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 Saâd Dahlab de Blida

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de projet de fin d'études
pour l'obtention du diplôme de Master 2 en Systèmes de Télécommunications

---

Sous thème :

Réseaux de capteurs sans fils appliqués dans l'amélioration de la productivité agricole

---

Réalisé par : **Khalida DJAOUI** et **Nassila BENALI**
Sous la direction de **M. A. BENBELKACEM**

Soutenu publiquement le 25/09/2022 auprès des membres du jury :

Président	**M.**	**Université Blida 1**
Promoteur	**M. A. BENBELKACEM**	**ENS Kouba**
Co-promoteur	**M. M. MAAMOUN**	**Université Blida 1**
Examinateur	**M. S. DAHMANI**	**Université Blida 1**
Examinatrice	**M^{me}**	**Université Blida 1**

Promotion 2021/2022

Remerciements

Nous tenons à remercier nos familles, nos fidèles amis et nos professeurs pour les valeurs qu'ils nous ont apporté au long de notre cursus.

RÉSUMÉ

ملخص : نتطرق في أطروحتنا إلى دراسة شبكات الاستشعار اللاسلكية بهدف تحسين الإنتاجية الزراعية في سياق مراقبة تغيرات درجات الحرارة والرطوبة داخل دفيئات زراعية. يدور الحديث في الجزء الأول حول ثورة الزراعة الذكية. أما الجزء الثاني، فعلى تفاصيل الاتصال داخل شبكات الاستشعار اللاسلكية و سنقدم في الجزء الأخير، محاكاة لشبكة الاستشعار في مزرعة ذات دفيئات مختلفة ، باستخدام أدوات MQTT و Python و Docker و InfluxDB.

RÉSUMÉ : Cette thèse consiste en l'étude des réseaux de capteurs sans fils avec comme objectif l'amélioration de la productivité agricole, dans le cadre du suivi des changements de températures et d'humidité dans des serres agricoles. Nous entamerons dans un premier chapitre les rouages de l'agriculture intelligente; dans un deuxième, la connectivité des réseaux de capteurs sans fils et dans un troisième, une simulation d'un réseau de capteurs dans une ferme à serres, utilisant les outils MQTT, Python, Docker et InfluxDB.

ABSTRACT : This thesis consists of the study of wireless sensor networks for improving agricultural production, in the context of monitoring changes in temperature and humidity in greenhouses. We will talk, in the first chapter, about smart agriculture; in the second, about the connectivity of wireless sensor networks and in the third, about simulating a sensor network in a farm, using the tools : MQTT, Python, Docker and InfluxDB.

Table des matières

Liste des acronymes et abréviations

AES	**Advanced Encryption Standard**
AF	**Sous-couche**
AODV	**Ad hoc On Demand Distance Vector**
APL	**Couche Application**
APS	**Sous-couche**
ASK	**Amplitude Shift Keying**
BSK	**Binary Shift Keying**
CSMA/CA	**Carrier Sense Multiple Access with Collision Avoidance**
IEEE	**Institute of Electrical and Electronics Engineers**
ISO	**Organisation Internationale de Normalisation**
MAC	**Couche MAC**
MQTT	**Message Queuing Telemetry Transport**
NWK	**Couche Network (Couche Réseau)**
O-QPSK	**Offset Quadrature Phase Shift Keying**
OSI	**Open System Interconnection**
PHY	**Couche physique**
RCSF	**Réseaux de Capteurs Sans Fils**
YAML	**Yet Another Markup Language**
ZDO	**Sous-couche**

LISTE DES FIGURES

Liste des tableaux

Introduction générale

L'agriculture est actuellement confrontée à deux défis majeurs : assurer la sécurité alimentaire d'une population en pleine croissance tout en prenant en compte l'évolution des modes de consommation, en plus des changements climatiques tels que l'augmentation des températures moyennes, les modifications des régimes de précipitation, etc. C'est là qu'est introduit l'Agriculture Intelligente : une approche qui aide les acteurs du secteur agricole à relever ses défis grâce à différentes technologies dont on nomme -parmi tant d'autres- : Les réseaux de capteurs sans fils.

