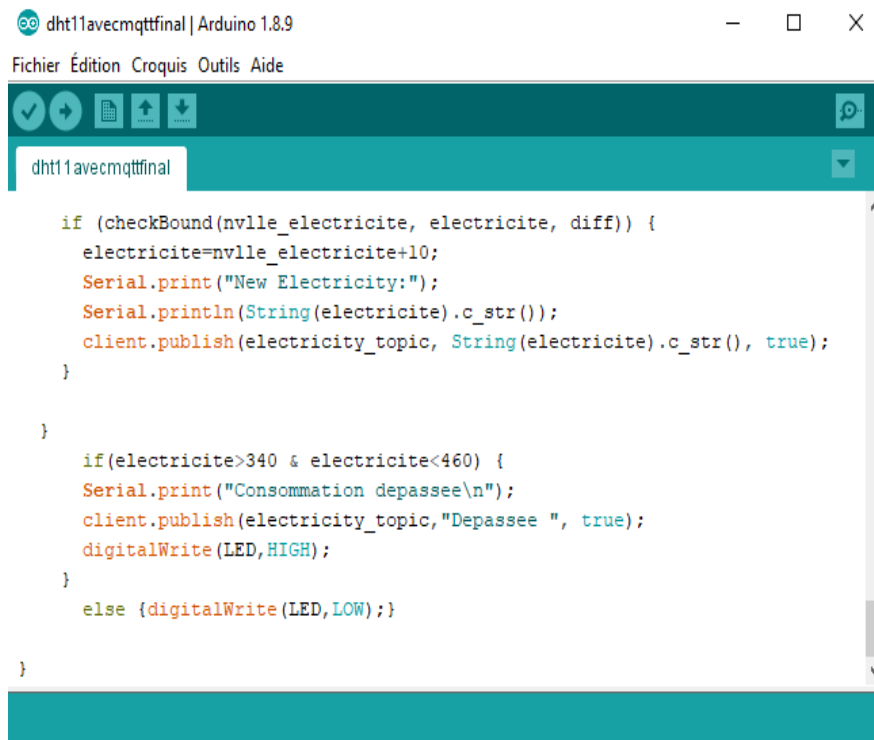
```
dht11avecmqtfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

dht11avecmqtfinal
long now = millis();
if (now - lastMsg > 2000) {
  lastMsg = now;

  if (checkBound(newTemp, temp, diff)) {
    temp = newTemp;
    Serial.print("New temperature:");
    Serial.println(String(temp).c_str());
    client.publish(temperature_topic, String(temp).c_str(), true);
  }

  if (checkBound(newHum, hum, diff)) {
    hum = newHum;
    Serial.print("New humidity:");
    Serial.println(String(hum).c_str());
    client.publish(humidity_topic, String(hum).c_str(), true);
  }
}
```

Figure III-15 Verification du changement des valeurs de temperature et humidite



```
dht11avecmqtfinal | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

dht11avecmqtfinal

if (checkBound(nvllle_electricite, electricite, diff)) {
  electricite=nvllle_electricite+10;
  Serial.print("New Electricity:");
  Serial.println(String(electricite).c_str());
  client.publish(electricity_topic, String(electricite).c_str(), true);
}

if(electricite>340 & electricite<460) {
  Serial.print("Consommation depassee\n");
  client.publish(electricity_topic,"Depassee ", true);
  digitalWrite(LED,HIGH);
}
else {digitalWrite(LED,LOW);}
}
```

Figure III-16 Affichage d'une alerte en cas de sucronsommation (valeur <460wh)

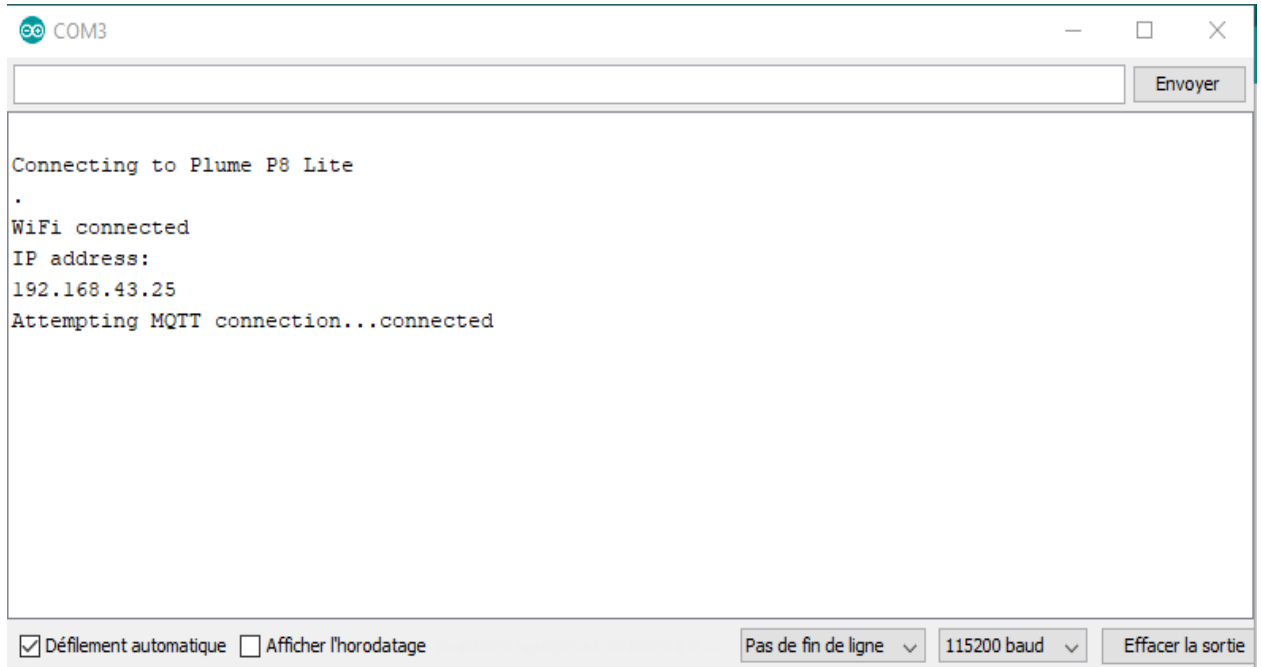


Figure III-17 Connexion del'esp8266 (vérification sur le moniteur serie)

