

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث 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

présenté par

Khaled NAHILIA

pour l'obtention du diplôme de Master en Réseaux et Télécommunications

Thème

Conception d'une plateforme autonome pour l'internet des objets

Proposé par : Yacine Kabir

Année Universitaire 2018-2019

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mr : Yacine KABIR, son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

*C'est avec profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d'étude à mes chers **parents et frères**, pour leur patience, leur encouragement et leur soutien. J'espère qu'un jour, je puisse leurs rendre un peu de ce qu'ils ont fait pour moi. Que dieu leur prête bonheur et longue vie.*

Je dédie aussi ce travail à mes chers amis

***Walid, Adel, Rafik, Kamil, Sidahmed,
Mohamed, Mourad, Hachem, Kheireddine,
Zaki, Aziz, Azzaz, Yacine***

en témoignage de l'amitié qui nous a unis et des souvenirs de tous les moments que nous avons passé ensemble.

A tous ceux qui m'ont aidé, soutenu, et encouragé pour la réalisation de ce modeste travail

Et à tous ceux qui me sont chers.

ملخص: إن مجال إنترنت الأشياء هو مجال حديث للغاية، فقد أحدث تطوير علم شبكات الكمبيوتر فوائد أكثر للمستخدمين. كان الافتقار إلى هذا النوع من التطبيق في بلدنا هو الدافع الرئيسي لمثل هذا العمل ، والذي يتمثل هدفه الأول في نشر فكرة إنترنت الأشياء وإعطاء نظرة عامة على تقنياتهم ومزاياها وتطبيقاتها. بالإضافة إلى ذلك ، وكتطبيق عملي في هذا المجال ، أنشأنا نظامًا يفترض أن يكون ، في الوقت نفسه ، مستقلًا وفعالًا وقابل للتخصيص.

تتمثل قيمة الانخراط في مجال جديد نسبيًا من الأبحاث مع تطبيقات غيرت الحياة في توفير دليل شامل للطلاب والباحثين المهتمين بهذا المجال ، وفتح آفاق وأفكار جديدة لاستمرارية هذا المشروع.

الكلمات المفتاحية: إنترنت الأشياء، الشبكات المتشعبة(المختلطة)، الأنظمة المدمجة.

Résumé : L'internet des objets est un domaine très récent, le développement de la science des réseaux informatiques et la technologie à basse échelle a dégagé davantage de bénéfices, dans le but de tendre vers la recherche et l'étude académique dans l'échelle industriel, et avec le manque de ce genre de système dans notre pays, on va essayer d'établir un système qui doit être autonome et bon marché , on va le modéliser pour qu'il soit accessible à distance, et capable de supporter plusieurs applications en même temps.

L'intérêt d'engager une recherche dans un domaine relativement très récent et dont les applications dans la vie sont en évolution, est de fournir un guide complet pour les étudiants et chercheurs intéressés par l'internet des objets, et ouvrir de nouvelles perspectives et idées pour une continuation de ce projet.

Mots clés : internet des objets, Réseaux maillés, Raspberry, NodeMCU, systèmes embarqués

Abstract: The Internet of Things is a very recent field, the development of the science of computer networks has generated more benefits for users. The lack of this kind of application in our country was the major motivation for such a work, whose first goal is to popularize the notion of IoT and give an overview of their technologies, their advantages and their applications. In addition, and as a practical application in this area, we have established a system that is supposed to be, at the same time, autonomous, efficient and customizable.

The value of engaging in a relatively new field of research with life-changing applications is to provide a comprehensive guide for students and researchers interested in this field, and to open new perspectives and ideas for continuation. of this project.

Keywords : internet of things, mesh network, Raspberry, NodeMCU, Embedded systems

Listes des acronymes et abréviations

ADC	Analog-to-Digital Converter
BD/DB	Base de Données / Data Base
CLI	Command line interface
FTP	File Transfer Protocol
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile communications
I2C	Inter-Integrated Circuit
I2S	Integrated Interchip Sound
IDE	integrated development environment
IHM	Interface Homme-Machine
IoT	Internet of Things
LDR	Light Dependent Resistor
LED	Light Emitting Diode
MCU	Micro Controller Unit
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
OS	Operating System
PWM	Pulse Width Modulation
RAM	Random-Access Memory
RFID	Radio-Frequency IDentification
SMS	Short Message Service
SPI	Serial Peripheral Interface
SSH	Secure Shell
SSHD	SSH Daemon
UART	Universal Asynchronous Receiver Transmitter

Table des matières

INTRODUCTION GENERALE	1
CHAPITRE 1 ETUDE THEORIQUE	3
1.1 INTRODUCTION	3
1.2 ETAT DE L'ART	3
1.2.1 <i>Recherches académiques :</i>	3
1.2.2 <i>Réalisations industrielles faites par des entreprises à but lucratif</i>	4
1.3 RAISONS DU CHOIX DU SUJET	5
1.4 PROPOSITION D'UNE SOLUTION	5
1.5 PRESENTATION DU CONCEPT	6
1.5.1 <i>Génération de l'information</i>	7
1.5.2 <i>Action</i>	7
1.5.3 <i>Déclenchement de l'actionneur et collecte de l'information</i>	8
1.5.4 <i>Traitement - Analyse de l'information</i>	9
1.5.5 <i>Archivage, visualisation et contrôle externe</i>	9
CHAPITRE 2 CONCEPT DU PROJET	12
2.1 INTRODUCTION	12
2.2 ARCHITECTURE DU SYSTEME	12
2.2.1 <i>Réseau local</i>	13
2.2.2 <i>Réseau externe</i>	16
2.3 PARTIE HARDWARE	16
2.4 PARTIE SOFTWARE	17
2.4.1 <i>Raspberry</i>	17
2.4.2 <i>NodeMCU</i>	37
CHAPITRE 3 REALISATIONS ET TESTS	42
3.1 INTRODUCTION	42
3.2 PRESENTATION DES MAQUETTES.....	42
3.3 L'INTERFACE WEB	44
3.4 L'ARCHIVAGE :	46
3.5 PROBLEMES RENCONTRES	46
3.5.1 <i>Le point d'Access</i>	47
3.5.2 <i>Le serveur FTP</i>	47
CONCLUSION GENERALE	49
<i>Annexe 1 : NodeMCU</i>	50
<i>Annexe 2 : Texas Instruments MSP430</i>	51
<i>Annexe 3 : Adafruit Feather M0 WiFi /ATWINC1500 [32]</i>	52
<i>Annexe 4 : Raspberry pi 3</i>	54
<i>Annexe 5 : Onion Omega2+ [35]:</i>	55
<i>Annexe 6 : Beaglebone Blue</i>	57
<i>Annexe 7 : Exemple d'utilisation du capteur DHT11 avec un NodeMCU</i>	58
<i>Annexe 8 : Exemple d'utilisation du capteur à ultrason avec un NodeMCU</i>	58
<i>Annexe 9 : Exemple d'utilisation d'une photorésistance avec un NodeMCU</i>	59
REFERENCES	60

Liste des figures

<i>Figure 0.1: Croissance rapide du nombre d'objets connectés à l'Internet [3]</i>	2
<i>Figure 1.1: Représentation fonctionnelle de capteur</i>	7
<i>Figure 1.2: Diagramme d'un Actionneur</i>	7
<i>Figure 1.3: Diagramme de bloc d'un System Embarqué d'un Nœud</i>	8
<i>Figure 2.1: implémentation du système pour une serre agricole</i>	12
<i>Figure 2.2: Architecture réseau maillé[12]</i>	13
<i>Figure 2.3: Diagramme de communication du protocole MQTT [14]</i>	14
<i>Figure 2.4: Communication par MQTT entre l'administrateur et les nœuds</i>	14
<i>Figure 2.5: Communication inter-nœuds par le protocole MQTT</i>	15
<i>Figure 2.6: GUI du logiciel Putty</i>	18
<i>Figure 2.7: CLI du Raspberry accédé depuis Putty</i>	18
<i>Figure 2.8: Bureau du Raspberry accédé depuis VNC Viewer</i>	19
<i>Figure 2.9: Fichier de configuration « /etc/gammu-smsdrc/ »</i>	20
<i>Figure 2.10: Fichier de configuration « /etc/sakis3g.conf »</i>	21
<i>Figure 2.11: exemple d'utilisation du programme sakis3g</i>	21
<i>Figure 2.12 : Exemple de lancement du programme « Ngrok »</i>	22
<i>Figure 2.13: Appareils connectés dans le compte « Ngrok »</i>	23
<i>Figure 2.14: connexion au Raspberry via l'adresse obtenue depuis « Ngrok »</i>	23
<i>Figure 2.15: CLI obtenu depuis l'adresse « Ngrok »</i>	24
<i>Figure 2.16 :Arborescences des fichiers down et flask</i>	25
<i>Figure 2.17: fichier rc.local</i>	26
<i>Figure 2.18: Organigramme du programme mqttsub.py</i>	28
<i>Figure 2.19: Organigramme du programme DB_handler2.py</i>	30
<i>Figure 2.20: Organigramme de la fonction program()</i>	30
<i>Figure 2.21: Organigramme de la fonction decision()</i>	31
<i>Figure 2.22: Organigramme du programme ftpclient.py</i>	32
<i>Figure 2.23: Organigramme du programme sms.py</i>	33
<i>Figure 2.24: Organigramme du sous-programme read2.py</i>	34
<i>Figure 2.25: Organigramme du programme writte.py</i>	34
<i>Figure 2.26: Fichier de configuration conf.ini</i>	35
<i>Figure 2.27: Organigramme du Programme Flask</i>	36

<i>Figure 2.28: Organigramme du programme du NodeMCU.....</i>	<i>38</i>
<i>Figure 2.29: Capteur DHT11 et son schéma de brochage[21].</i>	<i>39</i>
<i>Figure 2.30: Capteur NodeMCU HC-SR04</i>	<i>39</i>
<i>Figure 2.31: Principe de fonctionnement du capteur à ultrason [23].....</i>	<i>40</i>
<i>Figure 2.32: Représentation des signaux du capteur à ultrason [23].....</i>	<i>40</i>
<i>Figure 2.33: Représentation d'installation du capteur</i>	<i>41</i>
<i>Figure 2.34: Brochage de la photorésistance [24]</i>	<i>41</i>
<i>Figure 3.1: Maquette d'un nœud avec capteurs et actionneurs.....</i>	<i>42</i>
<i>Figure 3.2: Maquette d'installation du capteur HC-SR04</i>	<i>43</i>
<i>Figure 3.3: Deux nœuds installés</i>	<i>43</i>
<i>Figure 3.4: Page de visualisation des données</i>	<i>44</i>
<i>Figure 3.5: Page de configuration.....</i>	<i>45</i>
<i>Figure 3.6: Choix de l'application.....</i>	<i>45</i>
<i>Figure 3.7: Consultation de l'archive depuis la plateforme drivehq.com</i>	<i>46</i>
<i>Figure 3.8: Utilisation du protocole FTP depuis le CLI dans le réseau local</i>	<i>47</i>
<i>Figure 3.9: Visualisation de la commande netstat</i>	<i>48</i>
<i>Figure 3.10: Utilisation du protocole FTP depuis le CLI.....</i>	<i>48</i>
<i>Figure 3.11: Deuxième visualisation de la commande netstat.....</i>	<i>48</i>

Liste des tableaux

<i>Tableau 1.1: Couches d'un réseau d'internet des objets.....</i>	<i>6</i>
<i>Tableau 1.2: Tableau comparatif - Cloud hébergé en interne vs hébergé en externe</i>	<i>10</i>

Introduction générale

Le domaine de l'internet des objets (IoT) est un domaine très récent dans le monde, le développement de la science des réseaux informatiques et la technologie à basse échelle a dégagé davantage de bénéfices de la connectivité comme l'augmentation de capacité de traitement, stockage, la bande passante avec des coûts extrêmement à la portée de tout le monde.

Avec le développement de la science des réseaux informatiques, et avec le protocole IPV6 qui permet une connexion de 2^{128} (soit 340 282 366 920 938 463 463 374 607 431 768 211 456) personnes, processus, données et objets supplémentaires à l'Internet. Grace à cela, l'industrie est maintenant capable d'affronter ce monde qu'on peut dire vaste et multidisciplinaire.

Bien que ce domaine soit récent, on peut voir qu'il existe des systèmes d'internet des objets qui sont déjà fonctionnels et commercialisés. La première utilisation du terme « internet of things » était en 1999 quand Kevin Ashton, assistant chef de marque de « Procter & Gamble » a mentionné que l'IoT monté avec RFID et des capteurs seraient construits en 1999[1].

On a déjà rendu compte des bénéfices et avantages que l'informatique et l'électronique ont apporté à l'humanité, et avec un peu plus d'imagination on peut voir qu'avec l'IOT, on pourra bénéficier davantage et faciliter la vie dans tous les domaines (Médecine, industrie, Sécurité, Domotique, Agriculture, etc...).

L'IoT désigne l'environnement Internet qui consiste à générer, collecter, partager et utiliser des informations, en permettant à tous les éléments, tels que les personnes, les objets, les données, etc., d'être connectés à des réseaux filaires et non filaires. De plus, l'IoT désigne les technologies et les services qui englobent la génération de l'information (capteurs), collecte d'informations (composants, périphériques), partage d'informations (Cloud), maîtrise de l'information (Big Data) et logiciels d'application (softwares).

Le développement et la maîtrise des technologies et services ont permis aux intervenants de ce domaine de penser à étendre leurs idées et applications à des horizons qui n'ont pas été franchissables avant l'émergence de l'IoT et ses différentes technologies.

Actuellement, l'utilisation des nouvelles technologies est devenue un besoin indispensable, en effet, il est possible de constater (Figure 1) que le rythme de développement ne cesse d'accélérer. Selon Chuck Evanhoe, un futurologue et technologue qui connaît à fond le sujet de l'IoT : « L'IoT sera un formidable levier pour remonter de meilleures informations dans les environnements des consommateurs et

des entreprises. L'IoT aura, à mon avis, un impact dans tous les domaines. Tous les systèmes auxquels nous ne pensons pas dans notre vie quotidienne permettront d'augmenter la productivité des humains, ainsi l'effet ne sera pas cantonné à un seul domaine. »[2]. Tout cela, nous amène à dire dès maintenant que nous sommes en train d'entrer dans une ère où tout est connecté :

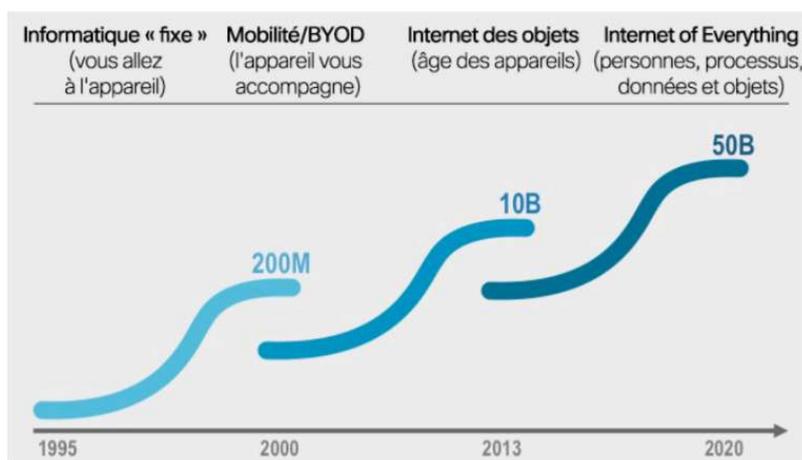


Figure 0.1: Croissance rapide du nombre d'objets connectés à l'Internet [3]

Ce projet est un essai de conception d'un système d'Internet des objets, ce système sera modélisé pour la prise de décisions basées sur les données acquises, ainsi que l'archivage, la connexion à l'Internet et au réseau GSM pour des fins de maintenance et d'urgences. L'aspect d'autonomie sera mis en place pour réduire les interactions humaines et garantir une mise marche courante du système.

Ce mémoire est organisé en trois chapitres. Le premier chapitre sera consacré à l'étude théorique d'un système de l'Internet des objets, état de l'art et les raisons du choix du projet, ainsi que la présentation du concept du prototype. Dans le deuxième chapitre, nous l'avons consacré pour la conception. Nous commencerons d'abord par la définition de l'authentification et l'explication du principe utilisé. Nous avons consacré le troisième chapitre à la réalisation et la présentation des tests et résultats obtenus, et enfin, un bilan et proposition des perspectives seront présentés dans la conclusion générale.

Chapitre 1 Etude théorique

1.1 Introduction

Ce chapitre sera consacré à l'étude théorique d'un système de l'internet des objets, nous présentons un état de l'art des travaux qui sont fait dans le domaine de l'internet des objets, ensuite nous citerons les raisons du choix du projet, ainsi que la présentation du concept du prototype proposé.

1.2 Etat de l'art

Dans la littérature, plusieurs travaux traitent le domaine de l'internet des objets, y compris, des recherches académiques, des brevets, des propositions des systèmes prototypes.

En outre, plusieurs intervenants dans le domaine de l'IoT sont passés de la théorie à la pratique en industrialisant leurs prototypes pour qu'ils soient exploités par le grand public.

1.2.1 Recherches académiques :

L'internet des objets peut être utilisé dans plusieurs domaines tel que la santé, l'industrie, l'agriculture, les services. Parmi ces domaines, nous citons quelques exemples :

Domaine de santé :

L'IoT promet de gérer la personnalisation des services de soins et peut maintenir une identité numérique pour chaque personne en se basant sur les caractéristiques biologiques, comportementales, sociales et culturelles uniques d'un individu, cela va impliquer le principe « les bons soins pour la bonne personne au bon moment » que l'IoT va bien l'adopter et approuver. L'utilisation de l'internet des objets peut être classée en deux types de soins personnalisés qui sont le soin clinique et la surveillance à distance, nous citons deux exemples :

- Observation d'une vieille personne à distance.
- Surveillance évolutive et continue des signes vitaux d'un malade à l'hôpital.

Une étude a été faite par A.Kulkarni et S.Sathe qui ont tentés dans leur article « Healthcare applications of the Internet of Things: A Review » [4] d'examiner les différentes applications de l'IoT dans les soins de santé personnalisés à des coûts abordables. Ils expliquent brièvement le fonctionnement de l'IoT et son utilisation en

conjonction avec des techniques de détection sans fil pour mettre en œuvre les applications de santé souhaitées.

Domaine de domotique et villes intelligentes :

Bien qu'il n'y ait pas encore de définition officielle et largement acceptée de la «ville intelligente», l'objectif final consiste à mieux utiliser les ressources publiques, en augmentant la qualité des services offerts aux citoyens, tout en réduisant les coûts opérationnels des administrations publiques.

La mise en place des villes intelligentes peut apporter certains nombre d'avantages dans la gestion et l'optimisation des services publics traditionnels tels que le transport et le stationnement, l'éclairage, la surveillance et l'entretien des espaces publics, la préservation du patrimoine culturel, la collecte des ordures ménagères, la salubrité des hôpitaux et des écoles.

A. Zanella et al. [5] dans leur article « Internet of Things for Smart Cities ». font une étude complète des technologies essentielles, des protocoles et de l'architecture pour un IoT urbain. Les auteurs présentent et discutent des solutions techniques et des directives de meilleures pratiques adoptées dans le projet Padova Smart City, en Italie, réalisé en collaboration avec la municipalité de la ville.

Domaine de l'agriculture :

L'IoT présentera des outils de choix pour la supervision de l'environnement des cultures, ce qui permettra une meilleure aide à la décision en agriculture. L'IoT servira non seulement à optimiser l'eau d'irrigation, l'usage des intrants et la planification des travaux agricoles, mais aussi, cette technologie peut être utilisée pour lutter contre la pollution (l'air, le sol et les eaux) et améliorer la qualité de l'environnement en général.

T.Megtit et al [6] proposent un système IoT pour la gestion d'une serre agricole pour mesurer la température, la lumière et l'humidité des serres. Les données de mesure seront mises à disposition des agriculteurs pour contrôler leur serre à partir d'un téléphone portable ou un ordinateur connecté à l'Internet.

1.2.2 Réalisations industrielles faites par des entreprises à but lucratif

Il existe des produits prêts à installer qui sont développés et fournis par des entreprises à but lucratif. Cependant, et vu que ces produits obéissent aux mesures de la propriété intellectuelle, il n'existe aucune documentation détaillée sur le principe de fonctionnement. Ainsi, on peut trouver que des manuels d'utilisateurs ou de la documentation (brochure, publicité etc...) destinée au grand public[7-9].

Par conséquent, le seul moyen de deviner le secret de ces produits est de exercer ce qu'on appelle le Reverse Engineering (ingénierie inverse), cependant, il faut faire attention de ne pas tomber dans des activités qui sont illégales.