Ce mémoire, réalisé dans le cadre de notre projet de fin d'études en Master de Systèmes de Télécommunications, est structuré en trois chapitres :

Dans le chapitre 1, nous abordons l'agriculture intelligente qui avait introduit l'Internet des Objets à l'agriculture de précision et qui a permis l'automatisation des systèmes agricoles et, par conséquent, la progression de la qualité de la productivité. Les serres intelligentes, leur systèmes d'automatisation et de gestion d'énergie en font l'objet principal.

Le chapitre 2 discute les principes de fonctionnement des réseaux de capteurs sans fils, leurs architectures internes, leurs protocoles de connectivité de même que leurs protocoles de gestion d'énergie.

Quant au chapitre 3, il est consacré à une simulation d'un réseau de capteurs sans fils visant à analyser les paramètres environnementaux : température et humidité, dans une ferme à serres; grâce au protocole de communication sans fil MQTT, l'outil Python pour la génération de données factices, Docker pour leur conteneurisation à travers le serveur Cloud et la plateforme de catégorisation et d'affichage de données InfluxDB.

Chapitre 1

Les Serres et l'agriculture intelligente

1.1. Introduction : Dans ce chapitre, nous allons présenter l'agriculture intelligente et aussi parler du rôle des serres intelligentes dans l'amélioration de la production agricole ainsi que de leur capacité de contrôler quelques paramètres environnementaux tels que la température, l'humidité, la lumière, etc.

1.2. L'agriculture intelligente : L'agriculture de précision fait référence à une série de stratégies et d'outils qui permettent d'optimiser et d'augmenter la qualité et la productivité des sols par une série d'interventions ciblées, un résultat qui peut être obtenu grâce à des technologies toujours plus avancées. Elle est dite « de précision » parce que, grâce aux outils les plus modernes, il est possible d'effectuer la bonne intervention, au bon endroit, au bon moment, en répondant aux besoins spécifiques de chaque culture et de chaque zone de champ, avec un haut niveau de précision. L'utilisation des technologies dans le domaine d'agriculture Les technologies sont utilisées tout d'abord pour collecter des données et des informations qui servent à prendre des décisions sur la façon d'améliorer la production et ensuite pour mettre en œuvre les corrections nécessaires pour atteindre cet objectif. Aujourd'hui, on parle aussi de plus en plus d'Agriculture 4.0, qui est l'évolution du concept d'agriculture de précision : cette expression désigne tous les outils et stratégies qui utilisent les technologies de pointe de manière interconnectée, à commencer par l'utilisation des données pour améliorer et optimiser la productions.

1.3. Serres intelligentes :

1.3.1. Définition d'une serre : Une serre est une structure close ou semi-ouvert translucide ,elle peut être en verre ou en plastique ,soutenue par une structure métallique ou en bois ,a pour but d'offrir un climat idéal pour l'amélioration de la production des plantes en qualité et en quantité ,elle accélère la croissance et pour les produire indépendamment des saisons Dans une serre agricole on peut contrôler la croissance des plantes grâce à des capteurs installé un peu partout comme le

pluviomètre et des capteurs de température et d'humidité qui nous permettent de surveiller le climat et de le rendre favorable au développement des plantes, on peut l'applet une serre connectée ou intelligente.