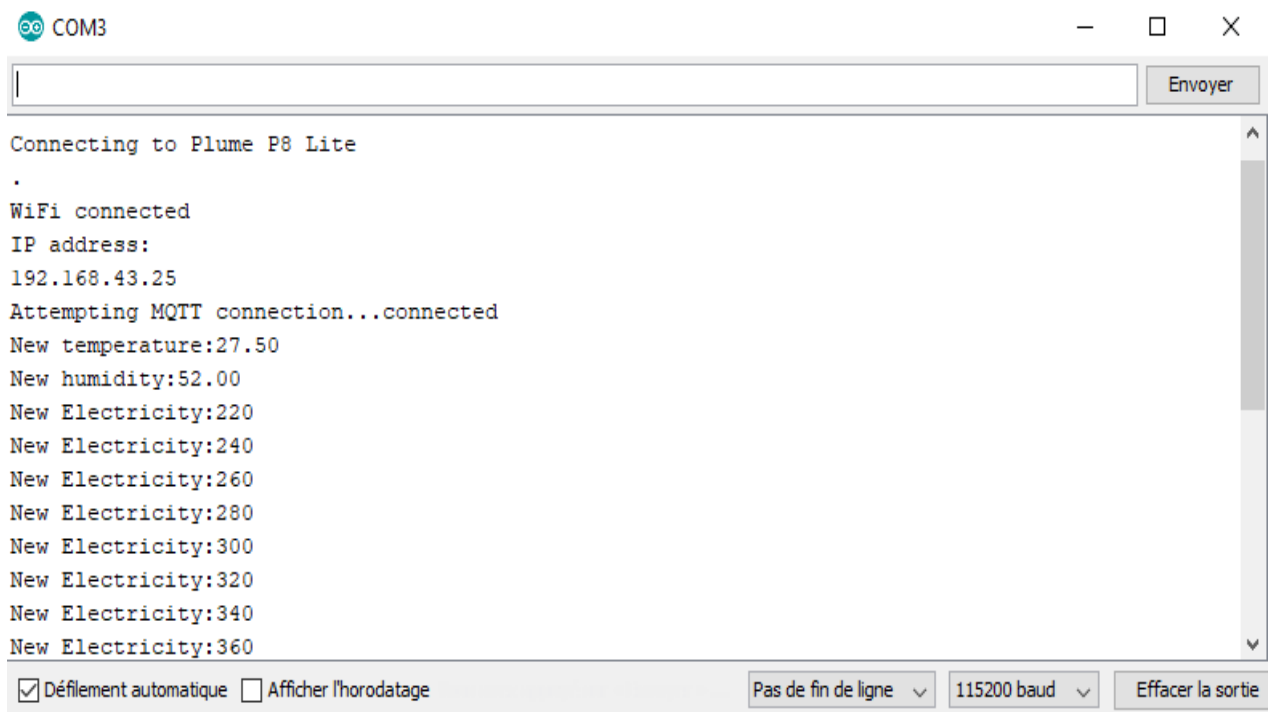


Figure III-18 Affichage des valeurs (temp/hum/elec) sur le moniteur série

Chapitre III : Implémentation et Réalisation

Cette figure représente un organigramme de programme Arduino qui fait la connexion entre la carte ESP8266 NodeMCU et le Broker MQTT et affiche les valeurs capturées.

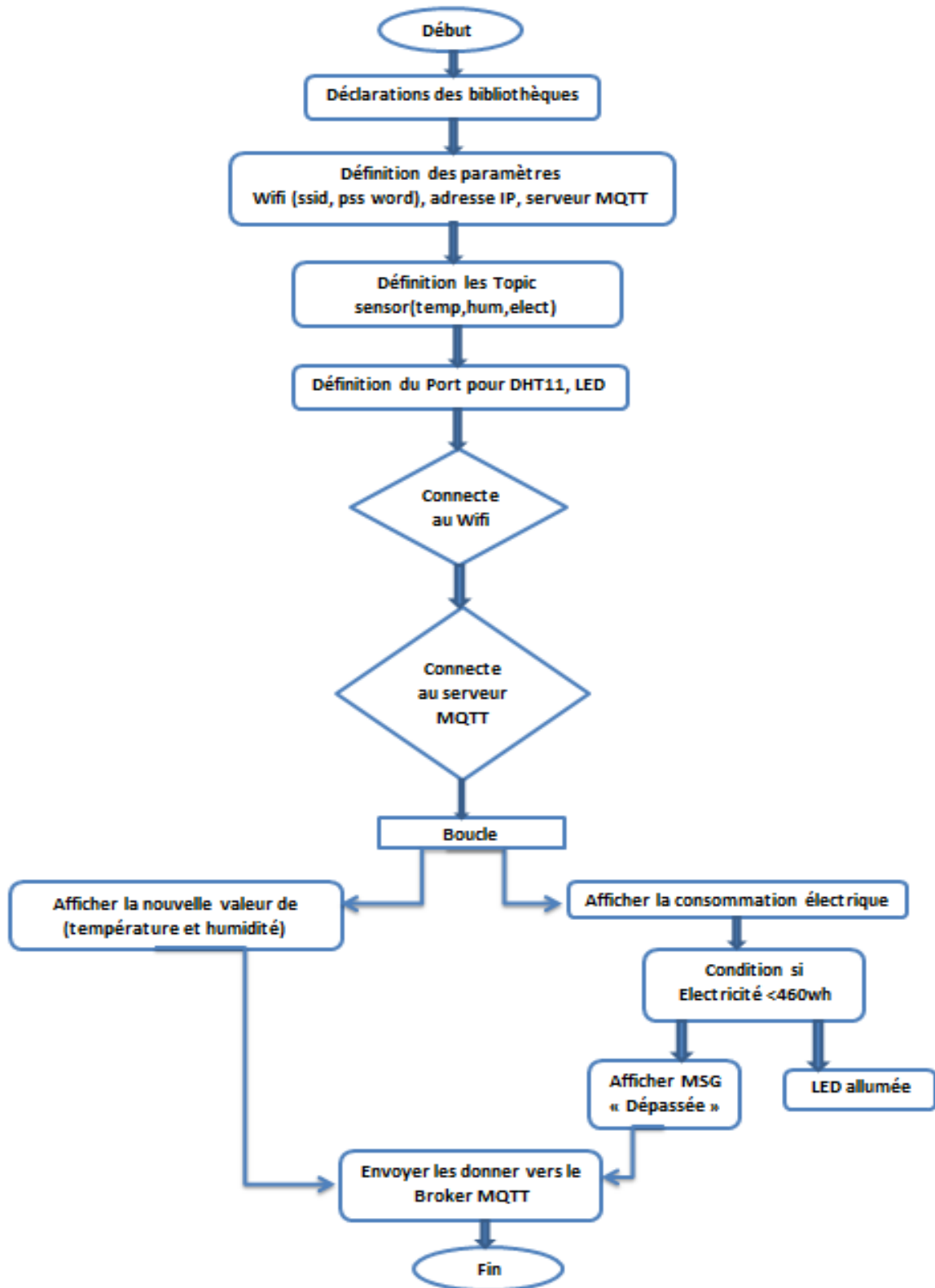
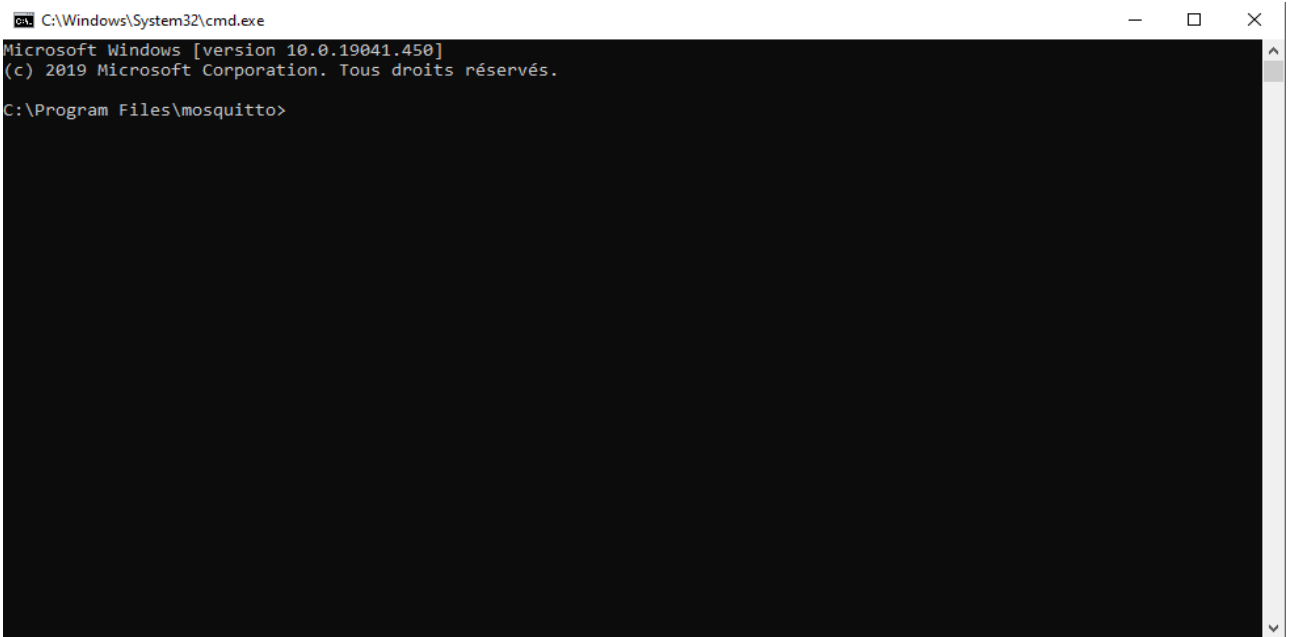


Figure III-19 Organigramme de programme Arduino.