Dans un deuxième volet, plusieurs prestataires de service dans le domaine des IoT ont émergés vu le besoin croissant des clients. Leurs principaux rôles sont :

- ❖ La conception des systèmes,
- ❖ La simulation,
- ❖ Réalisation des systèmes,
- ❖ Etc...

1.3 Raisons du choix du sujet

L'internet des objets est un domaine qui est actuellement dynamique et *foisonnant* d'idées, et chaque jour nous découvrons des applications industrielles qui visent à améliorer la vie des gens ; en effet Il ne se passe plus un jour sans que des objets connectés, plus insolites les uns que les autres, voient le jour

C'est dans ce contexte que nous nous plaçons, nous avons constaté qu'en Algérie, les objets connectés n'ont pas encore la cote, et les applications des IOT sont presque inexistantes.

Il y a eu quelques startups qui ont essayé d'investir dans ce domaine, mais qui ont rencontré beaucoup d'obstacles, nous citons par exemple le projet du robinet intelligent d'économie d'eau *Goutra* [10] et le projet d'*AquaSafe* [11], une entreprise qui propose aux industriels des outils connectés pour contrôler leurs rejets d'eaux usées.

Nous allons essayer de concevoir un système de monitoring générique autonome, bon marché, et utile ; On va le modéliser pour qu'il soit accessible à distance, et capable de supporter plusieurs applications en même temps.

Nous espérons que notre contribution permettra de :

- rendre cette recherche un guide complet pour les étudiants et chercheurs intéressés par le domaine de l'internet des objets.
- ouvrir de nouvelles perspectives et idées pour une continuation de ce projet.

1.4 Proposition d'une solution

La solution proposée est d'établir un système d'IoT générique qui est capable de supporter plusieurs applications en même temps, le cas échéant, d'avoir une méthode facile pour implémenter de nouvelles applications avec le moins d'interactions logiciels possibles. L'application qui concrétisera notre système est une application de surveillance et control d'une ferme de serres agricoles. Le système est capable d'avoir des applications en dehors des serres comme par exemple la surveillance et le contrôle logistique, surveillance du bétail ou ruches des abeilles etc...

Nous pouvons résumer les fonctionnalités principales du système dans les trois points ci-dessous :

- Le contrôle ou la prise de décision seront basés sur les données collectés depuis les capteurs du système. Le système pourra ainsi communiquer avec l'utilisateur (propriétaire – moniteur) en cas d'urgences.
- L'utilisateur peut visualiser les données acquises du proche et du loin à travers l'internet ou par SMS, activer/désactiver quelques fonctionnalités, changer l'application ou changer les critères de la prise de décision de l'application selon ses besoins.
- Avoir la possibilité d'accéder au système a distance pour des raisons de maintenance ou de mise à jour.

1.5 Présentation du concept

L'internet des objets est multi disciplinaire, il englobe l'électronique (hardwares), les techniques réseaux (architecture), les services (softwares, visualisation), il est à noter que ces derniers sont les organes pour un système IoT, mais cela varie d'un système à un autre tout dépend le scénario et les besoin de ce système, c-à-d, l'infrastructure d'un système IoT prenant en charge le traitement et la prise de décision sans interaction humaine peut être différente de celle d'un autre système qui n'est qu'une passerelle d'informations.

L'infrastructure de notre système est représentée selon les couches suivantes.

COUCHES	COMPOSANTS DE LA COUCHE
COUCHE INFÉRIEURE	1- Génération de l'information (capteurs, déclencheurs) 2- Action(actionneurs) 3- Déclenchement de l'actionneur et collecte de l'information (nœuds)
COUCHE INTERMÉDIAIRE	4- Traitement - analyse de l'information (administrateurs)
COUCHE SUPÉRIEURE	5- Archivage, visualisation et control externe

Tableau 1.1: Couches d'un réseau d'internet des objets

1.5.1 Génération de l'information

L'information est générée grâce à des capteurs installés sur site pour décrire l'état actuel sur le terrain. Par exemple pour avoir la température on utilise un capteur de température, Nous pouvons agir sur le système par un déclencheur en actionnant par exemple un ventilateur ou une pompe à eau. La Figure ci-dessous, représente la conversion d'une grandeur physique en un signal électrique par un capteur.

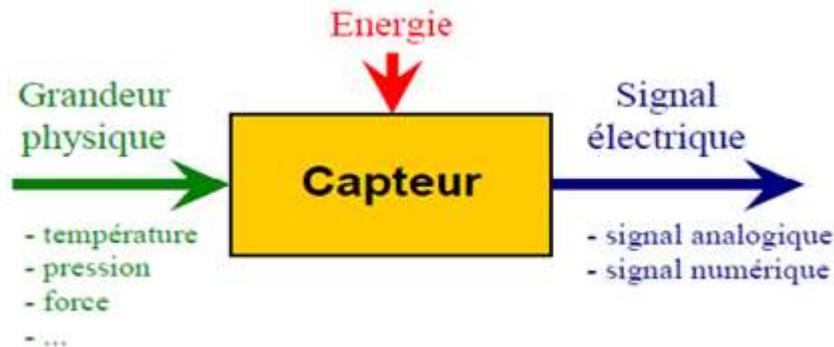


Figure 1.1: Représentation fonctionnelle de capteur

Présentons un scénario pour comprendre le cas d'un déclencheur :

Dans une salle d'attente gérée par un système de file d'attente automatique, le client entre, clique sur le bouton de la machine, la machine lui imprime un ticket portant son numéro file. En même temps la machine change l'image affichée sur l'écran qui montre le numéro du tour et le guichet de la queue actuel et le nombre de personnes en attente, quand l'agent du guichet termine avec un client il presse un bouton, l'image sur l'écran change et indique le numéro suivant et son numéro de guichet.

1.5.2 Action

L'action se fait par un actionneur qui sera déclenché grâce à un signal électrique, l'actionneur peut être soit électrique (écran, lampe, etc...), soit un périphérique électromécanique (moteur, haut-parleur etc...)

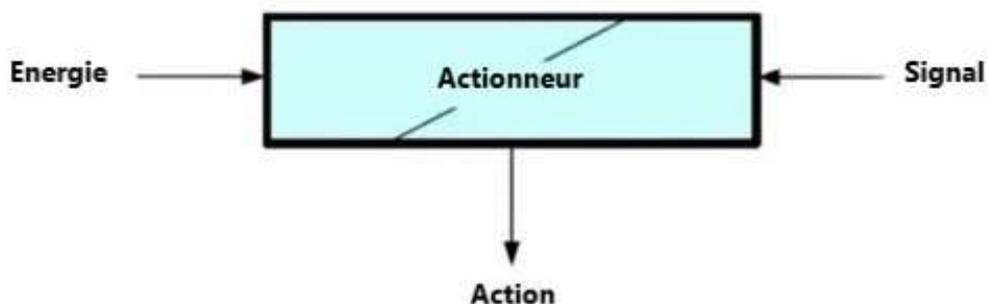


Figure 1.2: Diagramme d'un Actionneur.

Les capteurs et les déclencheurs seront connectés directement sur les nœuds.

1.5.3 Déclenchement de l'actionneur et collecte de l'information

Le déclenchement des actionneurs et la collecte des informations se font par des périphériques sur les Nœuds, ce sont des cartes possédant des modules d'entrées-sorties (I/O), les nœuds sont en général passifs (ne font pas la prise de décision), ils exécutent les ordres de l'administrateur.

Les nœuds sont des systèmes embarqués qui sont des ordinateurs indépendants du microcontrôleur qui utilisent des capteurs pour collecter des données ou effectuer une action. Ces systèmes embarqués peuvent communiquer avec l'administrateur, et peuvent dans certains cas communiquer entre eux. La communication peut être filaire ou non filaire, tout dépend des besoins et l'environnement où les nœuds sont placés.

La Figure 1.3 présente ce qu'un système embarqué peut porter comme modules supplémentaires au microcontrôleur.

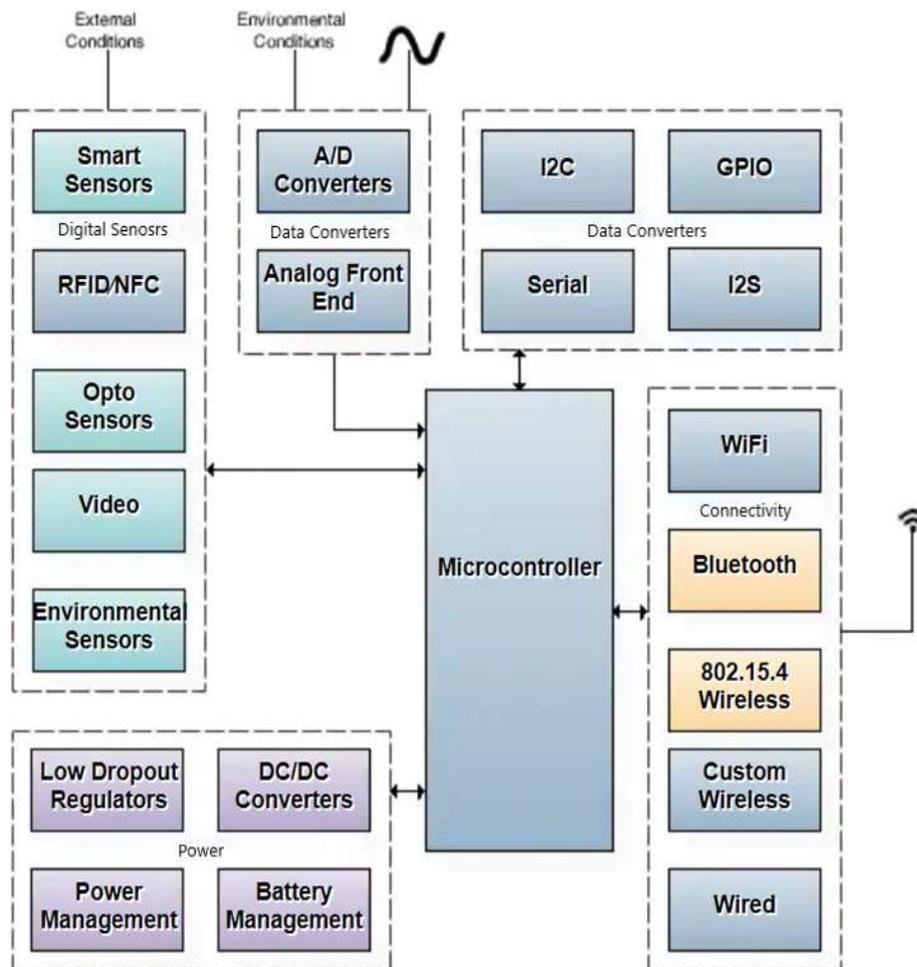


Figure 1.3: Diagramme de bloc d'un System Embarqué d'un Nœud

Nous présenterons quelques exemples de cartes à microcontrôleurs, veuillez voir l'annexe pour plus de détails:

- ❖ NodeMCU
- ❖ Texas instruments MSP430
- ❖ Adafruit Feather M0 WiFi /ATWINC1500

1.5.4 Traitement - Analyse de l'information

Le périphérique administrateur consacré pour la tâche d'administration sera plus performant par rapport aux nœuds (esclaves), il doit être de type microprocesseur, car, il sera le centre et le cœur de ce système, il aura la majorité du travail clé à effectuer.

Voici les tâches que l'administrateur doit effectuer :

- Gestion des nœuds.
- Contrôle des nœuds basé sur l'analyse des données collectées.
- Connexion à l'internet et GSM.
- Archivage local limité.
- Archivage via internet.
- Serveur local d'interface homme-machine.

Voici quelques cartes à microprocesseur adéquates pour la couche intermédiaire :

- ❖ Raspberry Pi 3
- ❖ Onion Omega2+
- ❖ Beaglebone Blue

1.5.5 Archivage, visualisation et contrôle externe

L'archivage sera établi à l'extérieur du site (sur internet), il suffit d'avoir une connexion pour établir cette tâche. On peut dire que l'administrateur sera une passerelle entre la couche inférieure (nœuds) et la couche supérieure (archives, etc...) en transmettant les données collectées en plus de son rôle de gestion, prise de décision etc...

a Archivage

Il existe deux options pour archiver les données en ligne, soit sur internet (Cloud), soit en installant un système de stockage privé hébergé en interne (exploité au niveau de l'entreprise), et les deux options ont leurs avantages et inconvénients.

	Avantages	inconvénients
Cloud hébergé en externe	<ul style="list-style-type: none">- Mises à jour et évolutivité- Redondance- Maintenance, sécurisation des données et mises à jour effectuées par le fournisseur- puissance de calcul- Coûts réduits- Paiement basé sur la consommation réelle.	<ul style="list-style-type: none">- Saturation de ressources et de la bande passante- Restrictions- Risque de mauvaise manipulation des données- Contraintes juridiques liées à l'hébergement des données (cas d'hébergement à l'étranger)
stockage en interne	<ul style="list-style-type: none">- contrôle et flexibilité totale- exploitation des services de manière plus flexible.- rapidité d'analyse d'informations.- déploiement plus rapide de nouvelles fonctions.- étendre / réduire les activités en fonction des besoins.- protection en interne des données à caractère confidentiel.	<ul style="list-style-type: none">- Travail et responsabilité supplémentaire.- Efficacité et sécurité relative aux performances du système installé.- Nécessité d'un savoir-faire technique.- Lourds investissements de départ.

Tableau 1.2: Tableau comparatif - Cloud hébergé en interne vs hébergé en externe

Nous pouvons constater qu'il est difficile de décider laquelle des deux solutions est la meilleure, en effet, le choix revient au constructeur de ce système qui est seul apte à estimer la solution la mieux adaptée en fonction de son cahier de charge (ampleur du projet, contrainte de l'environnement de travail, niveau de sécurité, coût initial de l'investissement, etc.)

b Visualisation

L'interface utilisateur (IHM) peut être réalisée avec une application web, cette interface peut garantir le monitoring et le contrôle total du système à travers l'internet. En absence d'une connexion internet, nous proposons un monitoring et contrôle minimal par SMS (Réseau GSM) pour les cas de secours.

L'interface web permet :

- Visualisation.
- Vérification de l'état du système
- Configuration
- Intervention manuelle
- Vérification du log
- Autres

Pour l'interface GSM :

L'interface GSM est choisi comme un outil de communication de secours seulement, mais il peut être une interface de commande et monitoring complète.

- Visualisation de l'état du system via SMS.
- Visualisation de dernier flux de données obtenu via SMS.
- Intervention manuelle via SMS.

Chapitre 2 Concept du projet

2.1 Introduction

Dans ce chapitre nous allons présenter en détail le système proposé dans le chapitre 1 sur trois parties, partie Architecture, Hardware et Software.

2.2 Architecture du système

Puisque nous allons appliquer notre système dans une ferme de serres agricoles, nous présentons un schéma illustratif de l'architecture physique de notre système :

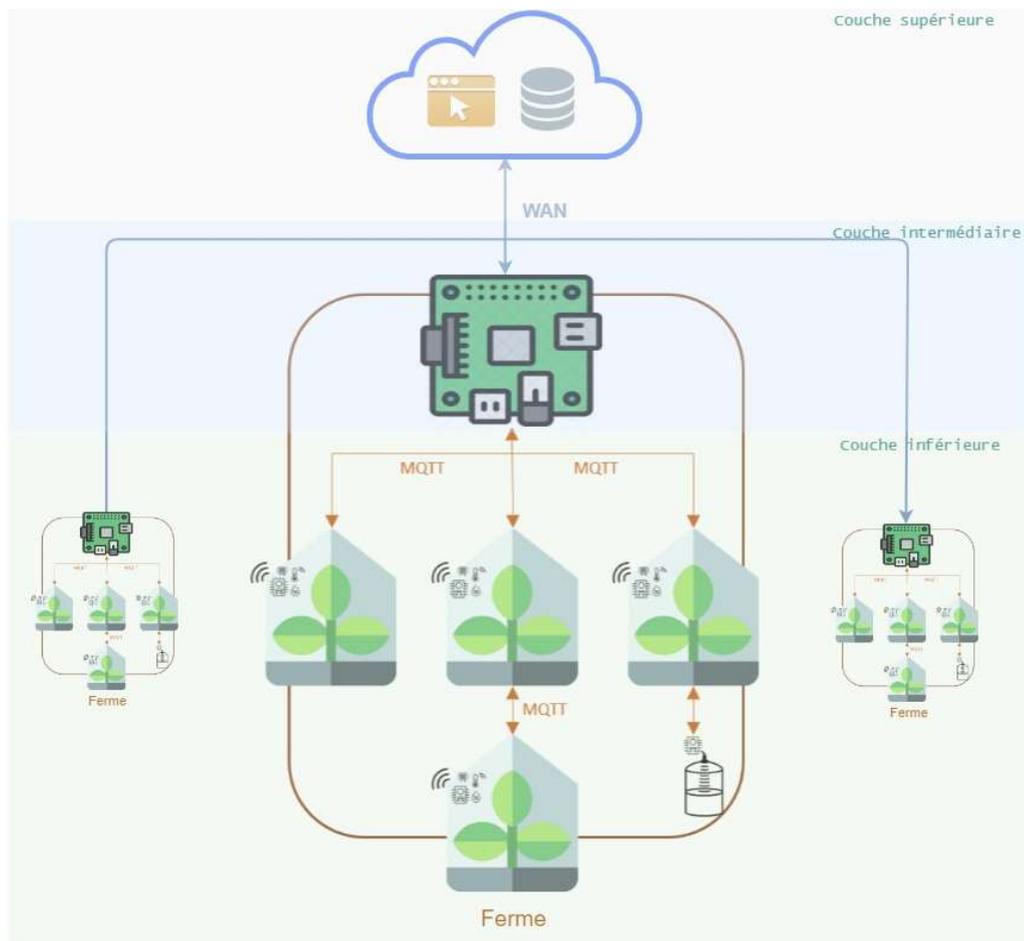


Figure 2.1: implémentation du système pour une serre agricole

La figure ci-dessus, nous montre que le système est reparti exactement sur les trois couches présentées précédemment dans le Tableau 1.1.

La couche inférieure représente les capteurs et les actionneurs qui sont installés à l'intérieur des serres, ils sont directement branchés aux nœuds. Ces nœuds représentent une passerelle entre ces capteurs/actionneurs et l'administrateur. La montée de la couche inférieure à la couche intermédiaire est garantie par la communication sans fil entre les nœuds et l'administrateur qui est gérée par le protocole MQTT, les nœuds peuvent communiquer entre eux-mêmes utilisant une topologie maillée.

2.2.1 Réseau local

La topologie utilisée dans le réseau local est une topologie maillée, nous avons choisi cette topologie car elle donne la possibilité d'insérer, de retirer ou de déplacer les nœuds d'une façon aléatoire sans affecter le réseau. Le réseau s'adapte d'une façon continue aux changements des positions des nœuds.

L'administrateur est le nœud racine de ce réseau, c-à-d, toutes les communications sortantes et entrantes du réseau local passent par l'administrateur.

La communication nœud à nœud est possible dans le cas où le nœud se trouve en dehors de la portée du réseau sans fil du nœud racine (l'administrateur) ou dans le cas d'une rupture inattendue d'une liaison directe entre les nœuds. Dans ce cas, ce nœud cherche le nœud le plus proche et se connecte avec lui, en outre le réseau maillé cherche à établir la meilleure liaison entre les nœuds afin de garantir une meilleure qualité de transmission[12].

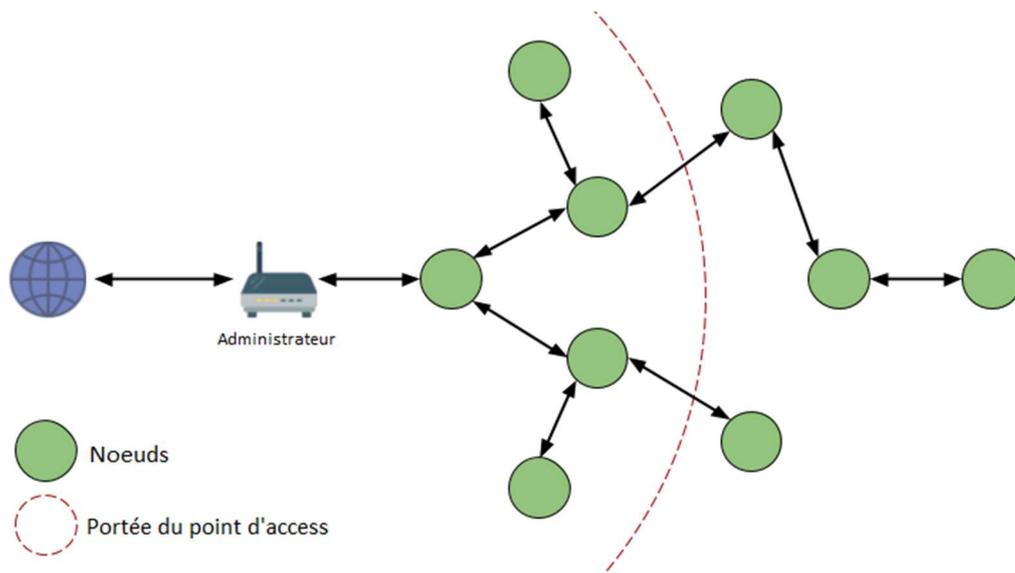


Figure 2.2: Architecture réseau maillé[12]

Les nœuds intermédiaires sont des nœuds passerelles qui permettent de passer l'information de ces nœuds distants, cela sera possible seulement avec les appareils capables de jouer le rôle de client et point d'accès Wi-Fi en même temps. Dans notre cas, nous avons choisi comme nœuds, des NodeMCU qui possèdent cette fonctionnalité (Soft-Access Point)[12].

a Communication MQTT:

MQTT (Message Queuing Telemetry Transport) est décrit comme un protocole de messagerie basé sur un broker à poids léger convenable aux appareils :

- à capacité de traitement et stockage faible comme celle dédiée à l'IoT,
- à communiquer sur une faible bande passante et aux réseaux peu fiables[13].