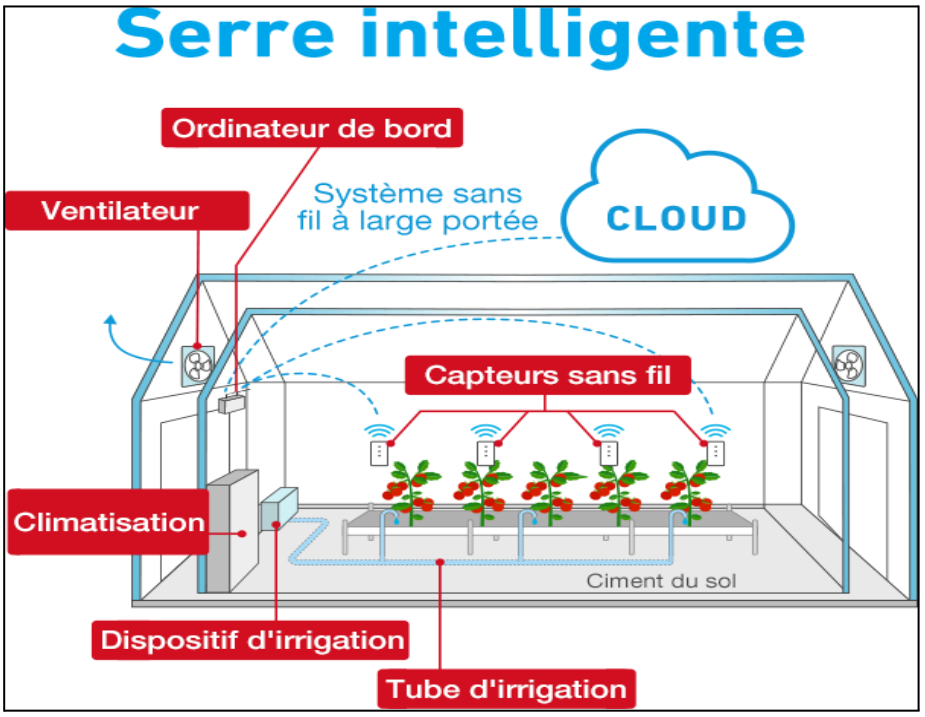


Figure 1.3. Serre intelligente

1.3.2. Types de serres :

- **Serre froide :** désigne une serre non isolée et/ou sans chauffage. Elle permet de protéger les fruits et légumes des conditions météorologiques, ainsi que les végétaux sensibles au froid et au gel. La chaleur et la lumière émises naturellement par le soleil stimulent la croissance des végétaux et participent à leur bon développement. La serre froide est généralement une serre tunnel en plastique La température idéale se situe entre 10 à 15°c. Cependant, elle peut descendre jusqu'à 0°C puisqu'elle n'est pas chauffée. La température peut varier en fonction de la région et de la saison.

- **Serre tempérée :** Sa température ne doit pas descendre au-dessous de +5° C en hiver. Mais elle ne doit pas dépasser 10-12° C non plus pour que les plantes hivernées et qui sont au repos, ne subissent pas de trop fortes variations de température. On aura donc recours, dans la plupart de nos

régions, à un chauffage d'appoint (électrique, à gaz ou à pétrole) pour compenser en hiver les températures inférieures à 0° C.

- **Serre tiède :** Le température minimale doit être de + 15° C. On devra la maintenir assez régulière, ce qui n'est pas très difficile en hiver. Dans certaines régions aux hivers froids, cela nécessite un chauffage efficace fonctionnant pendant plusieurs mois. On peut y cultiver la plupart des plantes exotiques (fleurs et fruits) exigeant un relatif repos en hiver.

- **Serre tropicale :** La Serre Tropicale a été conçue spécialement pour des climats chauds et humides. Elle est constituée de nefs de 12,8 mètres de large, équipées d'un système d'aération zénithale fixe de grande envergure et la hauteur au faîtage est de 6,9 mètres. Cette structure implique un grand volume d'air à l'intérieur de la serre et permet de lutter contre les hausses de température. La grande surface d'aération permet un taux important du renouvellement de l'air. Ce modèle assure une luminosité maximale à l'intérieur de la serre. Sa température ne doit pas descendre au-dessous de + 18° C. On devra l'équiper d'un solide chauffage . En été, il faudra veiller à ce que les températures ne dépassent pas + 30° C. Il faudra une bonne hygrométrie (supérieure à 90° si possible), une ventilation régulière et un ombrage aux heures chaudes. Grâce à la technologie récente on peut faire face aux problématiques d'agriculture en installant un système de ventilation qui permet de maîtriser la variations de la température dans les serres Ainsi il est conseillé d'utiliser des déshumidificateurs de serres thermodynamiques qui transforment l'air humide en chaleur.