III-2-3- Test de Mosquitto

Commencez par lancer le serveur mosquitto :



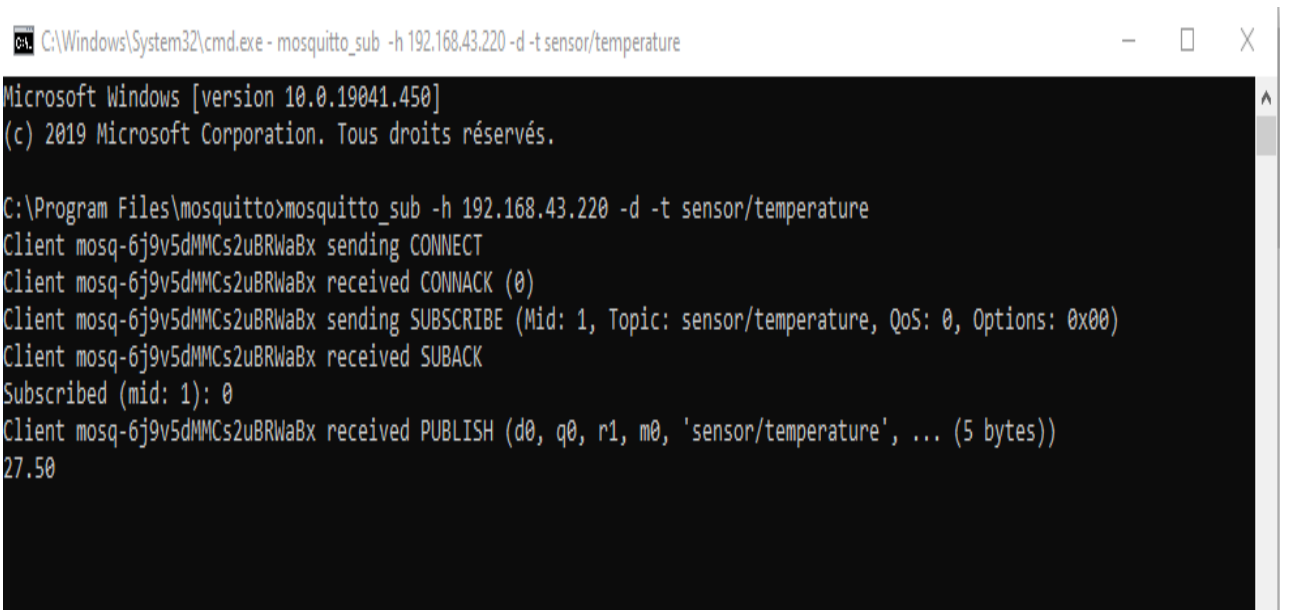
```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19041.450]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Program Files\mosquitto>
```

Figure III-20 Lancement de serveur mosquitto

Pour recevoir les valeurs de la température, il suffit d'ouvrir un terminal et d'exécuter la commande suivante :

mosquitto_sub -h 192.168.43.220 -d -t "sensor/temperature"



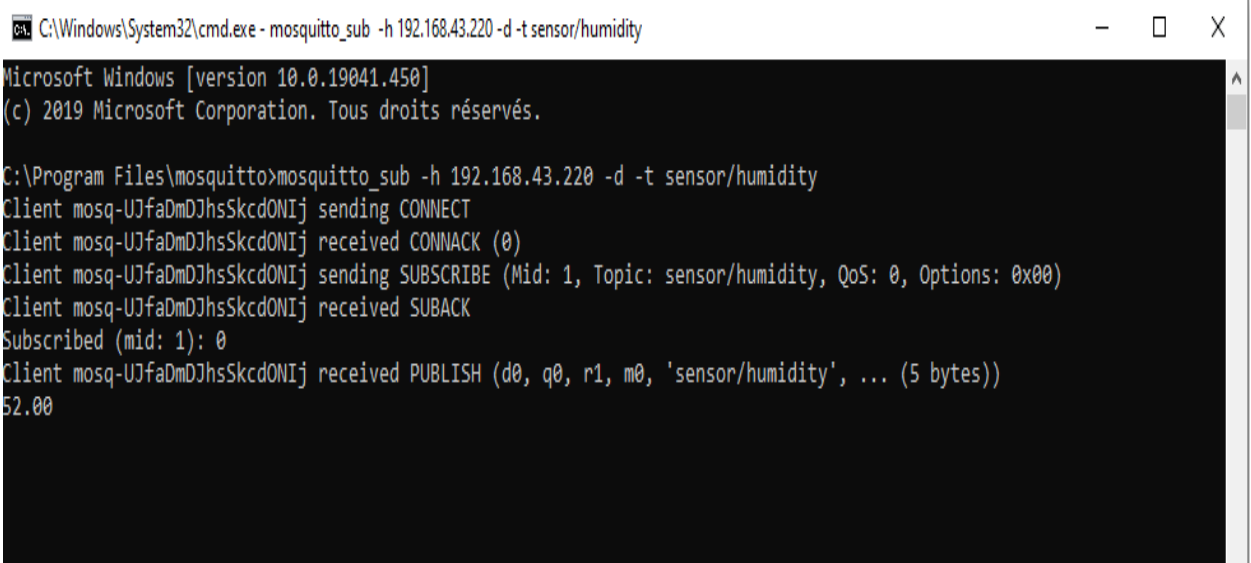
```
C:\Windows\System32\cmd.exe - mosquitto_sub -h 192.168.43.220 -d -t sensor/temperature
Microsoft Windows [version 10.0.19041.450]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.220 -d -t sensor/temperature
Client mosq-6j9v5dMMCs2uBRWaBx sending CONNECT
Client mosq-6j9v5dMMCs2uBRWaBx received CONNACK (0)
Client mosq-6j9v5dMMCs2uBRWaBx sending SUBSCRIBE (Mid: 1, Topic: sensor/temperature, QoS: 0, Options: 0x00)
Client mosq-6j9v5dMMCs2uBRWaBx received SUBACK
Subscribed (mid: 1): 0
Client mosq-6j9v5dMMCs2uBRWaBx received PUBLISH (d0, q0, r1, m0, 'sensor/temperature', ... (5 bytes))
27.50
```

Figure III-21 souscription au topic "sensor/temperature"

Pour recevoir les valeurs de l'humidité, il suffit d'ouvrir un terminal et d'exécuter la commande suivante

```
mosquitto_sub -h 192.168.43.220 -d -t "sensor/humidity"
```



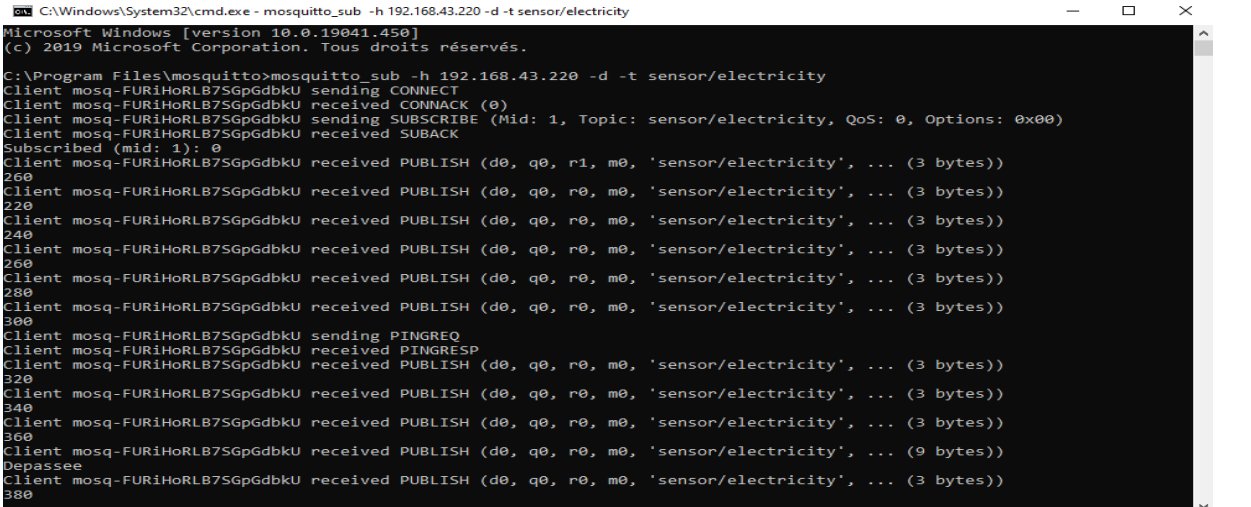
```
C:\Windows\System32\cmd.exe - mosquitto_sub -h 192.168.43.220 -d -t sensor/humidity
Microsoft Windows [version 10.0.19041.450]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.220 -d -t sensor/humidity
Client mosq-UJfaDmDJhsSkcdONIj sending CONNECT
Client mosq-UJfaDmDJhsSkcdONIj received CONNACK (0)
Client mosq-UJfaDmDJhsSkcdONIj sending SUBSCRIBE (Mid: 1, Topic: sensor/humidity, QoS: 0, Options: 0x00)
Client mosq-UJfaDmDJhsSkcdONIj received SUBACK
Subscribed (mid: 1): 0
Client mosq-UJfaDmDJhsSkcdONIj received PUBLISH (d0, q0, r1, m0, 'sensor/humidity', ... (5 bytes))
52.00
```