Le protocole MQTT utilise le principe de publication/inscription, ce principe permet aux utilisateurs de recevoir les informations qui les intéressent sans savoir qui est l'envoyant, il suffit de définir un point d'intérêt pour recevoir des informations à son propos, c'est ce qu'on appelle l'inscription à une topic (sujet), l'intermédiaire entre les publicateurs et les inscrits est appelé broker, c'est le serveur qui assure la gestion et la coordination entre les communicant par MQTT.

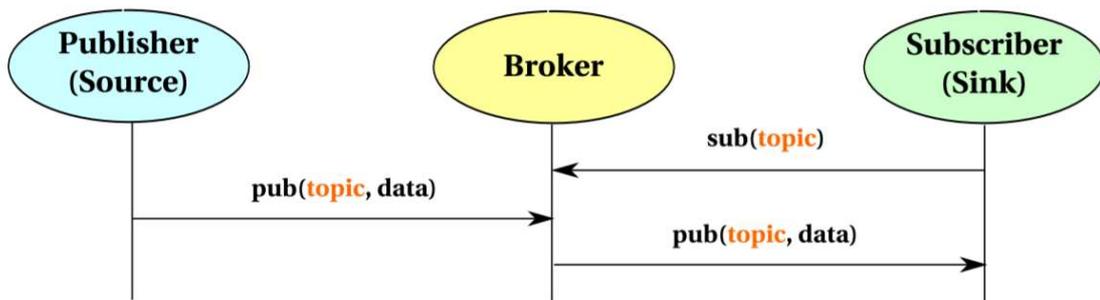


Figure 2.3: Diagramme de communication du protocole MQTT [14]

Nous allons utiliser le protocole MQTT pour établir la communication entre l'administrateur et les nœuds. Les nœuds seront les abonnés et l'administrateur sera un broker et un client en même temps.

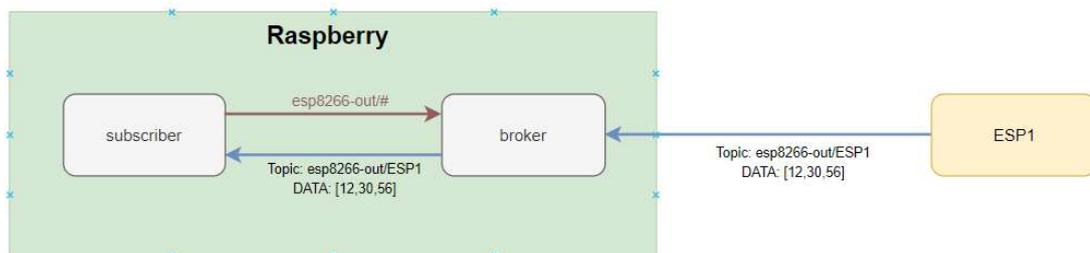


Figure 2.4: Communication par MQTT entre l'administrateur et les nœuds

Envoi des informations des capteurs vers le Raspberry :

Chaque nœud publie des informations avec un topic qui est relative à son adresse physique MAC, afin que l'administrateur puisse reconnaître chaque information qu'il reçoit à et à quel nœud elle appartient.

Un nœud ne communique qu'avec l'administrateur, l'interconnexion au serveur via d'autres nœuds ne se fait que dans le cas où ce nœud se trouve en dehors de la portée du point d'accès racine, la communication passe à travers d'autres points d'accès cachés générés par les nœuds, les nœuds passerelles conservent les paramètres de communication en aller et en retour (topic, data).

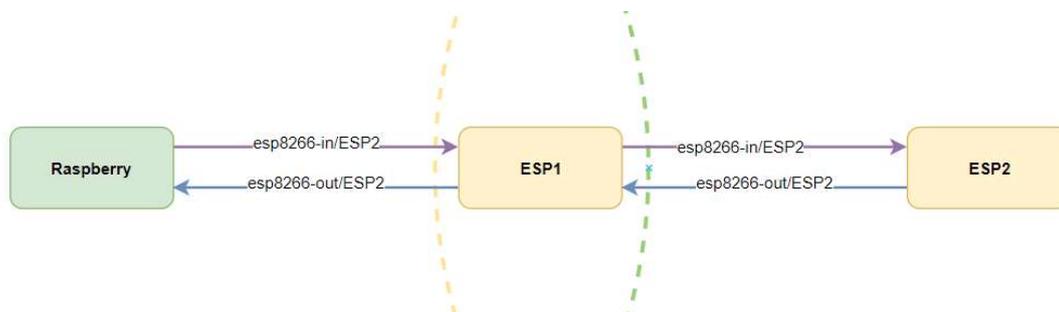


Figure 2.5: Communication inter-nœuds par le protocole MQTT

Formes des données envoyées :

Les données envoyées par le nœud sont sous forme de vecteur de 3 termes, chaque terme correspond à une donnée spécifique du capteur :

Exemple :

Voici le vecteur suivant comme exemple : [36, 9, 50]

Terme 0 = 36 : est la valeur du degré de température.

Terme 1 = 9 : valeur d'humidité en pourcentage.

Terme 2= 50 : volume du Réservoir d'eau en pourcentage.

Envoi des commandes d'action de l'administrateur vers les nœuds :

L'administrateur (Raspberry) envoie les commandes vers les actionneurs des nœuds en publiant dans le topic « esp8266/in/b1c07c/11649148 » sachant que le nœud (ESP) est inscrit exclusivement dans cette topic qui lui est propre.

2.2.2 Réseau externe

La communication de l'administrateur avec le monde externe se fait à travers le WAN à l'aide d'une clé 3G (Réseau 3G et GSM).

La communication au réseau WAN permet de communiquer avec la dernière partie du système qui est représenté par la couche supérieure dans le Tableau 1.1.

2.3 Partie hardware

Dans notre système il est possible d'utiliser tout type de capteurs, selon le besoin de l'utilisateur final. Pour la démonstration nous avons choisi deux types de capteurs :

- Capteur d'humidité et de température DHT11.
- Capteur à ultrason HC-SR04.

L'humidité et la température sont des paramètres très important dans une serre intelligente, ils permettent de mieux contrôler l'environnement assurant les bonnes conditions pour les cultures.

Actionneurs :

Pour des contraintes de temps et de faisabilité nous avons simulé les actionneurs par une LED pour représenter l'interaction physique du système dans le monde réel.

Nœuds :

- NodeMCU

Administrateur :

- Raspberry PI 3.

Autres :

- Clé 3G

Serveur d'archivage externe :

Le serveur d'archivage est un serveur hébergé sur internet, et la consultation des données se fait par un navigateur d'internet.

2.4 Partie Software

2.4.1 Raspberry

a Présentation et caractéristiques du Raspberry:

Raspberry Pi est l'une des plates-formes SBC les plus utilisées avec une variété d'utilisation. Il s'agit d'un petit ordinateur puissant, orienté vers l'éducation et le prototypage facile au service des scientifiques, des étudiants, des universitaires et des passionnés du développement de projets IoT. Elle possède la forme d'une carte de crédit avec de nombreuses fonctionnalités et a évolué au fil des années avec des mises à jour majeures et mineures au fil de l'évolution technologique[15].

b Systèmes d'exploitation :

Les cartes Raspberry Pi peuvent exécuter de nombreux systèmes d'exploitation, y compris Raspbian Linux, Windows 10 IoT Core. Il prend également en charge de nombreux langages de programmation tels que JavaScript, C / C++ et Python. Raspberry Pi a une grande communauté où les gens se réunissent pour en apprendre davantage sur la fabrication numérique avec Raspberry Pi. Raspberry Pi a donné lieu à divers projets intéressants, comme des systèmes de surveillance des patients et des systèmes domotiques[15].

Le système d'exploitation destiné aux cartes Raspberry est le Raspbian, Raspbian est un system linux Debian, il a été créé spécialement pour les cartes Raspberry par la fondation Raspberry[16].

Système choisi pour notre Raspberry :

Le système installé dans notre carte Raspberry est Ubuntu mate version 16.04, après plusieurs tests avec Raspbian, nous avons constaté qu'il ne dispose pas encore de certains drivers nécessaires à notre configuration hardware.

Utilisation du Raspberry :

Le Raspberry est équipé d'un processeur graphique et un connecteur HDMI, ce qui lui donne la possibilité d'afficher l'interface graphique de l'OS, et avec un clavier et souris nous pouvons le brancher avec un moniteur HDMI et l'utiliser comme un pc. Il est préférable de travailler avec Raspberry en utilisant un client SSH et VNC Viewer.

Logiciels linux requis :

Suivant les besoins de notre prototype, une configuration et installation de quelques logiciels est requise dans le système.

➤ Protocole SSH:

Il est possible d'utiliser le Raspberry depuis le pc à travers le Protocol SSH, il suffit d'installer le service SSH (sshd) au niveau du Raspberry et l'activer, nous pouvons ainsi nous connecter à l'interface de commande en ligne (CLI / terminal) à l'aide des logiciels Putty, MobaXterm, etc...

Le protocole SSH (également appelé Secure Shell) est une méthode de connexion à distance sécurisée d'un ordinateur à un autre. Il offre plusieurs options alternatives pour une authentification forte et protège la sécurité et l'intégrité des communications avec un cryptage renforcé. C'est une alternative sécurisée aux protocoles de connexion non protégés (tels que telnet, rlogin) et aux méthodes de transfert de fichiers non sécurisées (telles que FTP) [17].

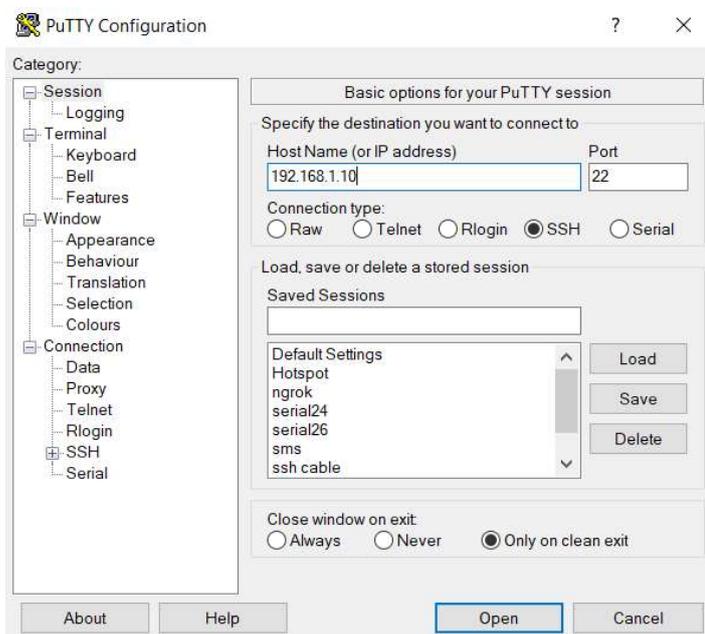


Figure 2.6: GUI du logiciel Putty

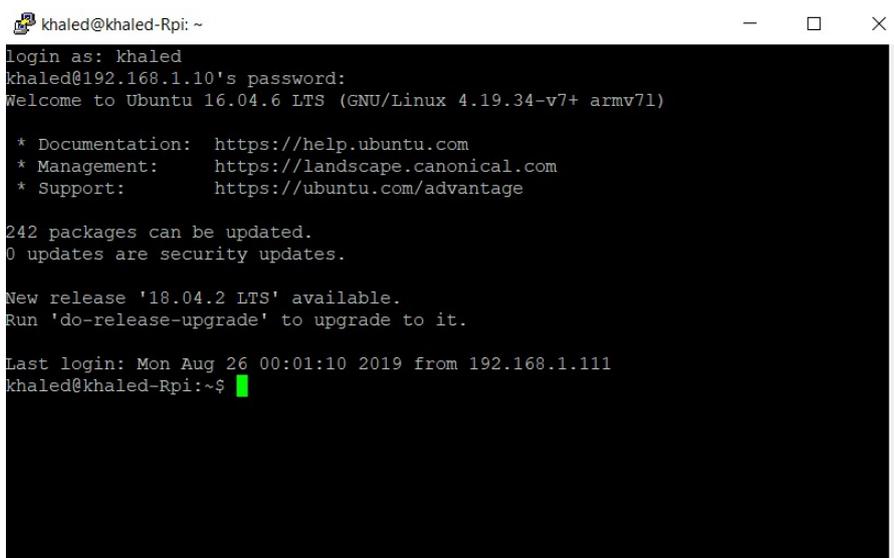


Figure 2.7: CLI du Raspberry accédé depuis Putty

Généralement, le protocole SSH est préinstallé et pré-activé dans la majorité des OS linux, dans le cas échéant nous pouvant l'installer et le lancer par les commandes suivantes:

```
apt-get install sshd  
service sshd start
```

➤ VNC Server :

Pour pouvoir voir l'interface graphique de l'OS du Raspberry depuis le PC il faut installer un logiciel de partage de bureau graphique. il suffit juste d'installer le serveur au niveau du Raspberry (VNC server par exemple) et le client (VNC Viewer par exemple) au niveau du pc.

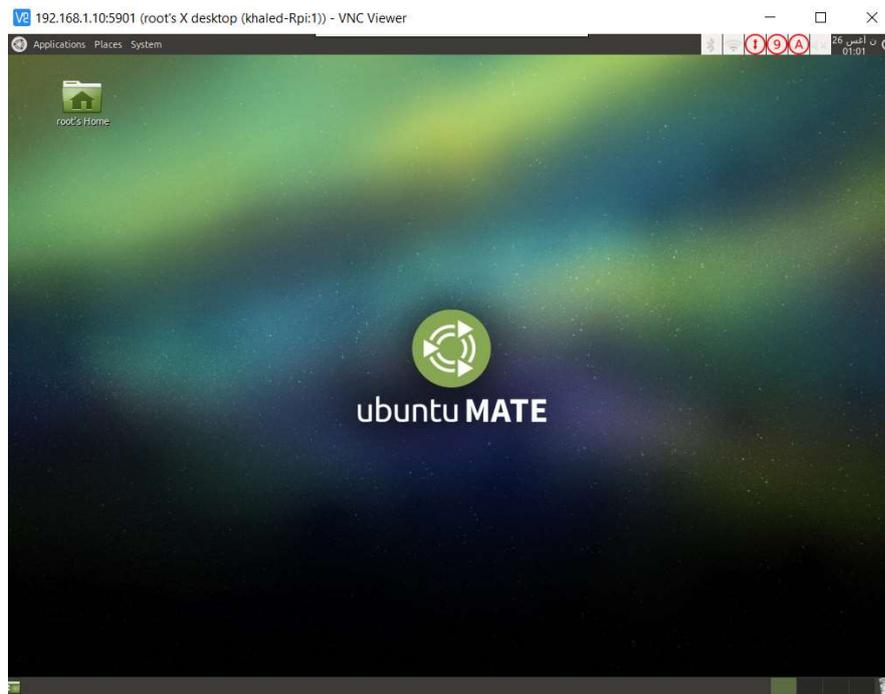


Figure 2.8: Bureau du Raspberry accédé depuis VNC Viewer

Voici la commande pour installer VNC sous Raspberry :

```
apt-get install vnc4server
```

Pour lancer VNC server on lance la commande suivante :

```
vncserver
```

A la première utilisation le logiciel va demander d'établir un mot de passe pour accéder au bureau depuis le client.

➤ Mosquitto, Mosquitto-clients :

Mosquitto est un logiciel (daemon) open source fournit par la fondation Eclipse, un broker (serveur) qui implémente le protocole MQTT. Avec son bon support communautaire, sa documentation et sa facilité d'installation sur tous les appareils (pc, serveurs, SBC) il est devenu l'un des logiciels MQTT les plus populaires.

Installation du broker MQTT Mosquitto sur le Raspberry :

```
apt-get install mosquitto
```

➤ Gammu-smsd :

Gammu-smsd est un programme qui analyse périodiquement les messages reçus dans le modem GSM, les stocke dans une mémoire définie et envoie également les messages mis en file d'attente dans cette mémoire. C'est l'outil idéal pour gérer de grandes quantités de messages reçus ou envoyés et les traiter automatiquement.

Installation de gammu-smsd:

```
sudo apt-get -y install gammu-SMS
```

Gammu nécessite une configuration selon les étapes suivantes :

- vérifier les fichiers « tty » dans le dossier « /dev », si le fichier ttyUSB0 existe, on ajoute son adresse au fichier de configuration, sinon, on ajoute le fichier ttyUSB1.
- Ouvrir le fichier de configuration « /etc/gammu-smsdrc »
- Coller la configuration suivante :

```
# Configuration file for Gammu SMS Daemon

# Gammu library configuration, see gammurc(5)
[gammu]
# Please configure this!
port = /dev/ttyUSB2
connection = at
# Debugging
#logformat = textall

# SMSD configuration, see gammu-smsdrc(5)
[smsd]
service = files
logfile = syslog
# Increase for debugging information
debuglevel = 0

# Paths where messages are stored
inboxpath = /var/spool/gammu/inbox/
outboxpath = /var/spool/gammu/outbox/
sentsmspath = /var/spool/gammu/sent/
errorspath = /var/spool/gammu/error/
```

Figure 2.9: Fichier de configuration « /etc/gammu-smsdrc/ »

➤ Sakis3g:

Sakis3G est un script qui fonctionne immédiatement pour établir une connexion 3G avec toute combinaison de modem ou d'opérateur. Il configure automatiquement les modems USB (clés 3G) ou Bluetooth et peut même détecter les paramètres de l'opérateur.

Pour plus de facilité, il est préférable de l'utiliser comme un daemon lancé à chaque démarrage.

Téléchargement et installation :

```
git clone https://github.com/Trixarian/sakis3g-source.git
cd sakis3g-source
./compile
./compile embedded
./compile stripped
cp build/sakis3gz /usr/bin/sakis3g
```

Configuration :

On crée le fichier dans le dossier /etc, et on colle les lignes suivantes de configuration de l'APN et l'ID de la clé :

```
APN="CUSTOM_APN"

CUSTOM_APN="internet"

APN_USER="internet"

APN_PASS="internet"

USBINTERFACE="0"

USBDRIVER=""

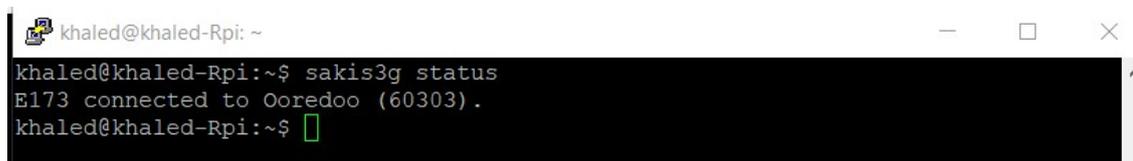
OTHER="USBMODEM"

USBMODEM="12d1:1436"
```

Figure 2.10: Fichier de configuration « /etc/sakis3g.conf »

Pour utiliser sakis3g on lance les commandes suivantes dans le CLI :

```
sakis3g help      : pour afficher le help de la commande sakis3g
sakis3g man       : pour afficher la page du manuel
sakis3g status    : pour afficher le statut
sakis3g connect   : pour connecter
sakis3g disconnect : pour déconnecter
```



```
khaled@khaled-Rpi: ~
khaled@khaled-Rpi:~$ sakis3g status
E173 connected to Ooredoo (60303).
khaled@khaled-Rpi:~$
```

Figure 2.11: exemple d'utilisation du programme sakis3g

➤ Ngrok :

Pour se connecter à l'administrateur depuis l'extérieur on utilise le service SSH qui utilise le port 22.

Nous avons constaté que les opérateurs de téléphonie mobile en Algérie ont mis des restrictions pour quelques services et ports TCP/UDP afin de diminuer la consommation inutile des ressources. Pour cela la connexion via le port 22 est impossible (contrairement à l'ADSL qui est possible). Le Service Ngrok permet de surpasser ces restrictions.

Ngrok est un logiciel de proxy inverse à tunnels multiplateformes qui établit des tunnels sécurisés à partir d'un terminal public à travers l'Internet, vers un service réseau local, tout en capturant tout le trafic pour une inspection et une relecture détaillées[18].

Pour l'utiliser, il faut créer un compte au niveau de son site officiel et obtenir une licence gratuite, puis le télécharger et l'installer avec les commandes suivantes :

```
Wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-arm.zip
./ngrok authtoken 5b9VEEAD2DU9Bxazeqsd8A_5w___1xQrjzyZZ7rK
```

La clé authtoken peut être obtenue depuis le compte, elle lie le compte avec l'appareil où Ngrok est installé

Utilisation :

```
./ngrok Tcp 22
```

```
root@khaled-Rpi: /home/khaled/Downloads/ngrok
```

```
ngrok by @inconstreveable
Session Status      online
Account             khaled NAHILIA (Plan: Free)
Update              update available (version 2.3.34, Ctrl-U to update)
Version             2.3.28
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           tcp://0.tcp.ngrok.io:16793 -> localhost:22

Connections         ttl    opn    rt1    rt5    p50    p90
0                   0      0      0.00  0.00  0.00  0.00
```

Figure 2.12 : Exemple de lancement du programme « Ngrok »

Nous pouvons consulter dans notre compte sur le site web de Ngrok les appareils connectés :

The screenshot shows the Ngrok dashboard with a sidebar on the left containing navigation links: Explore ngrok, Status (highlighted), Reserved, Auth, Team, Admin, Billing, and a 'Want more from ngrok? Upgrade now' link. The main content area is titled 'Tunnels Online' and features a search bar and a table of active tunnels.

URL	Client IP	Region	Established
tcp://0.tcp.ngrok.io:19249	105.101.203.138	us	Aug 30th, 2019 21:53:45 UTC

At the bottom of the dashboard, there is a copyright notice '© inconnshreveable 2019', a contact email 'contact@ngrok.com', and links for 'Terms of Service' and 'Privacy'.

Figure 2.13: Appareils connectés dans le compte « Ngrok »

Nous avons obtenu le lien public « 0.tcp.ngrok.io:19249 », il est possible maintenant de connecter via Putty.

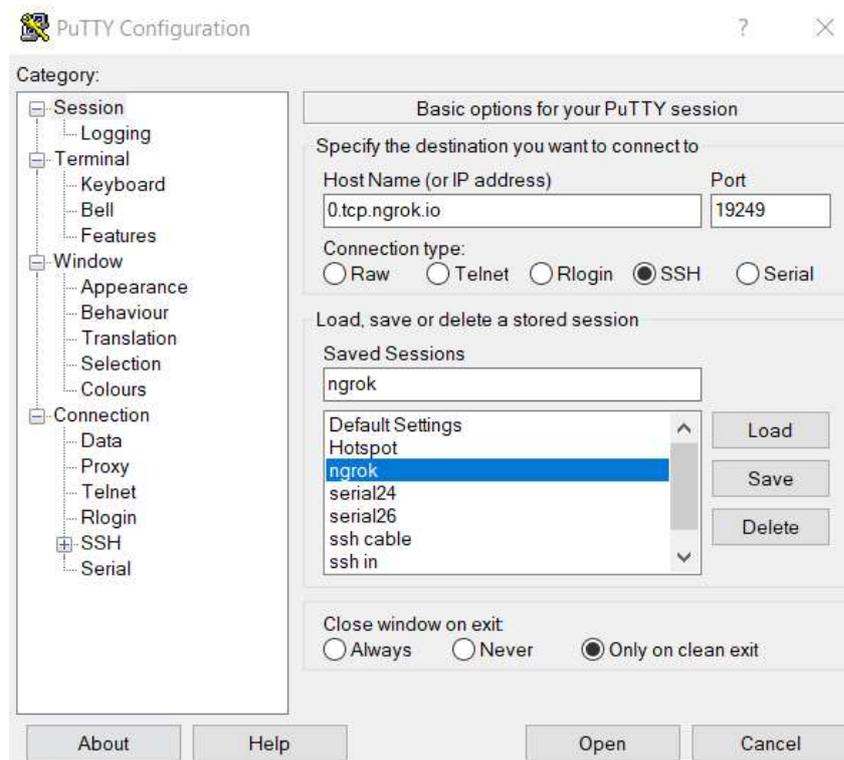
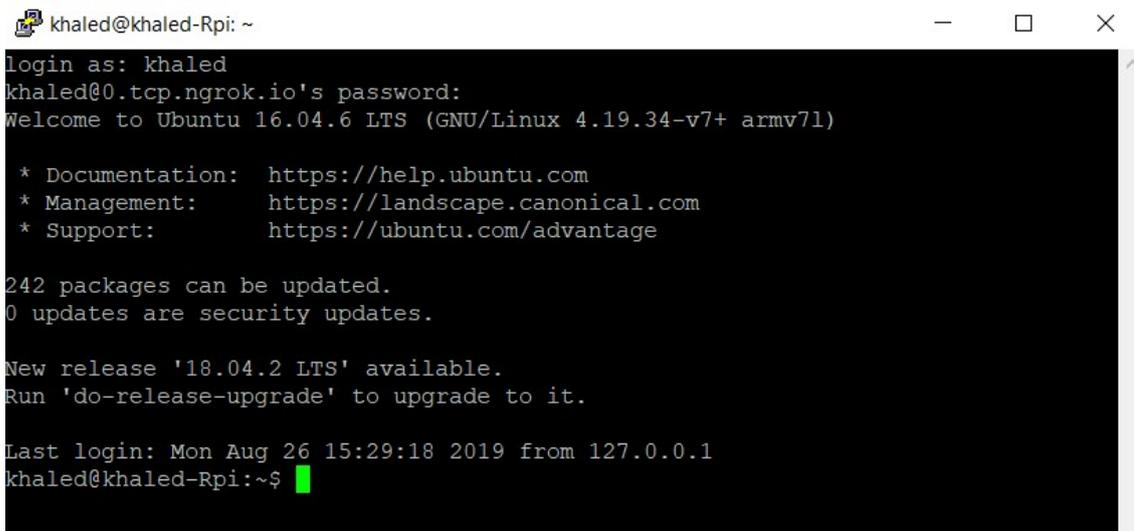


Figure 2.14: connexion au Raspberry via l'adresse obtenue depuis « Ngrok »

A terminal window titled 'khaled@khaled-Rpi: ~' with standard window controls. The terminal output shows a successful login for user 'khaled' on a system running Ubuntu 16.04.6 LTS. It displays system information, update status (242 packages can be updated), and a notification for a new release (18.04.2 LTS). The prompt is 'khaled@khaled-Rpi:~\$' with a green cursor.

```
khaled@khaled-Rpi: ~
Login as: khaled
khaled@0.tcp.ngrok.io's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.19.34-v7+ armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

242 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Aug 26 15:29:18 2019 from 127.0.0.1
khaled@khaled-Rpi:~$
```

Figure 2.15: CLI obtenu depuis l'adresse « Ngrok »

c Langages de programmation utilisés :

Les programmes utilisés dans le Raspberry sont écrit en langage python, nous avons utilisé python parce que c'est un langage de programmation robuste qui permet une utilisation facile des lignes de code, maintenance gérée de manière efficace et le débogage peut également être effectué facilement. Il a pris de l'importance dans le monde entier, car le géant de l'informatique Google en a fait l'un des langages de programmation officiels.

Avantages du langage python :

- Interactif
- Dynamique
- Portable
- Bibliothèques de support étendues
- Facile à apprendre et à coder
- base de soutien massive grâce à l'open source

L'interface web nécessite d'autres langages qui sont principalement utilisés pour la conception des applications web comme JavaScript.

JavaScript est utilisé principalement pour ajouter l'automatisation, l'animation et l'interactivité dans les pages web afin de donner plus de dynamisme et esthétique aux utilisateurs.

En plus du langage JavaScript, nous utilisons les langages HTML et CSS qui ne sont pas des langages de programmation mais des langages de balisage et d'esthétique pour la composition du contenu statique des pages web.

d Programmation de l'administrateur (Raspberry):

La programmation du Raspberry concerne :

- la gestion et contrôle des nœuds,
- le traitement et l'analyse de la base des données,
- l'archivage local,
- l'archivage externe,
- l'interface web.

Les programmes du Raspberry sont stockés dans les deux dossiers « down » et « flask ». Ces dossiers contiennent tous les fichiers et dossier nécessaires.

« flask » est le dossier contenant les programme et fichiers de la page web, et « down » comporte le reste des programmes et fichiers.

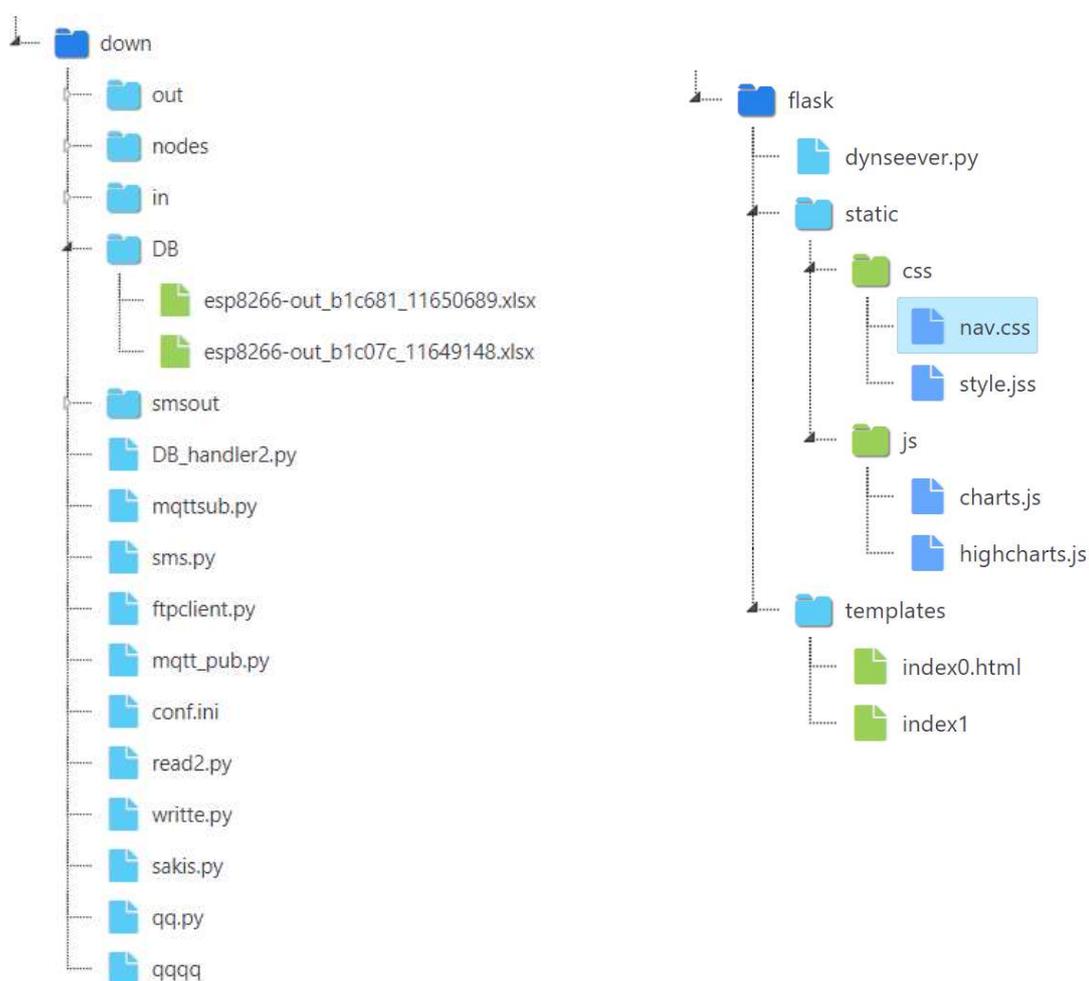


Figure 2.16 :Arborescences des fichiers down et flask

Ces programmes installés doivent être lancés au démarrage du Raspberry (auto startup). Pour les initialiser au startup, il existe plusieurs méthodes, la plus simple est de les mettre dans le fichier « rc.local » situé dans le dossier « /etc », ce dossier est un sous-service de « init.d », un des processus d'initialisations au démarrage de linux. « rc.local » est un fichier « bash », pour ajouter nos programmes il suffit d'ouvrir ce fichier et ajouter les commandes de lancement de ces programmes et logiciels.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

python /home/khaled/flask/dynserver.py > /home/khaled/serverlog &

sleep 2
#/usr/bin/modem3g/sakis3g --sudo start > /home/khaled/sakislog &
python /home/khaled/down/sakis.py > /dev/null &

sleep 3

cd /home/khaled/down
python /home/khaled/down/DB_handler2.py > /home/khaled/dbhandlerlog &
sleep 1

python /home/khaled/down/sms.py > /home/khaled/smslog &
sleep 1

python /home/khaled/down/ftpclient.py > /home/khaled/ftpclog &
sleep 1

cd /home/khaled/Desktop/
python /home/khaled/Desktop/aaa.py > /home/khaled/aaalog &

exit 0
```

Figure 2.17: fichier rc.local

Les programmes lancés automatiquement au démarrage du Raspberry sont :

- serveur de l'interface web
- programme sakis3g l'initialiseur de la clé 3g
- programme DB_handler2.py, gestionnaire de la BD.
- Programme SMS.py, le responsable de la communication SMS
- Le programme d'archivage ftp
- Le programme Ngrok (lancé dans aaa.py)

Pour lancer un programme python dans le CLI, il faut écrire la commande « python » suivi du nom du fichier python avec le chemin complet, suivi par d'autres commandes sur la même ligne dont la syntaxe est :

- « > » : est une commande pour canaliser les sortie du programme vers un autre fichier au lieu de les afficher sur le CLI,
- « & » : rend l'exécution en arrière-plan.

Exemple :