Figure III-22 souscription au topic "sensor/humidity"

Pour recevoir la valeur de la consommation de l'énergie électrique, il suffit d'ouvrir un terminal et d'exécuter la commande suivante

```
mosquitto_sub -h 192.168.43.220 -d -t "sensor/electricity"
```



```
C:\Windows\System32\cmd.exe - mosquitto_sub -h 192.168.43.220 -d -t sensor/electricity
Microsoft Windows [version 10.0.19041.450]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.220 -d -t sensor/electricity
Client mosq-FURiHoRLB7SgpGdbkU sending CONNECT
Client mosq-FURiHoRLB7SgpGdbkU received CONNACK (0)
Client mosq-FURiHoRLB7SgpGdbkU sending SUBSCRIBE (Mid: 1, Topic: sensor/electricity, QoS: 0, Options: 0x00)
Client mosq-FURiHoRLB7SgpGdbkU received SUBACK
Subscribed (mid: 1): 0
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r1, m0, 'sensor/electricity', ... (3 bytes))
260
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
220
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
240
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
260
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
280
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
300
Client mosq-FURiHoRLB7SgpGdbkU sending PINGREQ
Client mosq-FURiHoRLB7SgpGdbkU received PINGRESP
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
320
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
340
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
360
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (9 bytes))
Dépassée
Client mosq-FURiHoRLB7SgpGdbkU received PUBLISH (d0, q0, r0, m0, 'sensor/electricity', ... (3 bytes))
380
```

Figure III-23 souscription au topic "sensor/electricity"

Symbole # : Pour souscrire simultanément aux trois valeurs (Température, humidité, et l'électricité)

```
mosquitto_sub -h 192.168.43.220 -d -t "sensor/#"
```



```
C:\Windows\System32\cmd.exe - mosquitto_sub -h 192.168.43.220 -d -t sensor/#
Microsoft Windows [version 10.0.19041.450]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.43.220 -d -t sensor/#
Client mosq-XSSW9V005Bnmwt3R9Y sending CONNECT
Client mosq-XSSW9V005Bnmwt3R9Y received CONNACK (0)
Client mosq-XSSW9V005Bnmwt3R9Y sending SUBSCRIBE (Mid: 1, Topic: sensor/#, QoS: 0, Options: 0x00)
Client mosq-XSSW9V005Bnmwt3R9Y received SUBACK
Subscribed (mid: 1): 0
Client mosq-XSSW9V005Bnmwt3R9Y received PUBLISH (d0, q0, r1, m0, 'sensor/temperature', ... (5 bytes))
28.30
Client mosq-XSSW9V005Bnmwt3R9Y received PUBLISH (d0, q0, r1, m0, 'sensor/humidity', ... (5 bytes))
51.00
Client mosq-XSSW9V005Bnmwt3R9Y received PUBLISH (d0, q0, r1, m0, 'sensor/electricity', ... (4 bytes))
1000
```

Figure III-24 souscription au trois topics (température, humidité et électricité) commande "sensor/#"

III-2-4- l'Application avec dans MQTT Dash

Une fois MQTTDash lancé, nous commençons un nouveau projet en cliquant sur le signe «+». Une nouvelle configuration de connexion s'ouvrira. Ceci nous permet d'entrer les informations suivantes :

- Name
- Adresse
- Por



Figure III-25 Configuration App MQTT Dash N°1

Après avoir enregistré en appuyant sur l'icône de la disquette en haut à droite, il faudra s'assurer que la connexion créée est ouverte.

Si les paramètres sont corrects, une fois la connexion correctement configurée, nous cliquons sur le signe « + » dans le tableau de bord créé

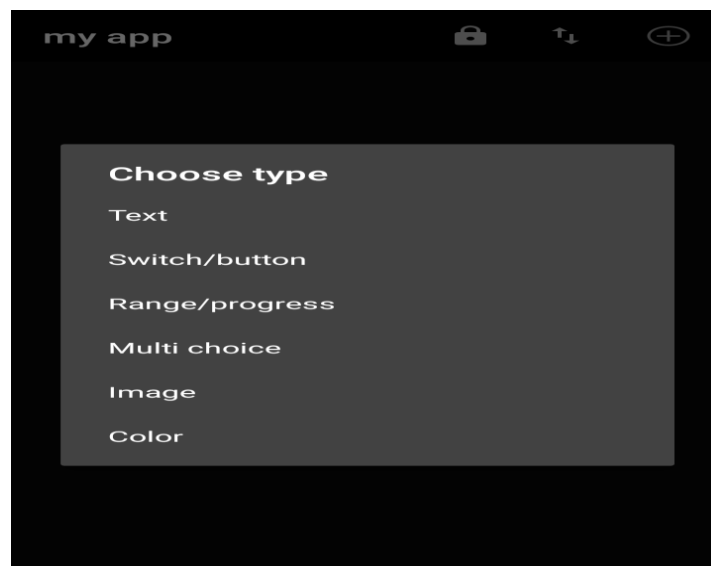


Figure III-26 Configuration App MQTT Dash N°2

Nous sommes en mesure de créer des boutons et des indicateurs que nous pouvons utiliser pour contrôler la température et l'humidité est la consommation de l'électricité.

Donc , nous insérons trois fois le switch/Button :

- ✓ Le premier sur le nom temperature avec comme topic « sensor/temperature »
- ✓ Deuxième sur le nom humidity avec comme topic « sensor/humidity »
- ✓ Troisième sur le nom electricity avec comme topic « sensor/electricity »

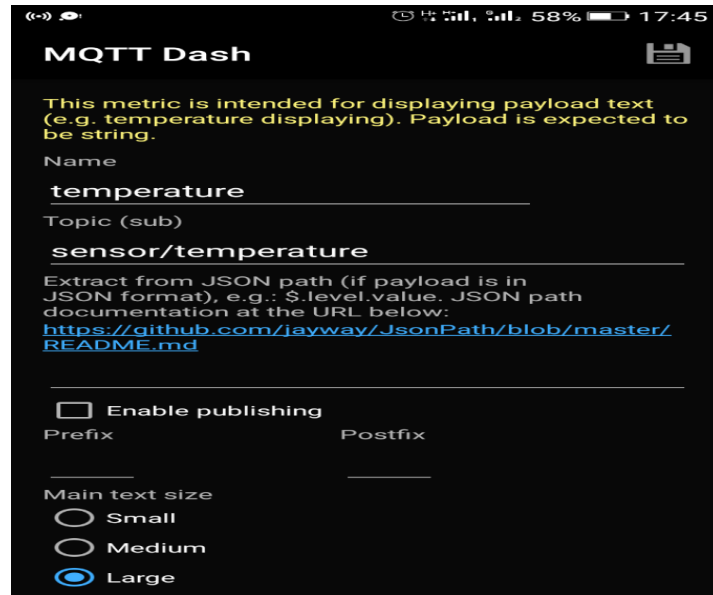


Figure III-27 Création de l'indicateur de température.