```
Python /home/khaled/down/sms.py > /home/khaled/smslog &
```

Python : Commande pour exécuter les fichiers python « .py ».

/home/khaled/down/sms.py : chemin complet du fichier sms.py.

> /home/khaled/smslog : Canaliser les sorties vers le fichier /home/khaled/smslog

& : Exécuter toute la commande en arrière-plan.

Environnement du travail « down » :

➤ « mqttsub.py » :

Mqttsub est un programme qui ouvre des communications MQTT, le Raspberry est en même temps un broker et un client, c-à-d, il reçoit les messages de la part des nœuds et de lui-même à l'aide d'un programme client, Mqttsub assure cette communication, il utilise d'autre sous-programmes pour compléter la tâche, voici le déroulement de ce processus :

- Importation des bibliothèques nécessaires.
- Importation du sous-programme « writte.py ».
- s'inscrire au topic 'esp8266-out/#', c-à-d, il entend tous les messages sortants des nœuds.
- En réception du message :
 - Lire le message reçu,
 - Utiliser le sous-programme « writte.py » pour mettre le message reçu dans un fichier temporaire.

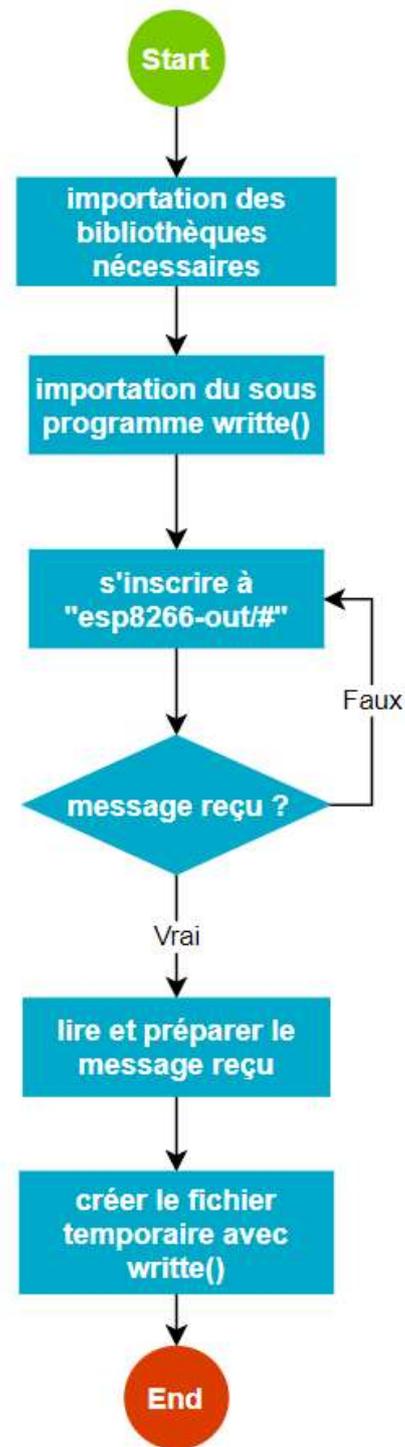


Figure 2.18: Organigramme du programme mqttsub.py

➤ « DB_handler2.py » :

Le *DB_handler* est le programme le plus important qui effectue plusieurs tâches.

Ce programme est responsable à la gestion des communications avec les nœuds, création et gestion des nœuds des tables de la base de données, prise de décision à base de données reçus soit par messages MQTT soit par SMS.

Voici le déroulement du programme DB_handler :

1. Définition des fonctions et importation des programmes externes :
Getnodes () : demander aux appareils d'envoyer les données.
Getconf : charger les paramètres depuis le fichier de configuration « conf.ini »
Decision : programme qui fait la prise de décision.
Read2 : lire les données depuis le fichier temporaire
Timee : programme qui retourne le temps en cours.
Newfram(k) : crée une fonction adaptée.
Program() : programme principale de Gestion de BD.
2. Charger les paramètres en lançant getconf().
3. Lancement de getnodes()
4. Repos pendant 5 secondes pour laisser aux nœuds le temps pour envoyer.
5. Lancer la fonction program() pendant une période définie depuis le fichier de configuration.
6. Retour vers (3)

Algorithme de la fonction program() :

1. Lire les données avec read2.py
2. Utiliser newframe() pour créer une nouvelle frame.
3. Si DB/<nom du nœud> existe :
Ouvrir la BD, ajouter nouvelle Frame, enregistrer et fermer le fichier.
4. Si non:
Créer une BD, ajouter la nouvelle Frame, enregistrer et fermer le fichier.
5. Lancer Decision().

Algorithme de la fonction decision()

1. Importation des données.
2. importation des critères depuis le fichier de configuration (temp-max,temp-min-hum-max, etc..).
3. Si data1 < temp-min: faire clignoter la LED 1 fois.
4. Si data1 >= temp-max et data2 > hum-min : faire clignoter la LED 2 fois.
5. Si data1 <= temp-max et data2 < hum-min : faire clignoter la LED 3 fois.
6. Si data 3 < vol-max et l'SMS est activé dans le fichier de conf :
 - Envoi d'un SMS d'alerte.
 - Faire clignoter la LED 4 fois

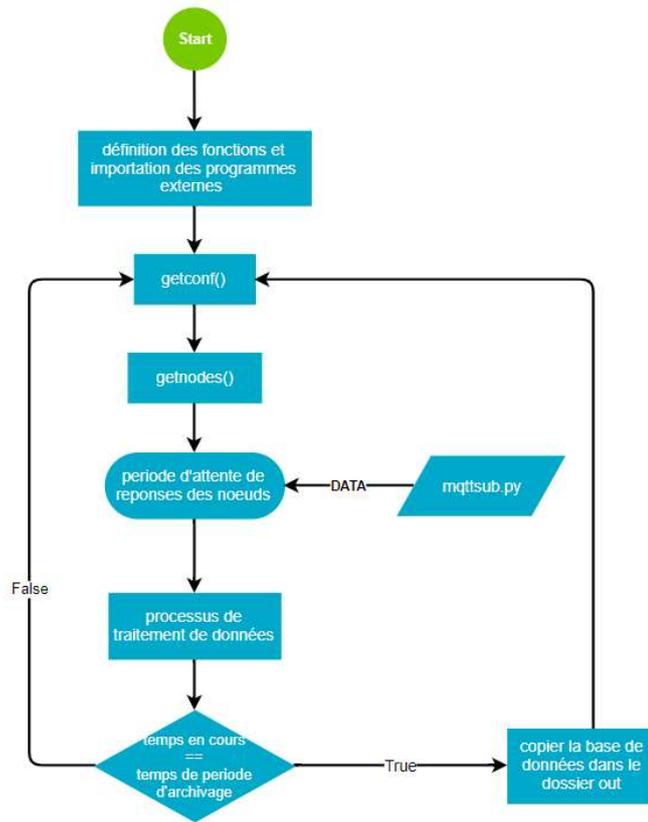


Figure 2.19: Organigramme du programme DB_handler2.py

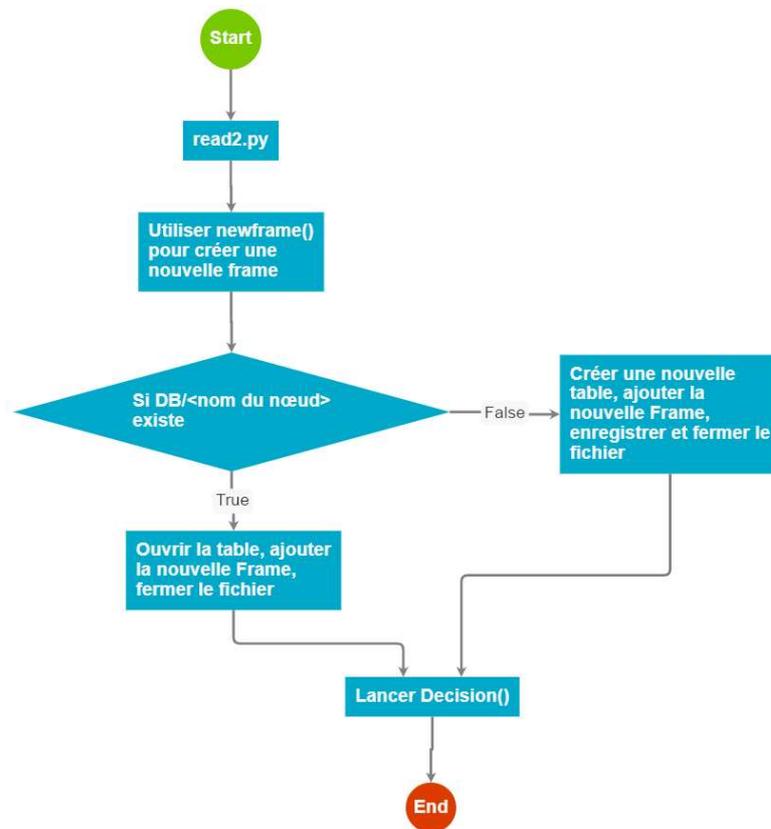


Figure 2.20: Organigramme de la fonction program()

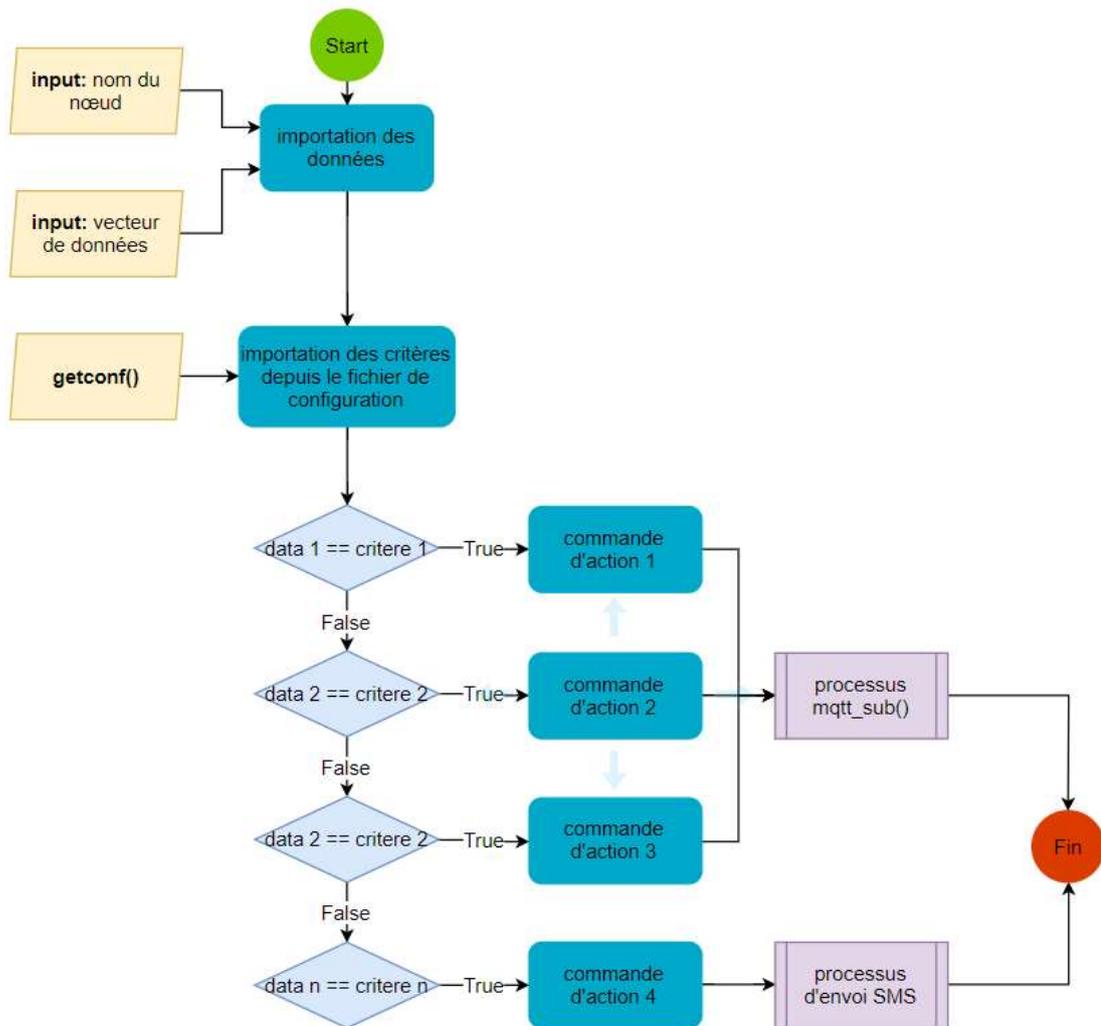


Figure 2.21: Organigramme de la fonction decision()

➤ « Ftpclient.py » :
 « ftpclient » est le programme responsable de la communication avec le serveur FTP d'archivage externe, ce programme est une simple automatisation d'envoi de fichiers via le protocole FTP, il consiste à surveiller le dossier « out » qui est la plus part du temps vide, du moment où le programme « DB_handler », copie la base de données dans ce dossier, « ftpclient » commence à envoyer ces fichiers.

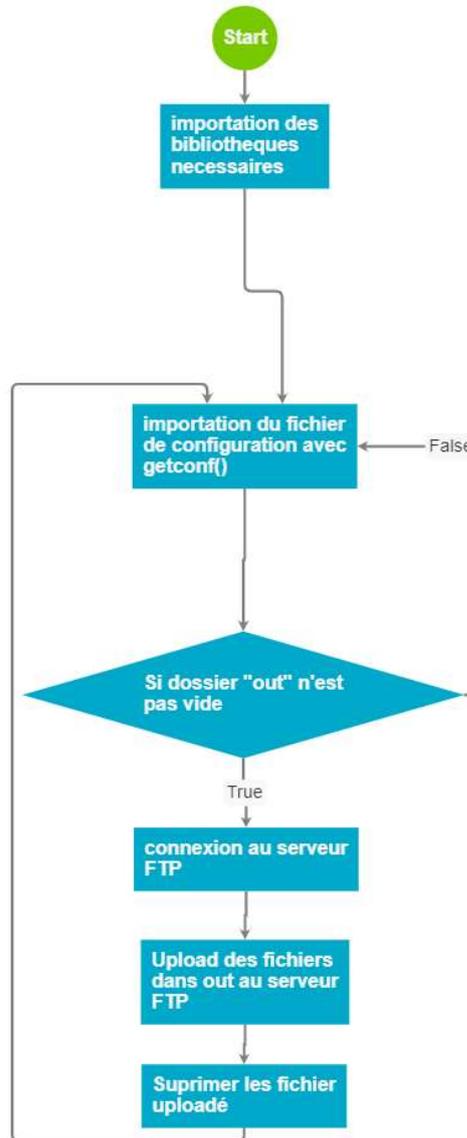


Figure 2.22: Organigramme du programme ftpclient.py

➤ « SMS.py » :

C'est le responsable de la communication par SMS, sa tâche est de surveiller la boîte de réception du logiciel « Gammu-smsd », lorsqu'un message est reçu, le programme retire le contenu du message et le numéro du téléphone envoyant. si le message est correcte il effectue la tâche relative à ce message, sinon, il revient au mode veille.

Les messages qu'on peut envoyer sont :

- « status » : pour avoir le contenu de fichier de configuration en message de retour.
- « secour on » : activer le mode secours qui consiste à envoyer les dernières données acquises de la part des nœuds d'une façon périodique ; la période se règle depuis le fichier de configuration « conf.ini ».
- « secoure off » : pour désactiver le mode secours.

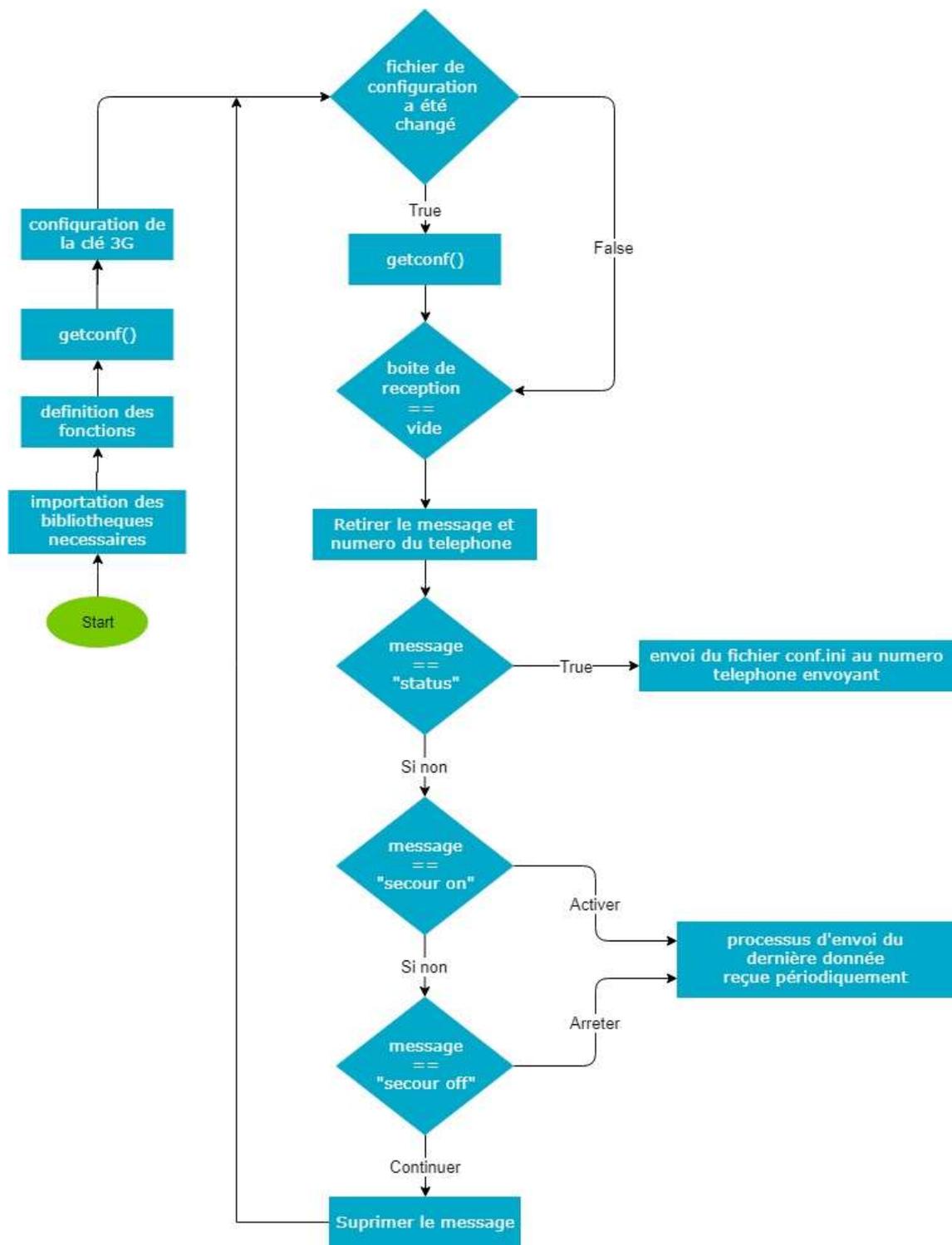


Figure 2.23: Organigramme du programme sms.py

➤ « read2.py » :

C'est un sous-programme qui aide à lire les fichiers temporaires créés par le programme « mqttsub.py »

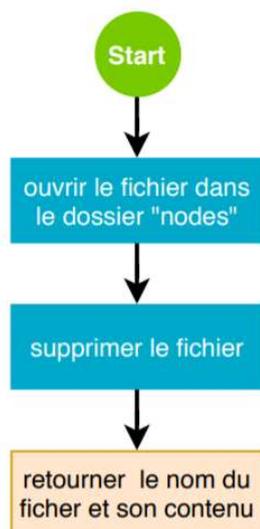


Figure 2.24: Organigramme du sous-programme read2.py

➤ « writte.py » :

C'est le sous-programme que « mqttsub.py » utilise pour créer les fichiers temporaires.

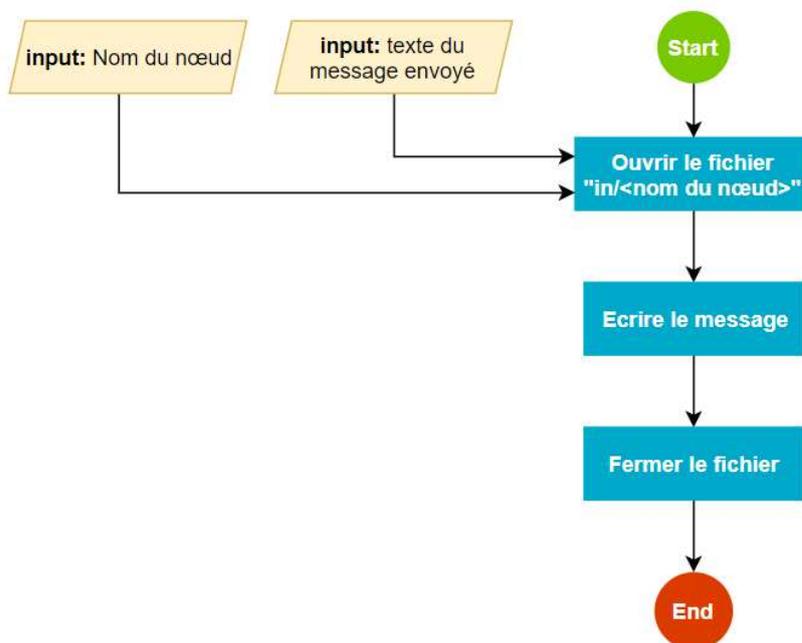
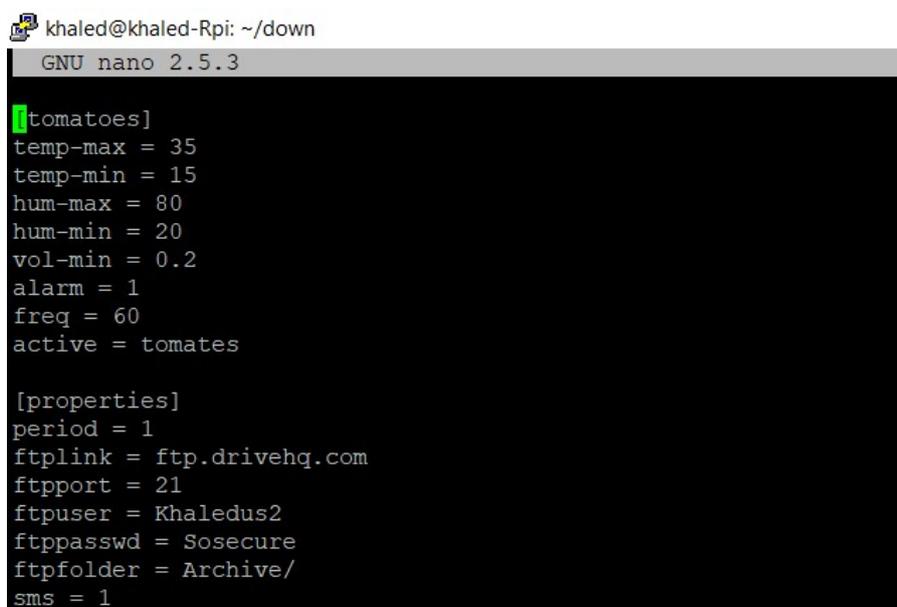


Figure 2.25: Organigramme du programme writte.py

Fichier de configuration « conf.ini » :

C'est un fichier externe de configuration et de paramétrage des critères de la prise de décision, l'activation/désactivation de quelques alertes, fréquence d'envoi de données (etc..).

On utilise le fichier de configuration pour donner la liberté à modifier quelques critères et paramètres sans avoir à les modifier dans le code et le recompiler, ce qui rend le programme plus dynamique.



```
khaled@khaled-Rpi: ~/down
GNU nano 2.5.3
[tomatoes]
temp-max = 35
temp-min = 15
hum-max = 80
hum-min = 20
vol-min = 0.2
alarm = 1
freq = 60
active = tomates

[properties]
period = 1
ftplink = ftp.drivehq.com
ftpport = 21
ftpuser = Khaledus2
ftppasswd = Sosecure
ftpfolder = Archive/
sms = 1
```

Figure 2.26: Fichier de configuration conf.ini

Environnement de travail « Flask » :

Flask est un Framework d'applications Web léger. Il est conçu pour permettre une mise en route rapide et facile, avec la possibilité d'évoluer vers des applications complexes, cela signifie qu'il s'agit essentiellement d'un ensemble d'outils et de bibliothèques facilitant la création d'applications Web en Python.

Nous avons choisi d'utiliser Flask pour les raisons suivantes :

- Possibilité d'interaction avec d'autres programmes python.
- Facile à apprendre et comprendre son principe de fonctionnement
- Léger et simple à utiliser.
- Possède d'un petit serveur web de tests et débogueur.
- Moins dépendant aux bibliothèques et outils tiers.
- Possibilité d'Intégration d'autres bibliothèques python.

La conception des pages web est enveloppée par la conception BackEnd et FrontEnd, le FrontEnd signifie l'interaction directe avec le client qui peut interagir avec un navigateur web comme les pages web et son affichage et interactions avec les liens et boutons. Le

BackEnd est la partie qui interagit indirectement avec le client comme les serveurs web, la communication avec la base de données et les requêtes HTML etc...

Puisque Flask est destiné pour la conception BackEnd, l'avantage, est qu'il est capable de gérer toute la page web depuis un seul programme python dans lequel on définit les routes des liens, les attributs, les redirections et les communications http , La communication avec la base de données et ainsi que d'autres.

Flask est basé sur le moteur de templates jinja2 (modèles de pages HTML) qui permet aux développeurs de produire des pages Web, contenant par exemple du code HTML de base et des espaces réservés pour que Jinja 2 les remplisse.

Outre le programme python principale, il faut ajouter des pages HTML, CSS nécessaires, ainsi que les programmes JavaScript qu'on aura besoin. Les fichiers statiques (CSS, Js) doivent nécessairement être séparés des templates HTML tels que l'arborescence illustrée à la Figure 2.16.

L'organigramme suivant qui illustre d'une façon simplifiée le principe de travail du Flask.

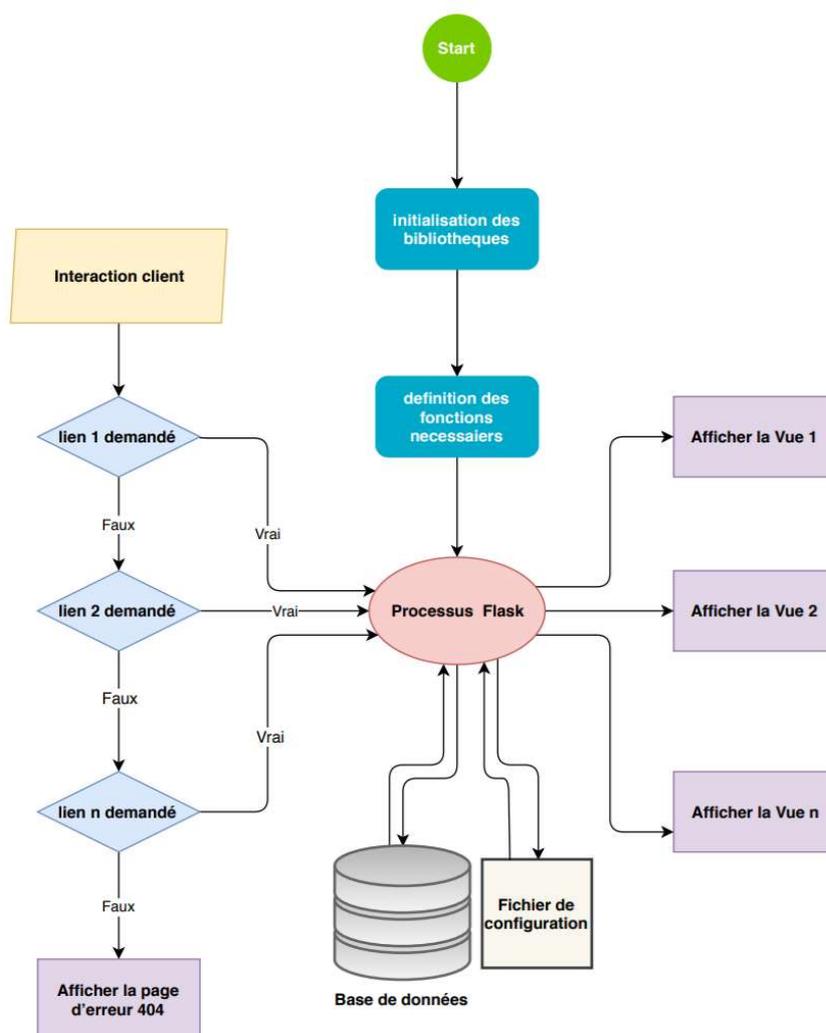


Figure 2.27: Organigramme du Programme Flask

2.4.2 NodeMCU

a Présentation et caractéristiques du NodeMCU :

NodeMCU est un petit module Wi-Fi puissant, pas cher et puissant pour la construction de projets IoT. Il est capable d'héberger une application ou de gérer toutes les fonctions de réseau Wi-Fi d'un autre processeur d'application.

Le NodeMCU peut fonctionner de manière autonome, comme il est possible de le piloter par une autre carte come Arduino.

NodeMCU est doté de la fonctionnalité d'auto-organisation du réseau où les nœuds peuvent analyser / sélectionner / se connecter / se reconnecter de manière autonome à d'autres nœuds et routeurs. Cette fonctionnalité permet à un réseau d'ESP de fonctionner avec un degré d'autonomie élevé en rendant le réseau robuste aux topologies et conditions de réseau dynamiques. Lorsque la mise en réseau auto-organisée est activée, les nœuds d'un réseau d'ESP peuvent exécuter les actions suivantes [12]:

- Sélection ou élection du nœud racine
- Sélection d'un nœud parent préféré
- Reconnexion automatique lors de la détection d'une déconnexion

b Outils et langages de programmation :

Le langage de programmation « Arduino C » est une sorte de langage C, en réalité la majorité des bibliothèques sont en C/C++, le sketch (script) subit des modifications mineures puis il sera transmis au compilateur C/C++ [19].

NodeMCU supporte la programmation et la compilation avec Arduino IDE et son langage « Arduino C », il existe d'autres IDE et langages pour programmer NodeMCU comme Micropython, C# ou le langage d'assembleur pour une programmation bas niveau.

Nous avons choisi Arduino parce qu'il est simple à comprendre et riche en bibliothèques des capteurs, ce qui rend la programmation facile ainsi que sa petite taille, les fonctionnalités du NodeMCU l'ont fait parfaitement convenir à notre système

c Programmation du NodeMCU :

La programmation du NodeMCU se résume dans les étapes suivantes :

- Vérification et établissement d'une connexion Wi-Fi et MQTT
- coordination avec les autres nœuds.
- Acquisition des données depuis les capteurs.
- Attendre la commande de l'administrateur pour envoyer les données

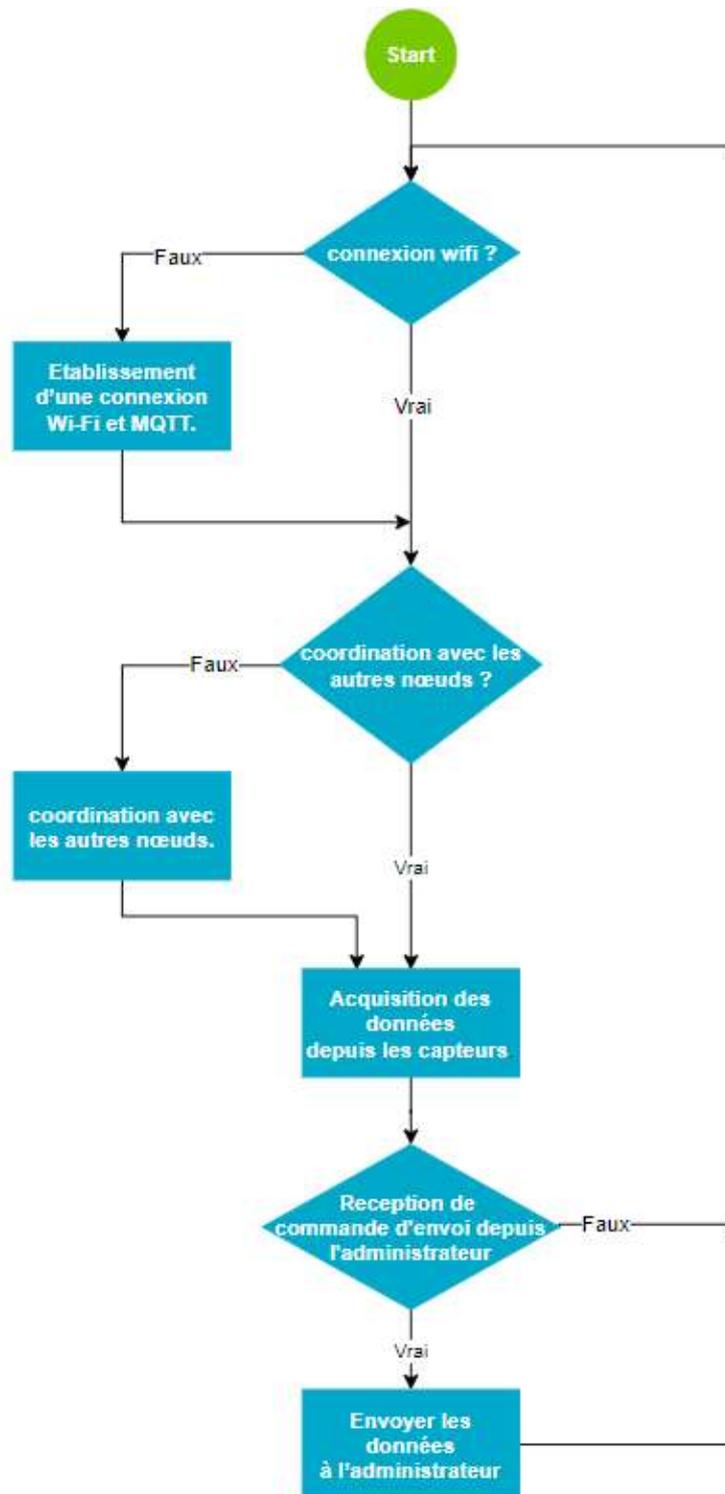


Figure 2.28: Organigramme du programme du NodeMCU

L'ajout des bibliothèques est nécessaire pour pouvoir utiliser quelques capteurs et fonctionnalités comme l'établissement du réseau maillé et le Protocol MQTT.

Principalement dans ce programme nous avons utilisé la bibliothèque « ESP8266MQTTMesh »[20]. Cette bibliothèque peut créer un réseau maillé entre les périphériques ESP8266, ce qui permettra à tous les nœuds de communiquer avec l'administrateur par le protocole MQTT.

Acquisition des données avec les capteurs :

➤ Capteur DHT11 :

Le capteur DHT11 fournit une information numérique proportionnelle à la température et l'humidité mesurée. Il est constitué d'un capteur de température (thermistances), d'un capteur d'humidité et un microcontrôleur qui s'occupe de faire les mesures, de les convertir et de les transmettre.

Il s'interface grâce au protocole OneWire qui permet de transmettre et de recevoir des données sur un seul fil, cette technologie utilisée par le capteur DHT11 garantit une grande fiabilité, une excellente stabilité à long terme et un temps de réponse très rapide[21].

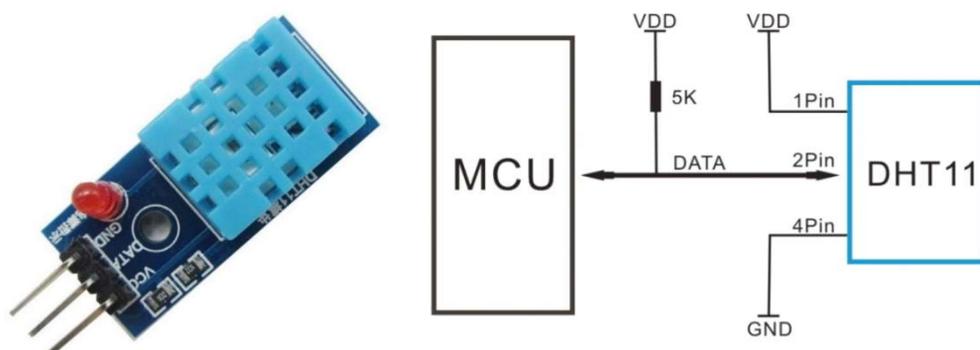


Figure 2.29: Capteur DHT11 et son schéma de brochage[21].

Du côté software, il existe plusieurs bibliothèques d'exploitation des module DHT11 qui permettent le retrait de la température et l'humidité, nous avons choisi la bibliothèque écrite par Mark Ruys [22]. Voir l'exemple d'utilisation de la bibliothèque dans l'annexe1.

➤ Capteur à Ultrason HC-SR04 :

Le capteur « HC-SR04 » fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm.



Figure 2.30: Capteur NodeMCU HC-SR04

Son principe de fonctionnement consiste à :

- Un court signal sonore est envoyé (inaudible car dans le domaine des ultrasons – environ 40kHz) ;
- Le son est réfléchi par une surface et repart en direction du capteur ;

- Ce dernier le détecte, une fois revenu à son point de départ [23].

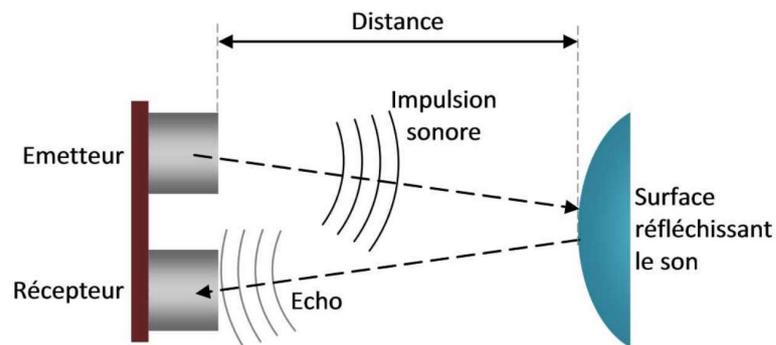


Figure 2.31: Principe de fonctionnement du capteur à ultrason [23]

La durée entre l'instant de l'émission et l'instant de la réception peut être mesurée. Le signal ayant parcouru 2 fois la distance entre le capteur et la surface (un aller-retour), on peut la calculer ainsi :

$$distance = \frac{vitesse\ du\ son}{2} \times durée \quad , \text{ la vitesse du son est environ égale à } 340\ m/s[23].$$

Fonctionnement :

- Le MCU envoie au capteur par un port numérique une courte impulsion (10µs environ) à l'entrée « trigger » du capteur ;
- cela déclenche l'émission d'un signal sonore très court (8 oscillations environ) ;
- lorsque ce signal est parti, la sortie « echo » du capteur passe à l'état haut ;
- dès que le signal sonore revient, il est détecté par le capteur et la sortie « echo », repasse à bas [23].

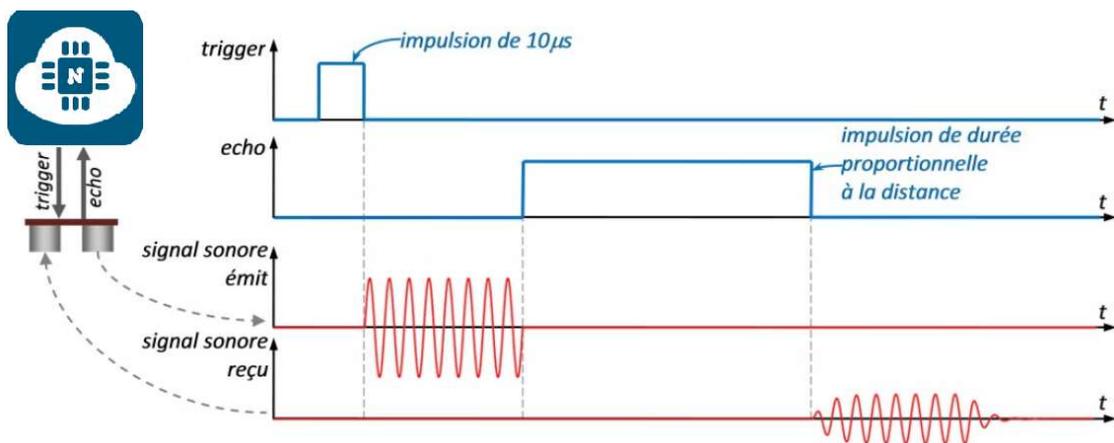


Figure 2.32: Représentation des signaux du capteur à ultrason [23]

Ce capteur est utilisé pour la surveillance du niveau d'eau du réservoir, il mesure la distance entre le capteur et la surface de l'eau. Avec une règle de trois, on peut savoir le pourcentage de remplissage d'eau dans le réservoir. Voir l'exemple d'utilisation dans l'annexe 2.

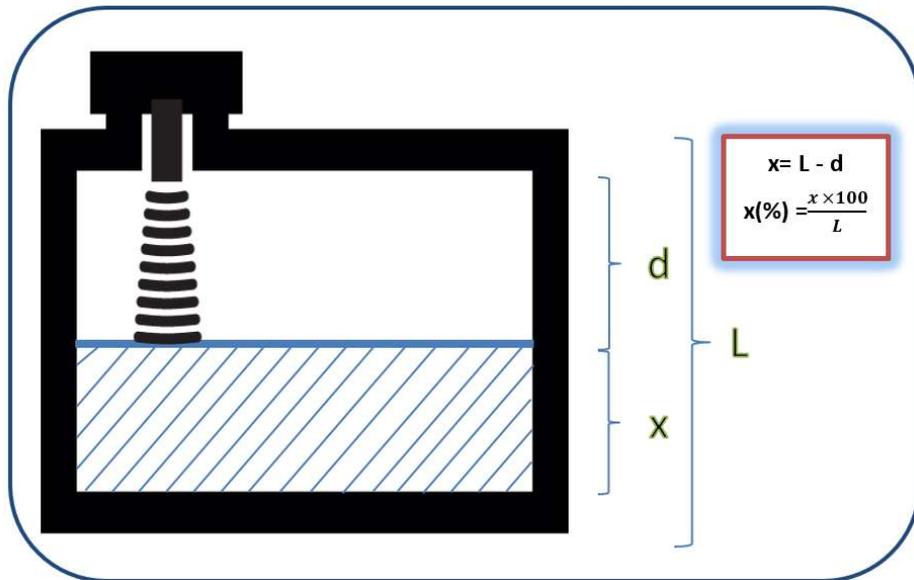


Figure 2.33: Représentation d'installation du capteur

➤ Photorésistance :

Une photorésistance ou LDR est un composant dont la résistivité dépend de la luminosité ambiante, c-à-d, plus l'obscurité augmente, plus sa résistivité augmente.

Pour déduire la valeur de sa résistivité nous appliquons le diviseur de tension dont le brochage sera comme la figure ci-dessous.

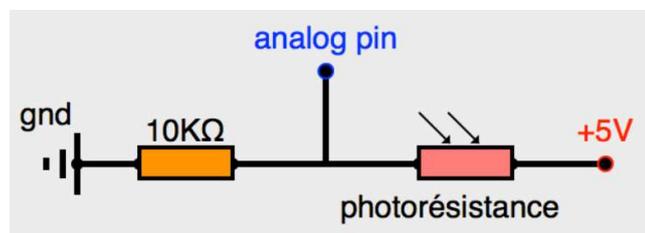


Figure 2.34: Brochage de la photorésistance [24]

La broche analogique est branchée à un convertisseur analogique numérique qui transforme une tension reçue entre 0V et 5V reçue en valeur numérique entre 0 et 1024. La valeur obtenue est convertit en pourcentage. Voir un exemple d'application dans l'annexe 3.

Chapitre 3 Réalisations et tests

3.1 Introduction

Dans ce chapitre nous allons présenter en premier lieu, les maquettes à l'aide du logiciel Fritzing ainsi que le système assemblé

Nous allons introduire tout le système en état de fonctionnement en présentant l'interface web et la consultation d'archive.

En second lieu, nous allons présenter l'interface web et le fonctionnement du système.

Et en dernier, nous discuterons les conditions d'application du système et les problèmes rencontrés.

3.2 Présentation des maquettes

Les maquettes sont réalisées avec le logiciel Fritzing afin de donner une présentation claire et compréhensible.

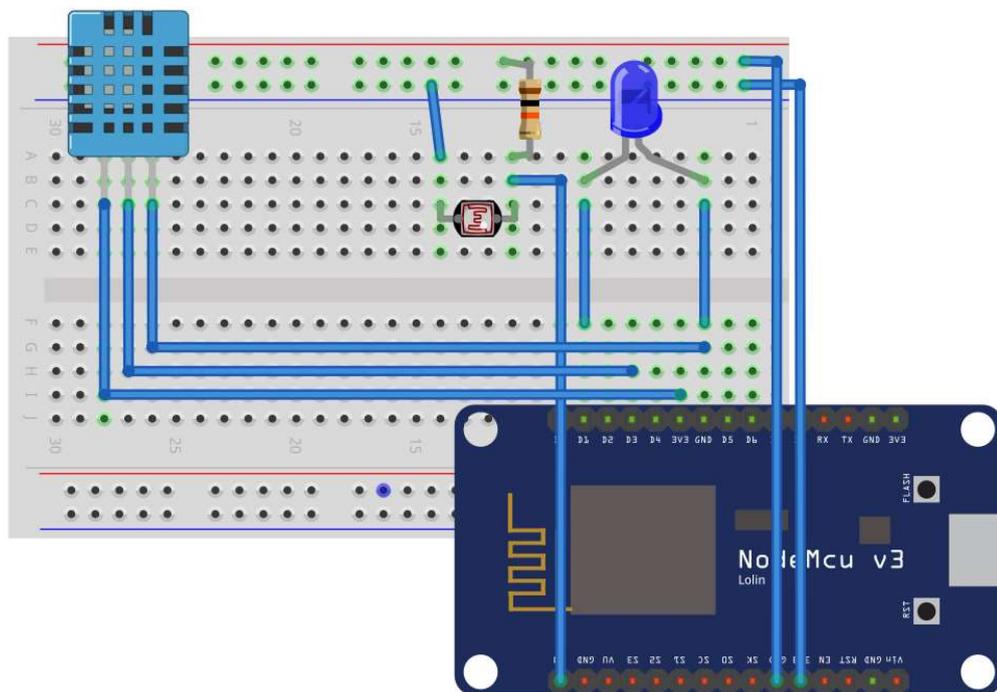


Figure 3.1: Maquette d'un nœud avec capteurs et actionneurs

La Figure 3.1 montre la maquette faite d'un nœud, les capteurs (DHT11, Photorésistance) la photorésistance remplace le capteur à ultrason HC-SR04 pour une meilleure présentation de résultats dans la plateforme.

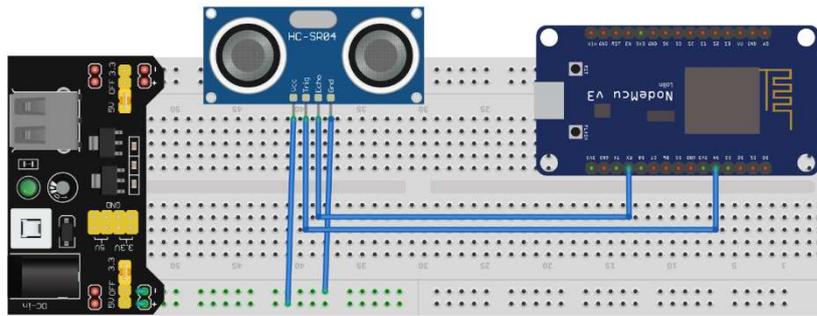


Figure 3.2: Maquette d'installation du capteur HC-SR04

Le capteur à ultrason nécessite une alimentation de 5 volts, NodeMCU ne fournit que 3,3V, alors, nous avons ajouté une alimentation externe de 5V.

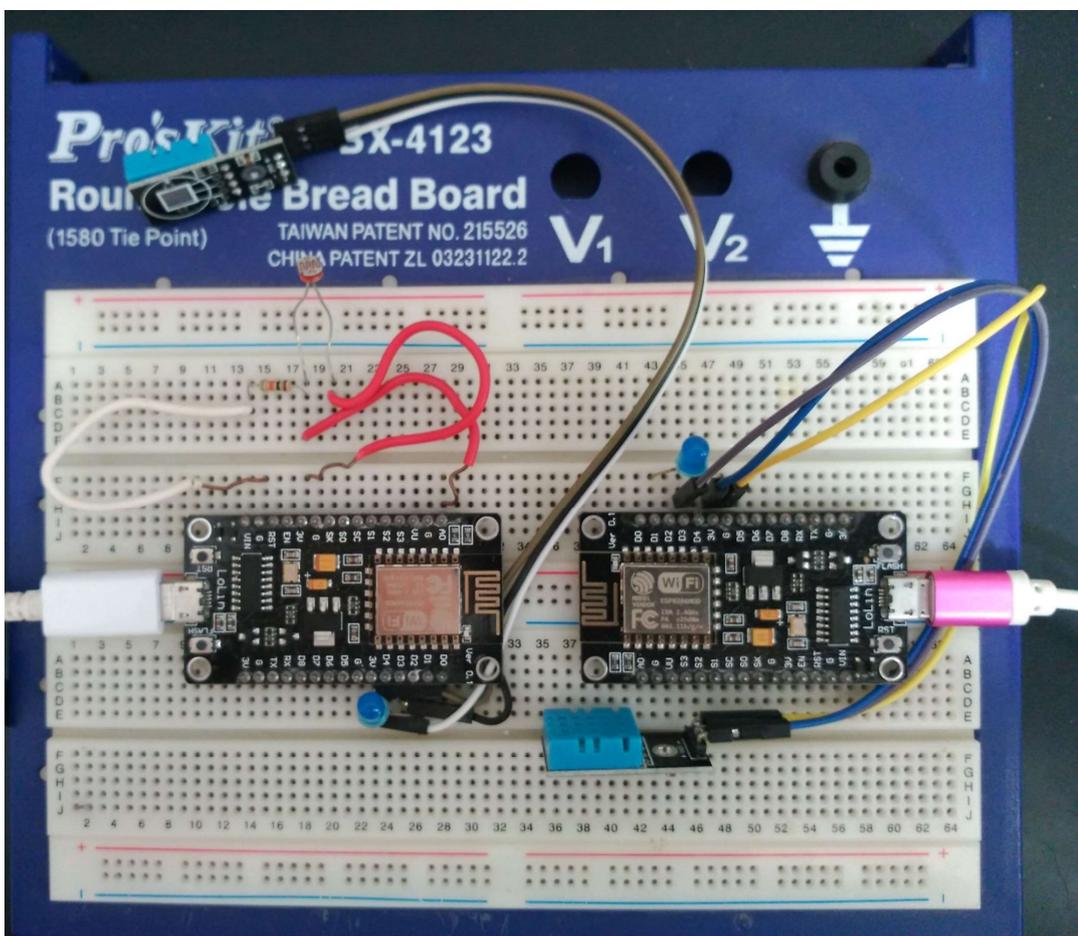


Figure 3.3: Deux nœuds installés

3.3 L'interface WEB

L'accès à l'interface web peut être réalisé sur le réseau local, avec l'adresse fixe attribué à l'administrateur(Raspberry). L'interface que nous avons créée comporte deux pages, une page pour la visualisation des données, et une autre pour la configuration.

L'interface web communique directement avec la base de données qui est enregistrée dans l'administrateur et avec le fichier de configuration pour lire ou écrire des nouveaux paramètres. La page est dynamique, car le système répond d'une façon directe sans avoir besoin de le rafraichir manuellement.

Page de visualisation :

La page de navigation permet de voir toutes les données enregistrées dans la base de données, ainsi la dernière données acquise pour chaque nœud, la barre de navigation donne la main pour choisir quel le nœud à visualiser.



Figure 3.4: Page de visualisation des données

Page de configuration :

The screenshot shows a web interface for configuration. At the top, there are navigation links for 'Home' and 'Configuration'. Below that, there are two file paths: 'esp8266-out_b1c681_11650689.xlsx' and 'esp8266-out_b1c07c_11649148.xlsx'. The main content area is divided into two sections: 'Application active : .' and 'Propriétés :'. The 'Application active' section has a dropdown menu labeled 'Application' with a blue highlight. Below it are several input fields for 'temp-max', 'temp-min', 'hum-max', 'hum-min', and 'vol-min', each with a numerical value. The 'Propriétés' section includes a toggle for 'SMS', a numeric input for 'freq', and a section for 'Archivage par FTP' with fields for 'ftppuser', 'ftplink', 'ftppassword', 'ftpport', and 'period'. A button labeled 'Appliquer' is at the bottom center.

Figure 3.5: Page de configuration

La page de configuration permet de faire quelques paramétrages par l'administrateur. La partie gauche de la page permet de changer les critères de décision on se basant sur l'application (tomates, patates, Custom), il est possible de choisir une des applications prédéfinies à travers le menu déroulant bleu. Une fois sélectionnée, les valeurs seront changées dans les champs des critères. Pour d'autres applications, il est possible de configurer une application personnalisée en modifiant les champs manuellement.

This screenshot shows the same configuration page as Figure 3.5, but with the 'Application' dropdown menu open. The menu lists three options: 'Tomates', 'Patates', and 'Custom'. The 'Application active' text now reads 'tomates'. The 'temp-max' input field is visible with the value '35'.

Figure 3.6: Choix de l'application

La partie droite permet de modifier certaines propriétés de fonctionnement de l'administrateur en personnalisant la fréquence d'acquisition de données, activer ou désactiver les alertes SMS, changer le serveur d'archivage ou changer la période d'archivage.

3.4 L'Archivage :

Le système est conçu pour transférer les données collectées d'une façon périodique vers un serveur externe. Pour effectuer ce travail, nous utilisons le Protocol FTP.

Le Serveur est hébergé sur Internet ftp.drivehq.com avec un service gratuit pour essai, le service gratuit est suffisant pour un usage non excessif. L'administrateur Raspberry est le client. Il est possible de changer le serveur et la période de l'envoi depuis le fichier de configuration « conf.ini » ou par l'interface web.

La consultation des fichiers - uploadés - est possible depuis le site web du serveur www.drivehq.com. La plateforme de consultation donne la main à beaucoup de fonctionnalités comme la visualisation des fichiers, le téléchargement, déplacement/suppression des fichiers, etc...

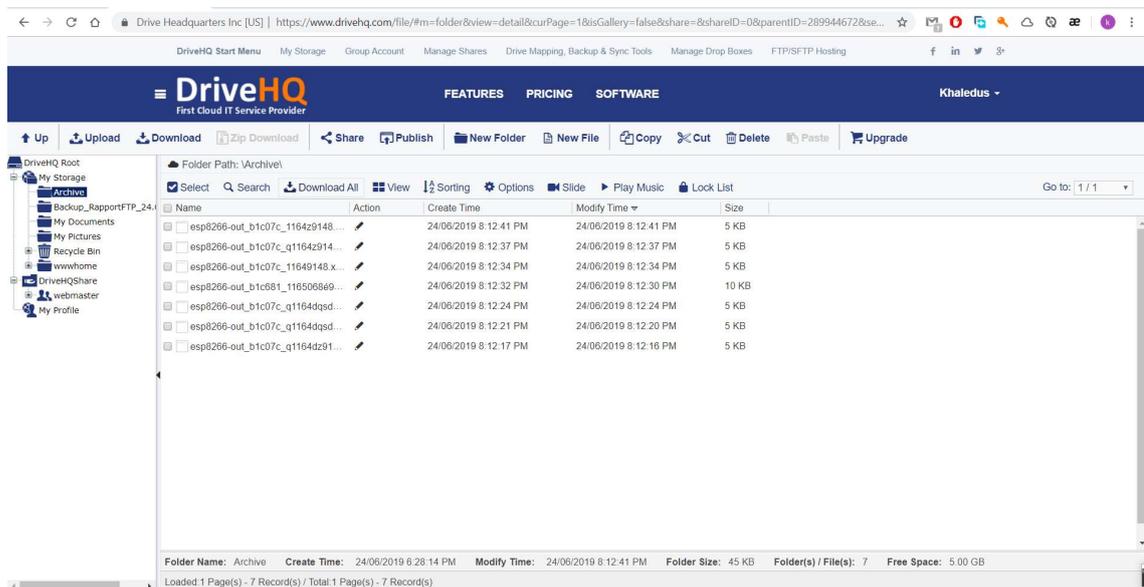


Figure 3.7: Consultation de l'archive depuis la plateforme drivehq.com

3.5 Problèmes rencontrés

Le système réalisé est encore au stade de test, nous n'avons pas pu tester toutes les fonctionnalités et respecter les conditions mentionnées dans le chapitre 1.

Même si le système réalisé est fonctionnel, il n'est pas tout à fait fiable pour une application commerciale. Ce travail reste perfectible et constitue une base solide pour un système plus performant, en effet, il nécessite toujours des améliorations. Nous mentionnons quelques problèmes que nous avons rencontrés et les parties que nous n'avons pas pu réaliser.

3.5.1 Le point d'Access

Dans le réseau maillé local du système, les nœuds doivent se connecter au point d'accès racine généré par l'administrateur. Dans la réalisation il était prévu que le 'Raspberry (l'administrateur) est capable de générer un point d'accès, les nœuds connectent a ce réseau, mais ça n'a pas été achevé, même si nous étions certain que le Raspberry puisse le faire selon nos recherches sur ce sujet [25] [26], la raison reste toujours inconnue, il est probable que cela est dû à de mauvaises manipulations des fichiers système du système linux.

3.5.2 Le serveur FTP

La méthode d'archivage que nous avons mentionné précédemment est basée sur un serveur hébergé sur internet, Nous avons besoin d'utiliser un serveur FTP hébergé et non pas un serveur propriétaire pour les raisons suivantes :

Puisque le Protocol FTP utilise deux connexions TCP[27] :

- Une connexion de contrôle est utilisée pour acheminer les commandes (ou requêtes) du client vers le serveur et les réponses (ou résultats) du serveur vers le client.
- Une connexion de transfert de données qui est créée à chaque fois qu'un fichier est transféré entre le client et le serveur.

Les ports de transfert de données seront ouverts d'une façon aléatoire et même si on peut les connaitre, il n'est pas possible d'établir un Mappage de ports (Port forwarding) pour une plage de ports dans un modem ordinaire pour les connexions ADSL.

D'ailleurs il est tout à fait possible d'effectuer cette connexion en réseau local puisque on n'aura pas besoin de faire le mappage de port dans le réseau local

Voici une capture de la commande « netstat » pour surveiller les adresses et les ports qu'utilise le protocole FTP :