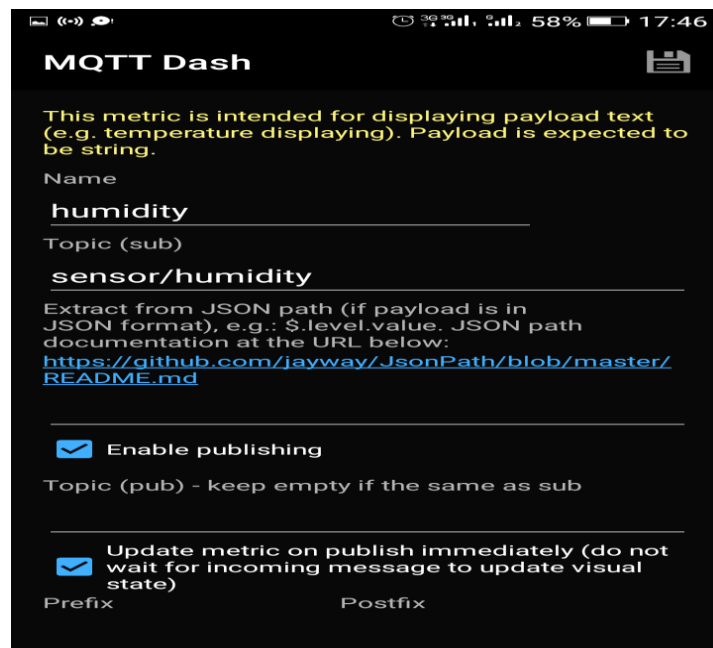


Figure III-28 Création de l'indicateur de l'humidité.

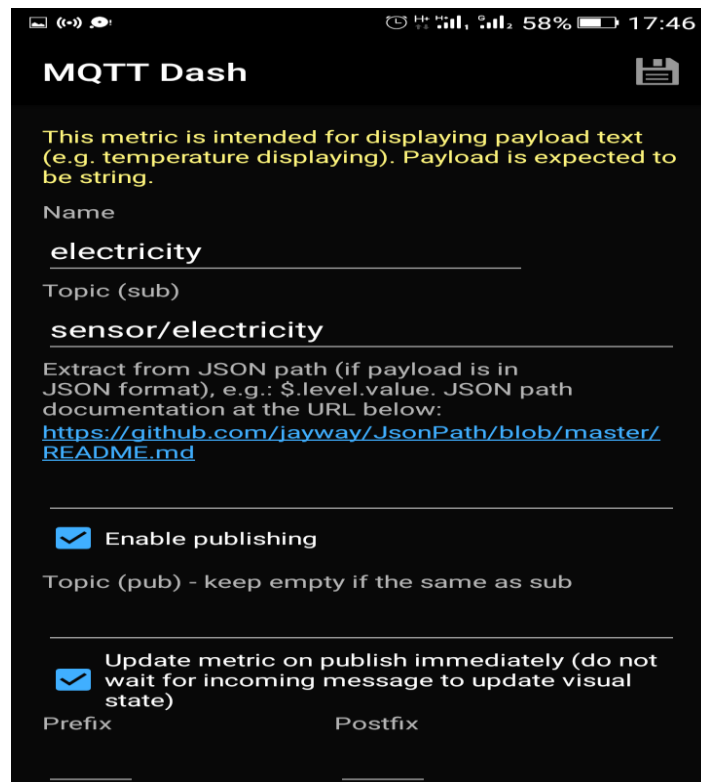


Figure III-29 Création de l'indicateur de l'électricité.

La figure suivante nous montre l'interface que nous avons créée sur MQTT Dash.

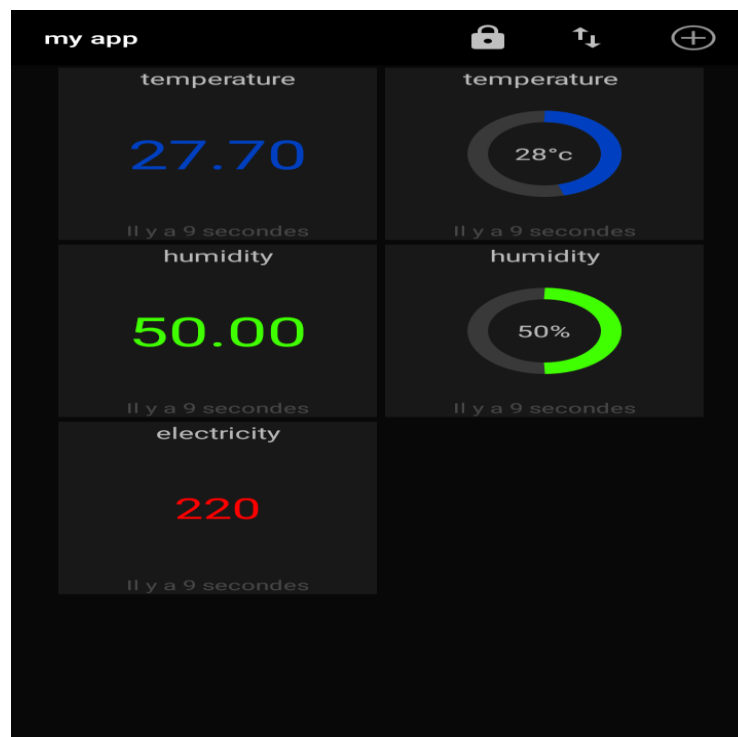


Figure III-30 Interface de notre application sur MQTT Dash

III-2-4-Resultats de simulation

L'ESP8266 récupère la valeur de la température et de l'humidité à partir du capteur DHT11 et les envoie au broker MQTT.

Pour la consommation de l'énergie électrique, nous avons écrit un programme pour générer des valeurs de façon virtuelle, et les envoyer automatiquement à notre serveur MQTT.

Cette simulation de la consommation de l'électricité affiche la valeur chaque 10 secondes, si cette valeur dépasse un seuil (460w/h par exemple) un message d'alerte est affiché et en même temps une LED (branché sur la carte microcontrôleur ESP8266) sera allumée pour indiquer que la consommation de l'électricité est élevée.

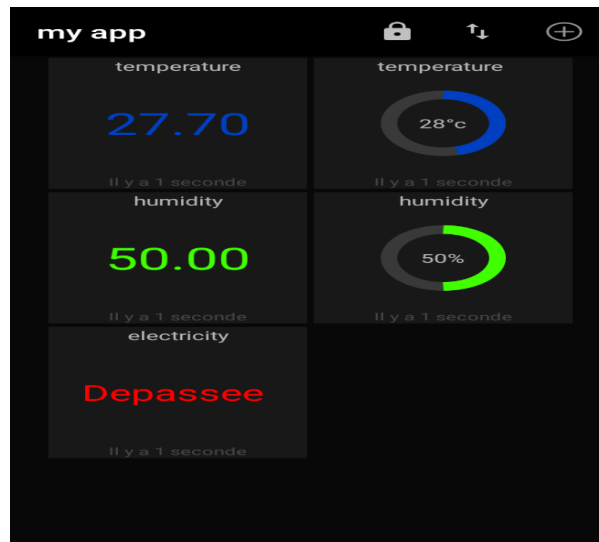


Figure III-31 Interface sur MQTT Dash (exemple avec alerte sur la consommation d'électricité > 460w/h)

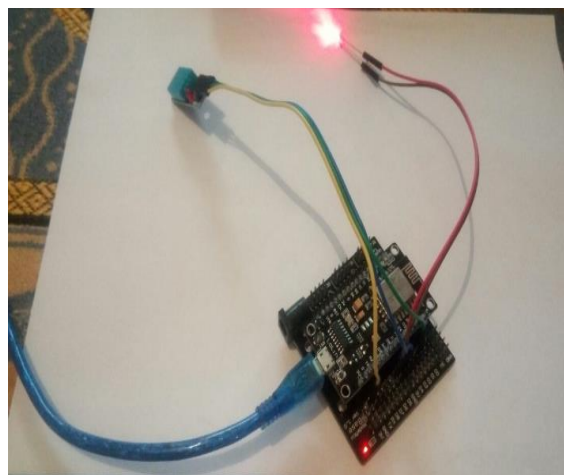


Figure III-32 Led (allumé) pour alerte de surconsommation sur l'ESP8266

III-2-5 Réalisation de la plateforme pour l'affichage web :

Nous avons aussi proposé une deuxième solution web en plus de la solution mobile proposée expliquée précédemment. L'application web repose sur une architecture 3 tiers (client/serveur web et serveur Bases de Données).