```
ftp> open localhost 2121
Connecté à DESKTOP-FLRLSR1.
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit https://filezilla-project.org/
202 UTF8 mode is always enabled. No need to send this command.
Utilisateur (DESKTOP-FLRLSR1:(none)) : kkk
331 Password required for kkk
Mot de passe :
230 Logged on
ftp> get aaalog
200 Port command successful
150 Opening data channel for file download from server of "/aaalog"
226 Successfully transferred "/aaalog"
ftp : 188 octets reçus en 0.00 secondes à 188000.00 Ko/s.
ftp>
```

Figure 3.8: Utilisation du protocole FTP depuis le CLI dans le réseau local

```
C:\Users\Notebook Khaled>netstat -ano | find "17120"
TCP    0.0.0.0:2121          0.0.0.0:0           LISTENING   17120
TCP    127.0.0.1:14147     0.0.0.0:0           LISTENING   17120
TCP    [::]:2121           [::]:0              LISTENING   17120
TCP    [::1]:2121         [::1]:11150         ESTABLISHED 17120
TCP    [::1]:14147        [::]:0              LISTENING   17120
TCP    [::1]:14147        [::1]:11137         ESTABLISHED 17120
```

Figure 3.9: Visualisation de la commande netstat

Captures pour un deuxième essai :

```
ftp> get aaalog
200 Port command successful
150 Opening data channel for file download from server of "/aaalog"
226 Successfully transferred "/aaalog"
ftp : 188 octets reçus en 0.00 secondes à 188000.00 Ko/s.
ftp>
```

Figure 3.10: Utilisation du protocole FTP depuis le CLI

```
C:\Users\Notebook Khaled>netstat -ano | find "17120"
TCP    0.0.0.0:2121          0.0.0.0:0           LISTENING   17120
TCP    127.0.0.1:14147     0.0.0.0:0           LISTENING   17120
TCP    [::]:2121           [::]:0              LISTENING   17120
TCP    [::1]:2121         [::1]:11180         ESTABLISHED 17120
TCP    [::1]:14147        [::]:0              LISTENING   17120
TCP    [::1]:14147        [::1]:11137         ESTABLISHED 17120
```

Figure 3.11: Deuxième visualisation de la commande netstat

On remarque que dans la deuxième utilisation le protocole a ouvert d'autres ports que la première utilisation pour le transfert des fichiers

Conclusion générale

Dans ce mémoire nous avons essayé de concevoir un système d'internet des objets, générique et multidisciplinaire, le premier volet de l'étude a porté sur une étude théorique qui résume d'autres propositions dans le domaine, plus une proposition d'un concept, concrétisé sur une application des serre agricole.

Il est évident que ce concept n'a pas été complété et qu'il a besoin davantage d'études et d'améliorations. Pour cela, nous considérons ce travail comme un fondement pour d'autres projets futurs.

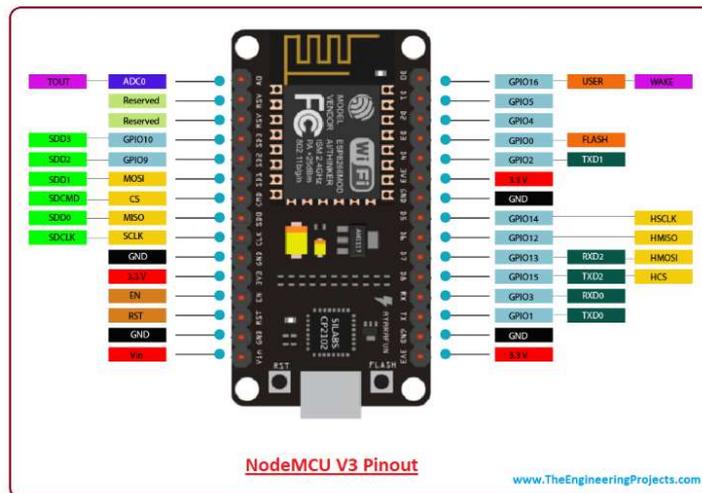
Le but lointain consiste à donner un coup de pouce à la recherche et à la fabrication dans le domaine de l'internet des objets, premièrement au niveau de l'université, puis, à l'échelle nationale.

Ce travail nous a permis d'approfondir et de pratiquer nos connaissances théoriques acquises tout au long de notre formation et d'enrichir l'esprit de la programmation et la conception, ainsi que de nous familiariser avec les nouvelles technologies dans le domaine de réseaux et services pour l'internet des objets.

A notre vue, ce travail a ouvert autant de perspectives que de solutions proposées, et a déclenché de nouvelles problématiques qui ouvrent la voie à d'autres sujets pour d'autres projets de recherches :

- Perfectionnement du système pour être fiable, générique, multidisciplinaire
- Degré de sécurité des protocoles MQTT (test de pénétration)
- Perfectionnement du système en termes d'acquisition et traitement dynamique
- Application de l'approche avec d'autres périphériques bas niveau et bon marché
- Etudier la capacité d'arborescence des nœuds dans la topologie maillée.

Annexe 1 : NodeMCU



Pinout du NodeMCU v3 [28]

NodeMCU V3 fonctionnalités [28]:

- Open-source
- Hardware de type Arduino
- LED de statut
- port MicroUSB
- Interactif et programmable
- Faible coût
- Wi-Fi intégré
- Convertisseur USB / UART
- Pins GPIO

Caractéristiques :

Alimentation	5 Vcc via MicroUSB 5 à 9 Vcc via broche Vin (régulateur intégré)
Microcontrôleur	ESP8266 12E
Microprocesseur	Tensilica LX106
Fréquence	80 à 160 MHz
Mémoire RAM	64 kB
Mémoire Flash	96 96 kB
Entrées sorties	16 GPIO
Sorties PWM	4 sorties
Interfaces	I2C, SPI, UART
Interface Wi-Fi	802.11 b/g/n 2,4 GHz
Antenne Wi-Fi	intégrée
Température de service:	-40 à 125 °C
Dimensions:	58 x 31 x 12 mm

Caractéristiques du NodeMCU v3 [29]

Annexe 2 : Texas Instruments MSP430



LaunchPad with MSP430G2553

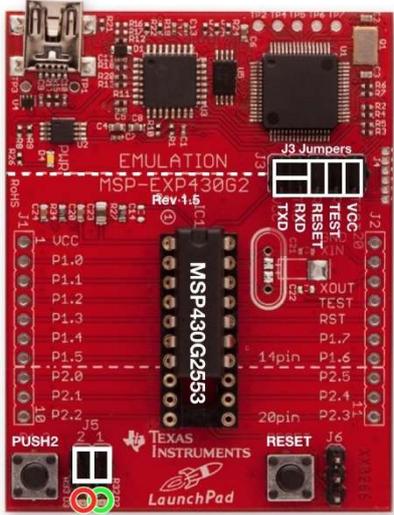
Revision 1.5

Flash	16	KB
RAM	512	B

Serial	Hardware
ADC	10 bits
Use pins numbers only!	
Default I ² C = (1)	
Software I ² C (1) master only	
PWM 4 or 14 or 19	
PWM 9 or 10	
PWM 12 or 13	

+3.3V				1
RED_LED		A0	P1_0	2
	RXD	A1	P1_1	3
	TXD	A2	P1_2	4
PUSH2		A3	P1_3	5
		A4	P1_4	6
	SCK (B0)	A5	P1_5	7
	CS (B0)		P2_0	8
	SCL (1)		P2_1	9
	SDA (1)		P2_2	10

temperature		A10	
-------------	--	-----	--



Hardware	
Pin number	
I ² C	
Serial UART	
SPI	
analogRead()	
digitalRead() and digitalWrite()	
digitalRead(), digitalWrite() and analogWrite()	

20				GROUND
19	P2_6			XIN
18	P2_7			XOUT
17				TEST
16				RESET
15	P1_7	A7	SDA (0)	MOSI (B0)
14	P1_6	A6	SCL (0)	MISO (B0)
13	P2_5			GREEN_LED
12	P2_4			
11	P2_3			

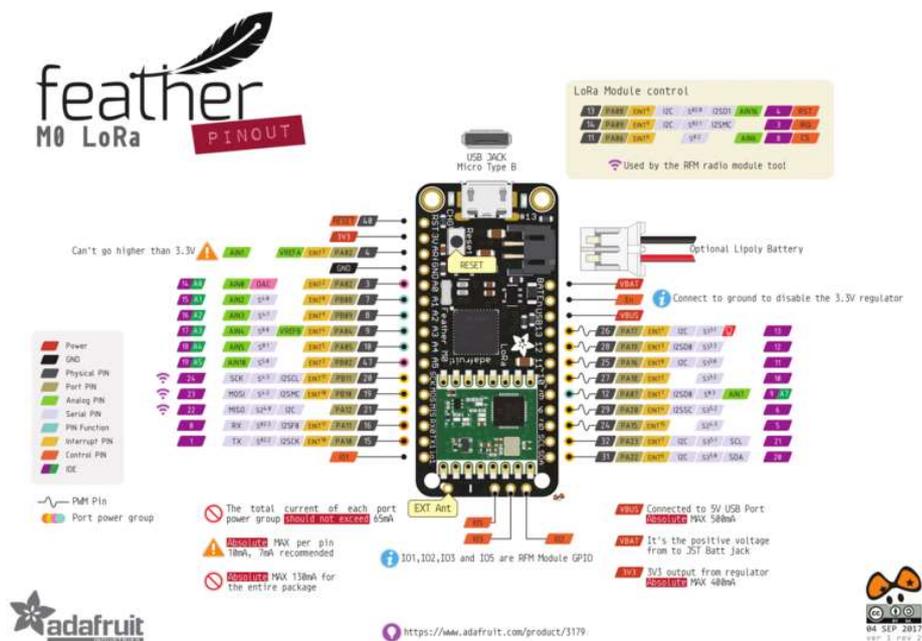
GND
GND
+3.3V

Pinout du MSP430G2553[30]

Feature	MSP430G2553 Controller
CPU architecture	16-Bit RISC Architecture
Frequency of Operation	Up to 16MHz
Flash Memory	16KBytes
RAM	512Bytes
Interface	1x I2C 1x SPI 1x UART
Programmable I/O pins	16
ADC	8 channels of 10 bit resolution
Comparator	8 inputs available
Timer	2 x 16 bit
Capacitive touch interface	Available
Internal temperature sensor	Available
Watchdog timer	Available

Caractéristiques du MSP430G2553 [31]

Annexe 3 : Adafruit Feather M0 WiFi /ATWINC1500 [32]



Pinout du Adafruit Feather M0 WiFi w/ATWINC1500

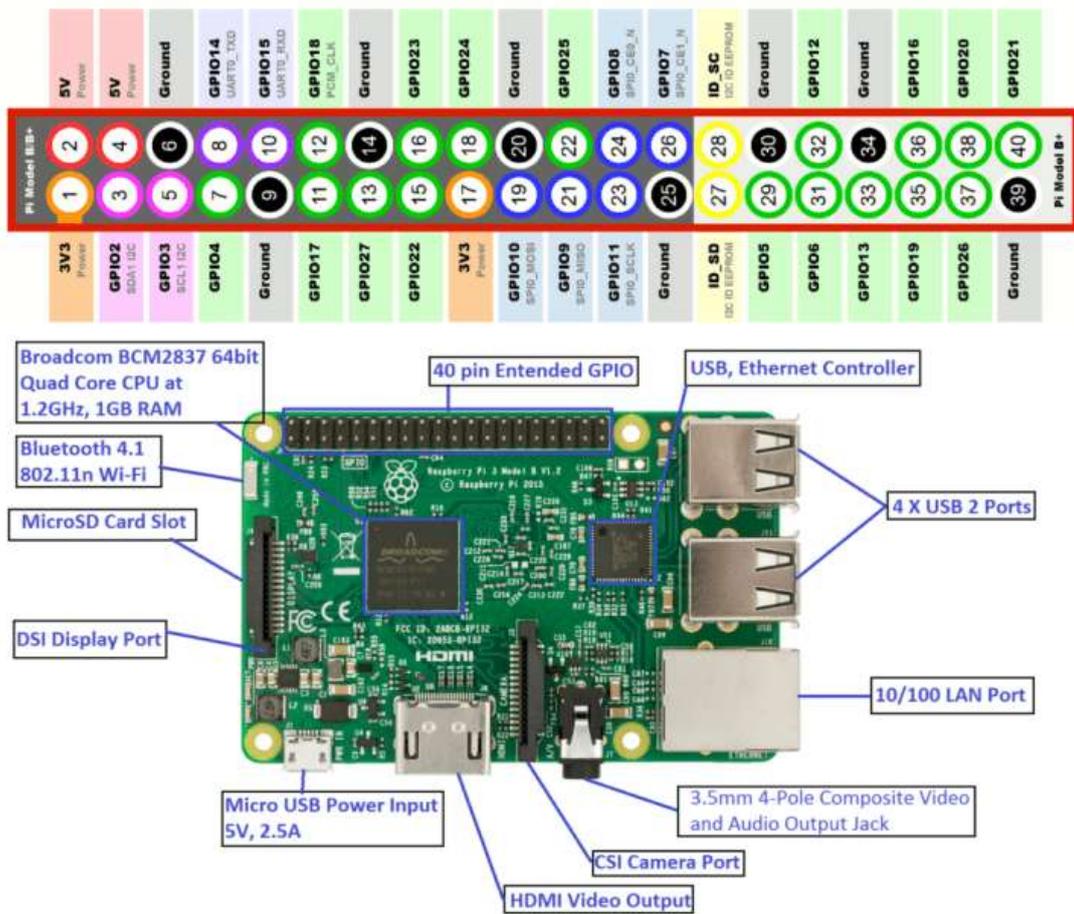
Alimentation	5 Vcc via MicroUSB 3.7 Vcc batteries lithium polymère 3.7V
Module Wifi	ATWINC1500
Microprocesseur	ATSAMD21G18
Fréquence	48MHz
Mémoire RAM	32 kB
Mémoire Flash	256kB
Entrées sorties	10 I/O digitales compatibles PWM
Sortie PWM	8 sorties
Interfaces	I2C, SPI, UART
Interface Wifi	802.11 b/g/n 2,4 GHz
Antenne Wi-Fi	intégrée
Dimensions:	53.65mm x 23mm x 8mm

Caractéristiques du Adafruit Feather MO WiFi w/ATWINC1500

Annexe 4 : Raspberry pi 3

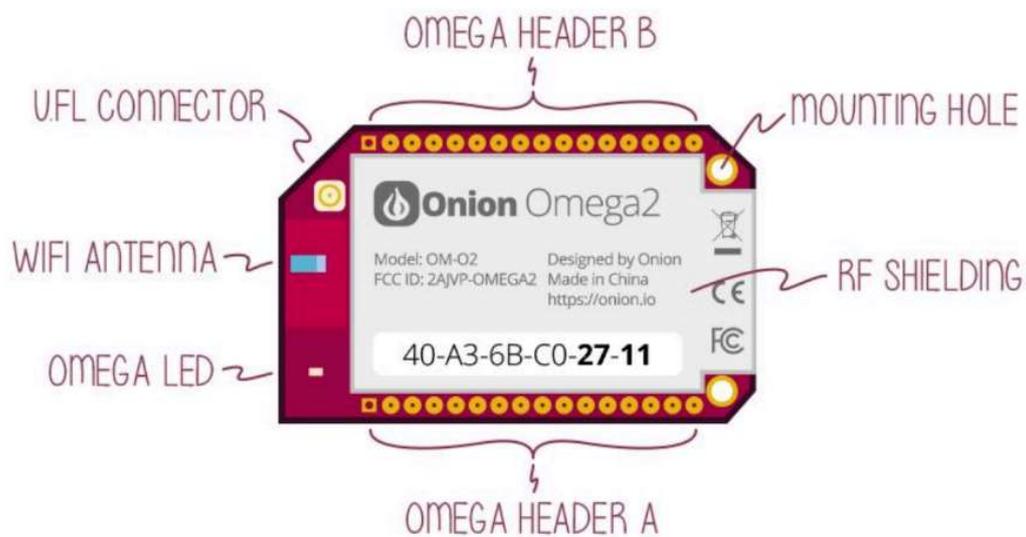
Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50 °C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

Caractéristiques du Raspberry Pi 3 Model B+ [33]



Hardware et Pinout du Raspberry Pi 3 [34]

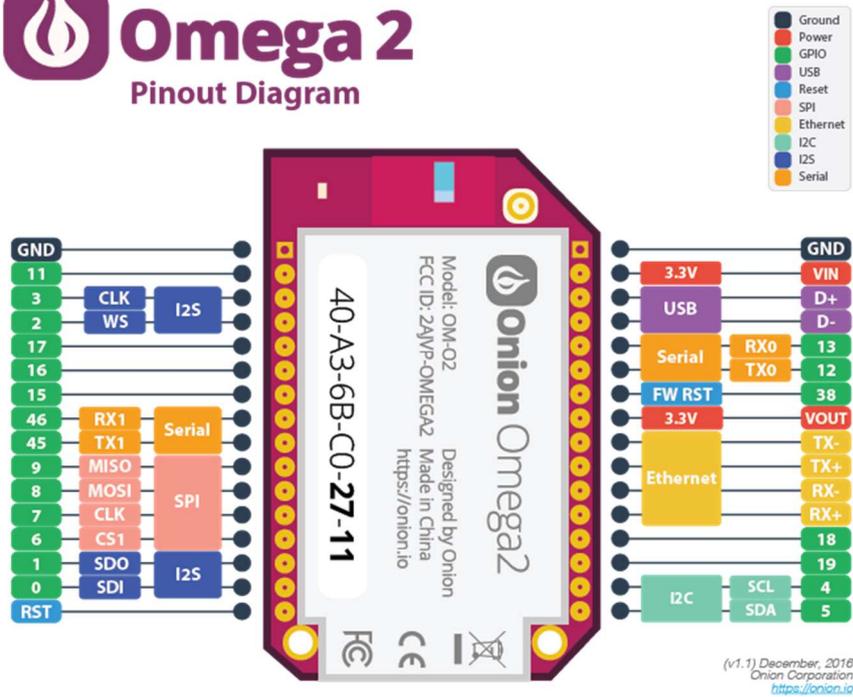
Annexe 5 : Onion Omega2+ [35]:



Hardware Du Omega2+

Omega 2

Pinout Diagram



Pinout Du Omega2+

Processeur	580MHz MIPS CPU
RAM	128MB
Mémoire	32MB
USB	USB 2.0
Wifi	b/g/n Wi-Fi
Alimentation	3.3V
GPIO	15
PWM	2
UART	2
I2C	1
SPI	1
I2S	1

Caractéristiques de l'Omega2+

Annexe 6 : Beaglebone Blue

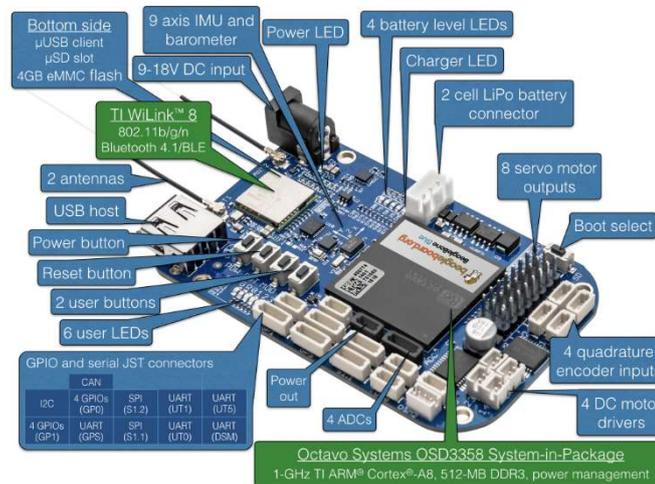
Processor: Octavo Systems OSD3358

- AM335x 1GHz ARM® Cortex-A8 processor
- 512MB DDR3 RAM
- 4GB 8-bit eMMC flash storage
- Integrated power management
- 2×32-bit 200-MHz programmable real-time units (PRUs)
- NEON floating-point accelerator
- ARM Cortex-M3
- USB2 client for power & communications, USB2 host
- Programmed with [Debian Linux](#)

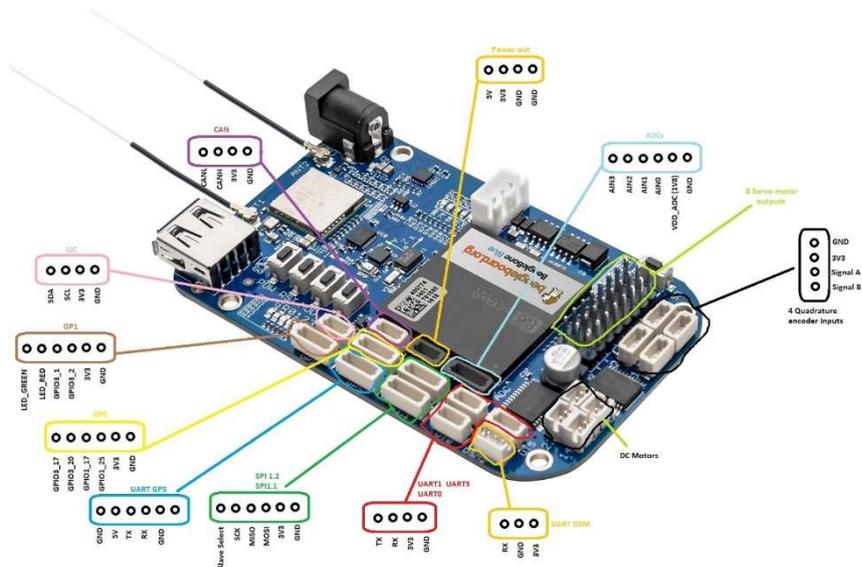
Connectivity and sensors

- Battery support: 2-cell LiPo with balancing, LED state-of-charge monitor
- Charger input: 9-18V
- Wireless: 802.11bgn, Bluetooth 4.1 and BLE
- Motor control: 8 6V servo out, 4 bidirectional DC motor out, 4 quadrature encoder in
- Sensors: 9 axis IMU (accels, gyros, magnetometer), barometer, thermometer
- User interface: 11 user programmable LEDs, 2 user programmable buttons
- Easy connect JST interfaces for adding additional buses and peripherals including: GPS, DSM2 radio, UARTs, SPI, I2C, 1.8V analog, 3.3V GPIOs

Caractéristiques du Beaglebone [36]



Hardware du Beaglebone Blue [37]



Annexe 7 : Exemple d'utilisation du capteur DHT11 avec un NodeMCU

```
1 #include "DHT.h"
2
3 DHT dht;
4 |
5 void setup()
6 {
7     Serial.begin(9600);
8     Serial.println();
9     Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
10
11     dht.setup(2); // data pin 2
12 }
13
14 void loop()
15 {
16     delay(dht.getMinimumSamplingPeriod());
17
18     float humidity = dht.getHumidity();
19     float temperature = dht.getTemperature();
20
21     Serial.print(dht.getStatusString());
22     Serial.print("\t");
23     Serial.print(humidity, 1);
24     Serial.print("\t\t");
25     Serial.print(temperature, 1);
26     Serial.print("\t\t");
27     Serial.println(dht.toFahrenheit(temperature), 1);
28 }
```

Annexe 8 : Exemple d'utilisation du capteur à ultrason avec un NodeMCU

```

1 /* Constantes pour les broches */
2 const byte TRIGGER_PIN = 2; // Broche TRIGGER
3 const byte ECHO_PIN = 3; // Broche ECHO
4 /* Constantes pour le timeout */
5 const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/
6 /* Vitesse du son dans l'air en mm/us */
7 const float SOUND_SPEED = 340.0 / 1000;
8 void setup() {
9   /* Initialise le port série */
10  Serial.begin(115200);
11  /* Initialise les broches */
12  pinMode(TRIGGER_PIN, OUTPUT);
13  digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au repos
14  pinMode(ECHO_PIN, INPUT); }
15 void loop() {
16  /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche TRIGGER */
17  digitalWrite(TRIGGER_PIN, HIGH);
18  delayMicroseconds(10);
19  digitalWrite(TRIGGER_PIN, LOW);
20  /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (si il existe) */
21  long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);
22  /* 3. Calcul la distance à partir du temps mesuré */
23  float distance_mm = measure / 2.0 * SOUND_SPEED;
24  /* Affiche les résultats en mm, cm et m */
25  Serial.print(F("Distance: "));
26  Serial.print(distance_mm);
27  Serial.print(F("mm "));
28  Serial.print(distance_mm / 10.0, 2);
29  Serial.print(F("cm, "));
30  Serial.print(distance_mm / 1000.0, 2);
31  Serial.println(F("m"));
32  /* Délai d'attente pour éviter d'afficher trop de résultats à la seconde */
33  delay(500); }

```

Annexe 9 : Exemple d'utilisation d'une photorésistance avec un NodeMCU

```

1 void setup() {
2
3   Serial.begin(9600);
4 }
5
6 void loop() {
7
8   // Mesure la tension sur la broche A0
9   int sensorValue = analogRead(A0);
10  float lum = sensorValue * (100.0 / 1023.0);
11
12  Serial.println(lum);
13  delay(250);
14 }

```

Références

- [1] Y. Kang, M.-R. Han, K. Han, and J.-B. Kim, "A Study of Internet of Things (IoT) Applications," 2015.
- [2] E. Gasiorowski-Denis, "How the Internet of Things will change our lives," *ISO News*, vol. 5, 2016.
- [3] J. Barbier and D. Handler, "Adoptez l'Internet of Everything et exploitez le potentiel économique de 14, 4 trillions de dollars."
- [4] A. Kulkarni and S. Sathe, "Healthcare applications of the Internet of Things: A Review," *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 6229-6232, 2014.
- [5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, pp. 22-32, 2014.
- [6] D. D. MEGTIT Tedjini, "Réalisation d'une serre agricole intelligente et contrôlable à distance par Internet," Master, génie électrique et électronique, Université Abou Bakr Belkaid–Tlemcen, 2018.
- [7] Smarthome. *Your Trusted Advisor in Home Automation*. Available: <https://www.smarthome.com/>
- [8] LUTRON. *Quantum, a lighting control and energy management system*. Available: <http://www.lutron.com/en-US/Products/Pages/WholeBuildingSystems/Quantum/Overview.aspx>
- [9] Belkin. *Wemo - IoT solution for home automation*. Available: <https://www.belkin.com/us/c/wemo/>
- [10] (14/09/2019). *Goutra - Meilleure solution breveté d'économie de l'eau*. Available: <http://fr.goutra.com/>
- [11] A. Mayard. (July 07, 2016). *Is Algeria ready for Aquasafe's IoT? - Wamda*. Available: <https://www.wamda.com/memakersge/2016/07/is-algeria-ready-for-iquasafe-iot>
- [12] Espressif, *ESP-IDF Programming Guide*, 21/08/2019.
- [13] B. Aziz, "On the security of the MQTT protocol," in *Engineering secure Internet of Things systems*, ed: Institution of Engineering and Technology, 2016, pp. 159-178.
- [14] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, 2008, pp. 791-798.
- [15] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, "A Review of Low-End, Middle-End, and High-End IoT Devices," *IEEE Access*, vol. 6, pp. 70528-70554, 2018.
- [16] *Raspberry Pi Foundation - About Us*. Available: <https://www.raspberrypi.org/about/>
- [17] (02/09/2019). *SSH Protocol – Secure Remote Login and File Transfer | SSH.COM*. Available: <https://www.ssh.com/ssh/protocol/>
- [18] (26/08/2019). *Open source technology -What is Ngrok?* Available: <https://vmokshagroup.com/blog/expose-your-localhost-to-web-in-50-seconds-using-ngrok/>
- [19] Arduino.cc. (29/08/2019). *Arduino - FAQ* Available: <https://www.arduino.cc/en/Main/FAQ>
- [20] (15/04/2019). *ESP8266MQTTMesh*. Available: <https://github.com/PhracturedBlue/ESP8266MQTTMesh>
- [21] N. M. Rafik LEMDANI, "Etude, conception et réalisation d'une plateforme pour l'automatisation et le contrôle à distance des serres agricoles.," Mémoire de Master,

Faculté des Sciences de l'Ingénieur, UNIVERSITE M'HAMED BOUGARA-BOUMERDES, 2017.

- [22] M. Ruys. *Arduino-DHT-Master*. Available: <https://github.com/markruys/arduino-DHT>
- [23] CFAURY. (15/01/2016). *Capteur de distance à Ultrasons – Arduino : l'essentiel*. Available: <http://arduino.blaisepascal.fr/capteur-de-distance-a-ultrasons/>
- [24] J.-n. Rousseau. *Les capteurs électroniques*. Available: <https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino/3285400-les-capteurs-electroniques>
- [25] (05/09/2019). *Setting up a Raspberry Pi as a Wireless Access Point - Raspberry Pi Documentation*. Available: <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>
- [26] (05/09/2019). *How to Create WiFi Hotspot in Ubuntu 16.04 (Android is Supported) | UbuntuHandbook*. Available: <http://ubuntuhandbook.org/index.php/2016/04/create-wifi-hotspot-ubuntu-16-04-android-supported/>
- [27] B. S. Berrabah Rahma, *Administration des services de partage et de transfert des fichiers (samba, nfs et ftp) par l'outil webmin*, 15-oct-2014.
- [28] A. Aqeel. (2018, 10/08/2019). *Introduction to NodeMCU V3 - The Engineering Projects*. Available: <https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html>
- [29] (10/08/2019). *Module NodeMCU ESP8266*. Available: <https://www.gotronic.fr/art-module-nodemcu-esp8266-27744.htm>
- [30] (10/08/2019). *Guide to the MSP430G2 LaunchPad (MSP-EXP430G2)*. Available: <https://energia.nu/pinmaps/msp-exp430g2/>
- [31] (8/12/2018 24/08/2019). *MSP430 Launchpad Pinout, Features, Programming & Datasheet*. Available: <https://components101.com/microcontrollers/msp430-launchpad-pinout-datasheet>
- [32] l. ada. (Jan 26, 2016, 12/08/2019). *Adafruit Feather M0 WiFi with ATWINC1500*. Available: <https://learn.adafruit.com/adafruit-feather-m0-wifi-atwinc1500/>
- [33] (24/08/2019). *Raspberry Pi 3 Model B+ product brief*. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>
- [34] M. Aqib. (2019, 23/08/2019). *What is Raspberry pi? Getting Started with Raspberry Pi 3*. Available: <https://electronics hobbyists.com/tutorial-1-what-is-raspberry-pi-getting-started-with-raspberry-pi-3/>
- [35] *Omega2 Project Book* vol. 1: Onion.
- [36] (24/08/2019). *What is BeagleBone® Blue?* Available: <https://beagleboard.org/blue>
- [37] (24/08/2019). *BeagleBone Blue Github page - Pinouts*. Available: <https://github.com/beagleboard/beaglebone-blue/wiki/Pinouts>