Cette solution nous permet non seulement l'affichage des données capturées à chaque instant tout comme l'application mobile, mais aussi le stockage de toutes ces données dans une base de données MySQL pour des fins de statistiques et d'analyse.

Notre solution est très efficace vu l'indépendance entre la communication mobile et la communication web avec le broker (i.e. si le mobile est en panne l'affichage web n'est pas touché et vice versa).

Comme illustré dans la figure III.27, les données récoltées par le broker MQTT seront stockées dans la base de données appelée "office".

Cette base de données comporte trois tables, chacune stockera les données d'un capteur.

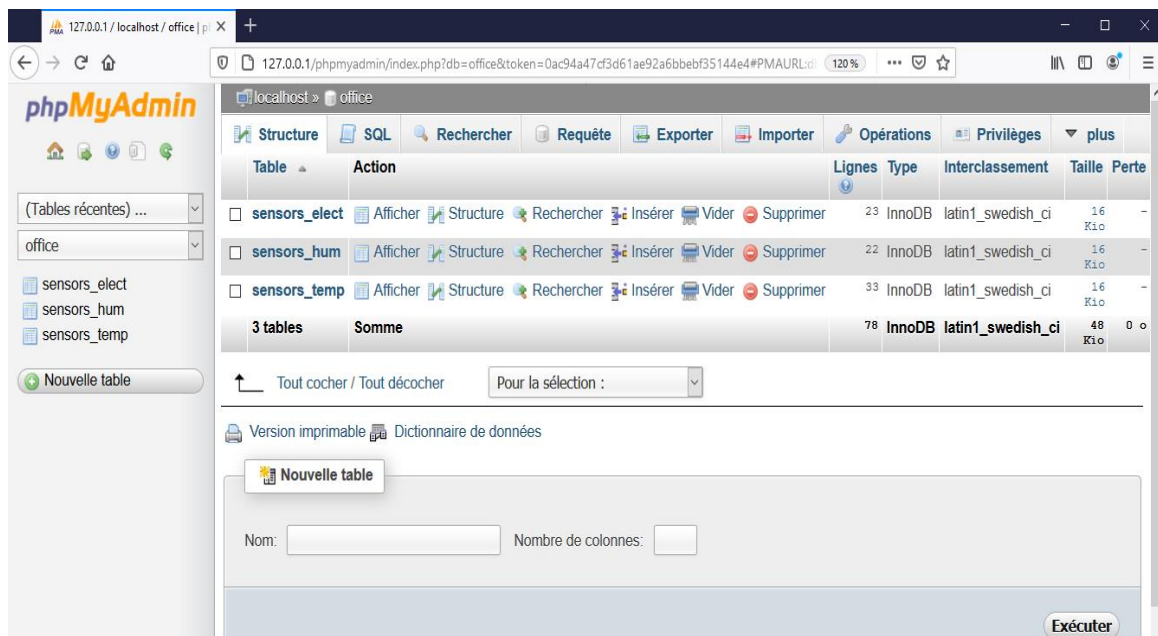


Figure III-33 La base de données MySQL "office"

Chaque table comporte un identifiant (device_id), dans notre cas c'est le type de la donnée captée (électricité, humidité ou température), une colonne pour la valeur à stocker, et qui est captée par le capteur et renvoyée par le broker, et une dernière colonne qui représente la date et l'heure de la capture de la donnée. Ce dernier attribut est la clé de notre table.

Les figures suivantes (III.34, III.35, III.36) illustrent une partie des données renvoyées par les différents capteurs vers la base de données.

			device_id	valeur	last_heard	
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3280	2020-09-27 22:08:21
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3300	2020-09-27 22:08:23
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3320	2020-09-27 22:08:31
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3500	2020-09-27 22:10:06
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3520	2020-09-27 22:10:11
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3540	2020-09-27 22:10:21
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	260	2020-09-28 12:07:55
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	280	2020-09-28 12:08:01
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	540	2020-09-28 12:10:14
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	560	2020-09-28 12:10:21
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3000	2020-09-28 17:03:15
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3020	2020-09-28 17:03:25
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3040	2020-09-28 17:03:36
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	3060	2020-09-28 17:03:49
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	960	2020-09-28 20:46:32
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	980	2020-09-28 20:46:42
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	980	2020-09-28 20:46:52
<input type="checkbox"/>	Modifier	Copier	Effacer	electricity	1000	2020-09-28 20:46:57

Figure III-34 La table du capteur de consommation d'électricité

				device_id	valeur	last_heard
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	48	2020-09-27 21:45:49
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	48	2020-09-27 21:46:11
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	48	2020-09-27 21:46:46
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	51	2020-09-27 21:46:48
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	53	2020-09-27 21:46:57
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	57	2020-09-27 21:51:31
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	55	2020-09-27 21:51:48
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	40	2020-09-28 12:08:09
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	42	2020-09-28 12:08:21
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	40	2020-09-28 12:11:36
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	38	2020-09-28 17:03:15
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	38	2020-09-28 17:03:25
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	38	2020-09-28 17:03:36
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	38	2020-09-28 17:03:49
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	46	2020-09-28 20:41:40
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	46	2020-09-28 20:41:55
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	46	2020-09-28 20:42:38
<input type="checkbox"/>	Modifier	Copier	Effacer	humidity	46	2020-09-28 20:46:16

Figure III-35 La table du capteur Humidité

	device_id	valeur	last_heard
<input type="checkbox"/>	temperature	24.6	2020-09-27 13:57:01
<input type="checkbox"/>	temperature	28.2	2020-09-27 21:46:04
<input type="checkbox"/>	temperature	27.1	2020-09-27 21:47:11
<input type="checkbox"/>	temperature	26	2020-09-27 21:51:24
<input type="checkbox"/>	temperature	28.6	2020-09-27 21:57:50
<input type="checkbox"/>	temperature	25.5	2020-09-28 12:08:28
<input type="checkbox"/>	temperature	25.5	2020-09-28 12:12:49
<input type="checkbox"/>	temperature	28.2	2020-09-28 12:13:40
<input type="checkbox"/>	temperature	27	2020-09-28 12:13:42
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:43:54
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:01
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:08
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:15
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:22
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:30
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:37
<input type="checkbox"/>	temperature	27.6	2020-09-28 16:44:44
<input type="checkbox"/>	temperature	28.1	2020-09-28 17:01:18

Figure III-36 La table du capteur Température

Cette figure représente un organigramme de programme qui fait la connexion entre le Broker MQTT et notre base de données MySQL pour afficher et stocker toutes les données capturées.

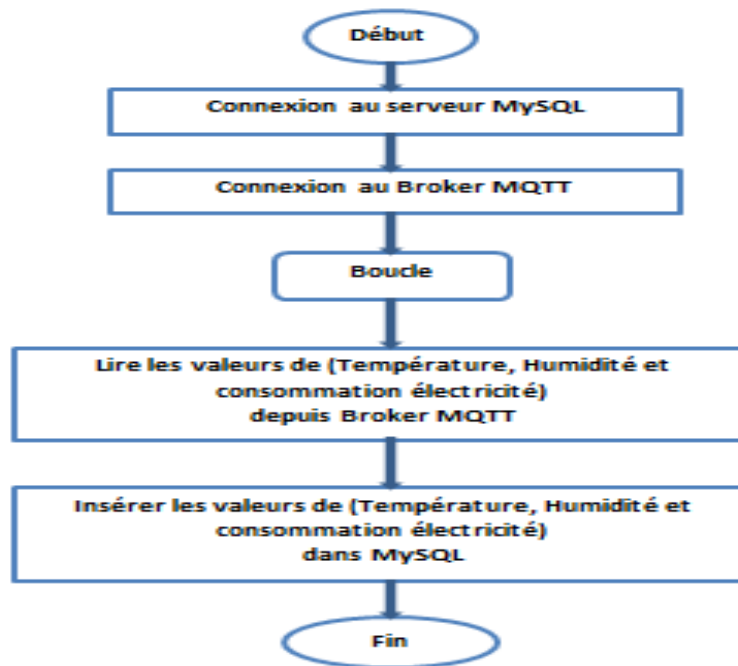


Figure III-37 Organigramme de connexion entre le Broker MQTT et la base de données MySQL

Un script en python a été utilisé comme intermédiaire entre le broker MQTT et la base de données MySQL.

Les captures d'écran des figures III.38 et III.39 comportent une partie des scripts de communication avec MQTT et MySQL.

```
|import paho.mqtt.client as mqtt
import pymysql.cursors
import sys
import json
import time

# User variable for Gateway ID
myGatewayID = "30:AE:A4:DA:3F:BC"

#User variable for database name
dbName = "office"

# it is expected that this Database will already contain one table called sensor
# CREATE TABLE sensors(device_id char(23) NOT NULL, transmission_count INT NOT N

# User variables for MQTT Broker connection
mqttBroker = "192.168.43.220"
mqttBrokerPort = 1883
mqttUser = '*'
mqttPassword = '*'

mysqlHost = "localhost"
mysqlUser = "root"
mysqlPassword = ""

# This callback function fires when the MQTT Broker connecton is established.
def on_connect(client, userdata, flags, rc):
    print("MQTT Client Connected")
    client.subscribe("sensor/electricity/#")
    try:
        db = pymysql.connect(host=mysqlHost, user=mysqlUser, password=mysqlPassw
        db.close()
        print("MySQL Client Connected")
    except:
        sys.exit("Connection to MySQL failed")
```

Figure III-38 Script de connexion python-MQTT

```
def on_message(client, userdata, msg):
    print("Transmission received msg")
    print(float(msg.payload))
    sensor_id="electricity"
    db = pymysql.connect(host="localhost", user=mysqlUser, password=mysqlPassword, db=dbName,charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
    cursor = db.cursor()
    insertRequest = "INSERT INTO sensors_elect(device_id, valeur, last_heard) VALUES('%s',%f,CURRENT_TIMESTAMP)" % (sensor_id, float(msg.payload))
    cursor.execute(insertRequest)
    db.commit()
    db.close()
```

Figure III-39 Script de connexion python-My-SQL

Une fois la connexion entre le broker MQTT et MySQL établie, et les données stockées correctement, nous avons procédé à la construction d'une page web pour l'affichage des valeurs capturées. Pour cela, nous avons eu recours au langage de programmation PHP. La figure (III.40) suivante illustre notre interface web.



Figure III-40 L'interface Web

Cette figure représente un organigramme de programme qui permet d'afficher les données capturées dans une interface Web.

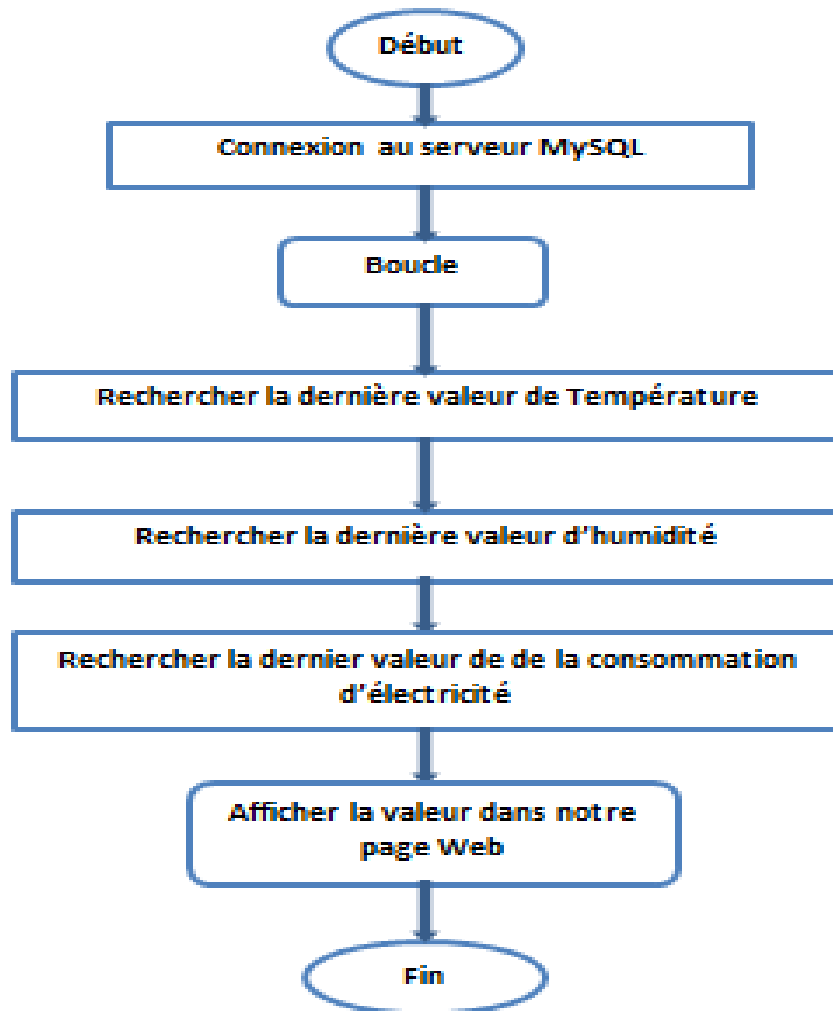


Figure III-41 Organigramme d'affichage les dernier valeur capturer.

Si on clique sur le bouton « consommation » une autre interface sera affichée. Cette interface nous permet d'afficher les consommations de l'électricité maximum et minimum et moyenne et la somme des consommations dans un intervalle du temps.

Nous affichons aussi d'autres valeurs significatives comme la température max et min et moyenne, l'humidité max et min et moyenne dans le même intervalle de temps.

Les consommations

Début Fin

[Envoyer](#)

	Temperature (°C)		Humidity (%)		Electricity (KWatt)	
	Valeur	Date	Valeur	Date	Valeur	Date
Minimum	24.6	2020-09-27 13:57:01	38	2020-09-28 17:03:15	0.26	2020-09-28 12:07:55
Maximim	32.2	2020-09-28 20:53:46	95	2020-09-28 20:54:10	6.24	2020-09-28 23:00:37
Moyenne	27.377777989705404		47.21875		2.4947619047619045	
Cumul					104.78	

[Retour](#)

Figure III-42 Interface d'affichage de la consommation

Cette figure représente un organigramme de programme qui permet de rechercher les valeurs (MAX, MIN, MOY) dans un intervalle de temps précis et les afficher dans un tableau.

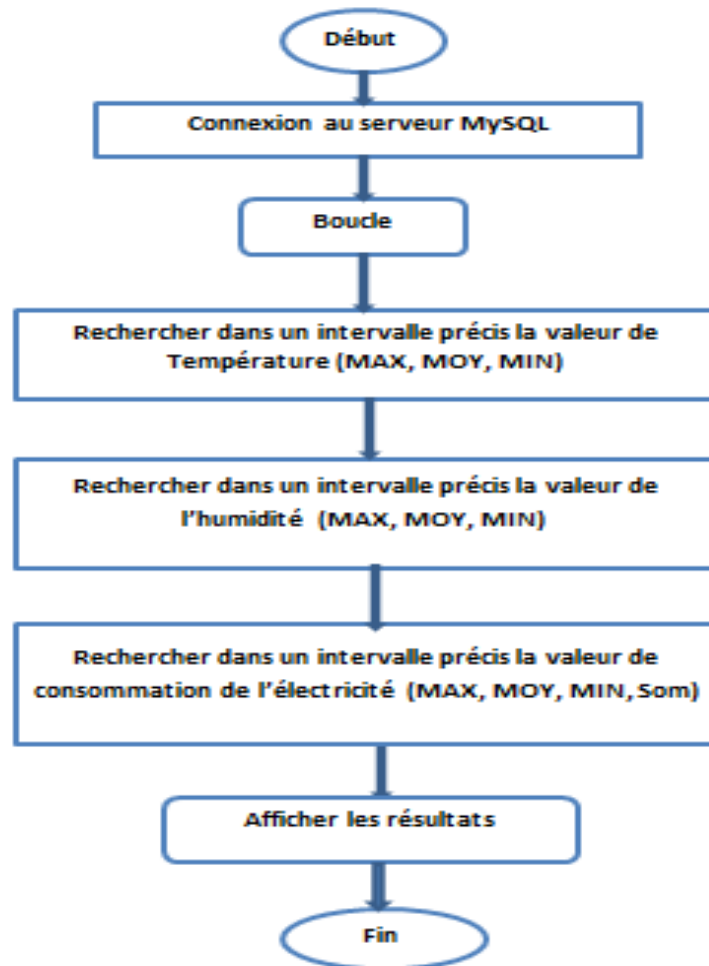


Figure III-43 Organigramme qui affiche les valeurs dans un intervalle précis du temps.

Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de la solution proposée. Nous avons réalisé un système qui permet de suivre la consommation d'énergie électrique en temps réel. En plus, des capteurs de température et d'humidité permettent de mieux calculer l'efficacité énergétique du système de climatisation (chaud et froid), et ainsi aider à mieux optimiser ses consommations.

CONCLUSION GENERALE

À travers ce modeste travail, nous avons eu l'occasion d'utiliser plusieurs outils informatiques et électroniques qui sont nécessaires pour la réalisation de notre projet de fin d'étude, pour créer un système permettant de gérer la consommation d'énergie dans des locaux pour habitation ou entreprise.

Le système, permet de suivre la consommation d'énergie électrique en temps réels, en plus des capteurs de température et d'humidité permettent de mieux calculer l'efficacité énergétique du système de climatisation (chaud et froid).

Le système est en mesure de renseigner l'utilisateur sur les horaires de pics de consommations, et ainsi l'aider à mieux optimiser ses consommations.

Au cours de ce mémoire, nous avons présenté au début une vue générale sur l'internet des objets (définition, domaine d'application, l'architecture, et les défis et les bénéfices de l'IOT), dans le deuxième chapitre nous avons vu les différents protocoles de l'IOT, puis, nous avons présenté le protocole MQTT (Définition, mode fonctionnement, souscription et publications, les topics, la Qualité de service, et la sécurité). Dans le dernier chapitre relatif à l'implémentation et réalisation, nous avons présenté d'abord en première partie, l'environnement de travail (le logiciel Arduino, carte microcontrôleur ESP8266 Node MCU, le capteur DHT11, l'application mobile, et le broker MQTT), et dans la deuxième partie la réalisation pratique de notre projet.

Nous souhaitons vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies et pouvant même être intégré dans d'autres système IOT plus complexes.

REFERENCES BIBLIOGRAPHIQUES

- [1] Mehdi Nemri, l'internet des objets, France STRATEGIE, <https://www.strategie.gouv.fr/publications/demain-linternet-objets#>, consulté le 11/05/2020
- [2] Mekriou Ryma et Mazari Walid, Introduction à l'internet de l'objet et réalisation D'un système domotique, Mémoire de master en télécommunication, Université A/Mira de Bejaïa, 2016.
- [3] L'Internet des Objets (IoT), qu'est-ce que c'est ? : Synox. <https://www.synox.io/internet-des-objets-iot-qu-est-ce-que-cest>, consulté le 11/05/2020.
- [4] Internet des Objets, Comment l'évolution actuelle d'Internet transforme-t-elle le monde ? , Dave Evans, Cisco Internet Business Solutions Group (IBSG), Avril 2011.
- [5] Dr. Ovidiu et all, Internet of Things–From Research and Innovation to Market Deployment, EU, Belgium , river publishers' series in Communications, 2014.
- [6] Rafael Marques, Système de produits et services basés sur l'internet des objets, Mémoire de maîtrise des sciences appliquées, école polytechnique de Montréal, juillet 2018.
- [7] Zeinab Kamal Aldein Mohammed, Elmustafa Sayed Ali Ahmed, Internet of Things Applications, Challenges and Related Future Technologies. WSN 67(2) 126-148, 2017.
- [8] Les bénéfices de l'IoT, site Intesens <https://www.intesens.com/les-benefices-de-liot/>, consulté le 20/05/2020.
- [9] Technologies et protocole, Partie 2 <https://azure.microsoft.com/fr-fr/overview/internet-of-things-iot/iot-technology-protocols/>, consulté le 20/05/2020.
- [10] Protocoles à connaître pour l'Internet des objets (IoT), RS Components 20 April 2015, <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about-fr>, consulté le 14/06/2020.
- [11] S. Feng et all, Sécurité des objets Connectés. Rapport Auditeurs Cigref Inhesj, Institut national des hautes études de la sécurité et de la justice, 2014.
- [12] R. Achour et N.Makhloufi, Authentification dans l'internet des objets. Université A/MIRA de Bejaia, 2017
- [13] La Rédaction JDN, 2019 <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203403-tcp-transmission-control-protocol-definition-traduction/>, consulté le 14/06/2020.

- [14] Yahiaoui Billal et Sfaïhi Ali, Technologie RFID, Étude et application, Mémoire de master, Université A/Mira de Béjaïa, 2014/2015
- [15] Edited by Andrew Banks and Rahul Gupta, MQTT Version 3.1.1, OASIS , 29 October 2014.
- [16] mqtt-topics-best-practices
URL: <http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>, consulté le 28 /06/2020.
- [17] Benaoune Siham, Evaluation de performance de protocoles dans l'internet des objets à base de méthodes formelles, Mémoire de fin d'études, Université Mohamed Khider – Biskra, 06/05/2017.
- [18] GroupeSII Le protocole MQTT dans l'IoT,2017
<https://blog.groupe-sii.com/le-protocole-mqtt-dans-liot/>, consulté le 10 /07/2020.
- [19] Eric J. Bruno, MQTT Programming In Depth, March 2016,
URL: <http://programmingwithreason.com/article-mqtt-in-depth.html>, consulté le 10 /07/2020.
- [20] Travaux de la 4e promotion (2013-2014) du Cycle « Sécurité des usages numériques », p 28.
- [21] Frédéric Génévey et Jean_Pierre Dulex, Arduino à l'école, Livre, Edition Septembre 2018.
- [22] Djafri Menad et Chelouche Djalal, Etude et Réalisation d'une Carte Arduino, mémoire de master, Université A. Mira de Bejaia ,2016.
- [23] Simon Landrault (Eskimon) et Hippolyte Weisslinger (olyte), Arduino : 1er Pas en informatique embarquée, Livre, Édition du 01 juin 2014.
- [24] SAHRAOUI Seif eddine, Etude et réalisation d'une canne intelligente pour les non-voyants, Mémoire de fin d'études, Université Larbi Ben M'hidi - Oum El Bouaghi, juillet 2019.
- [25] Utiliser un capteur de température et d'humidité DHT11 / DHT22 avec une carte Arduino / Genuino.
<https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>, consulté le 16 /08/2020.
- [26] Débuter avec le Broker MQTT Mosquitto (Raspberry Pi, Windows, macOS, Linux), DOMITIQUE ET OBJETS CONNECTE, <https://projetsdiy.fr/mosquitto-broker-mqtt-raspberry-pi/>, consulté le 16/08/2020.
- [27] Brahimi Youssouf, Conception et réalisation d'un site Web pour la gestion d'un jardin, d'enfants, Mémoire de fin d'études, 26 mai 2015.

- [28] Gartner's 2014 hype cycle for emerging technologies maps the journey to digital business", August 2014, <https://www.gartner.com/en/newsroom/press-releases/2014-08-11-gartners-2014-hype-cycle-for-emerging-technologies-maps-the-journey-to-digital-business>, consulté le 06 /09/2020.
- [29] <https://www.bluetooth.com/>, consulté le 12/09/2020.