1.4. **Gestion de température et humidité dans les serres :** Les entrepreneurs du secteur de l'horticulture sous serre cherchent non seulement à réduire leurs émissions de CO2, mais aussi à optimiser leur productivité. Un contrôle climatique actif a un rôle important à jouer dans ces domaines. Le contrôle climatique à l'intérieur de la serre peut représenter un véritable challenge à cause des variations de températures et d'humidité. Les causes externes dépendent du climat, de la région et de la saison. Les causes internes incluent la culture, la surface, la hauteur de la serre et la présence ou l'absence d'un écran. A. Climat idéal dans la serre Une température et une humidité idéales

permettent à la plante dans la serre de développer tout son potentiel, sans être gênée par les ravageurs et les champignons. Autrement dit, un climat optimal dans la serre est le gage d'une récolte plus abondante et de meilleure qualité, ayant un effet positif sur le résultat d'exploitation. B. Température Si la température est trop élevée, la plante va « respirer » plus rapidement et brûler du glucose (des sucres) qui ne sera donc plus disponible pour la croissance. Une température trop basse peut toutefois avoir un effet négatif sur la photosynthèse et donc sur le développement de la culture. C. Humidité de l'air Si l'air n'est pas assez humide, les nutriments risquent de brûler et la plante va croître moins vite. Si l'air est trop humide, les plantes auront du mal à absorber les nutriments et l'eau ; elles vont devenir plus sensibles aux champignons et aux ravageurs. Pour atteindre le niveau de transpiration idéal pour la plante, l'air doit être à la bonne température, présenter le taux d'humidité adéquat et la vitesse idéale.

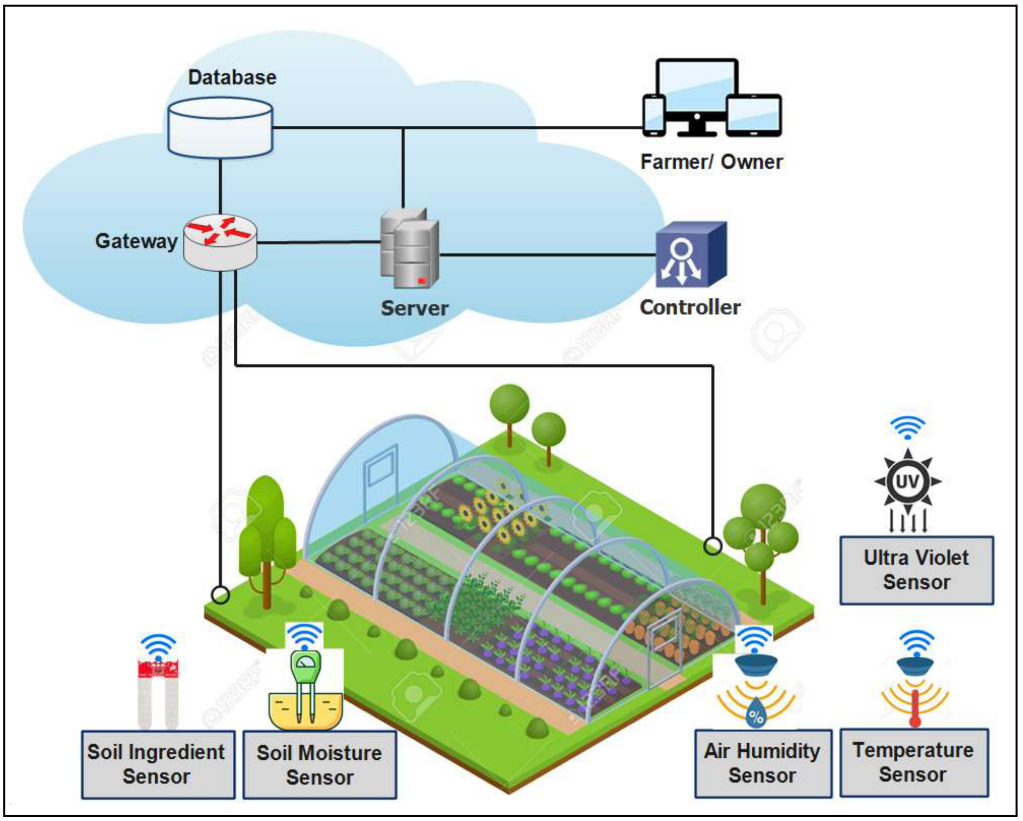


Figure 1.4. Gestion de temperature et d'humidite dans une serre

1.5. **Conclusion :** L'agriculture intelligente est l'approche ultime pour assurer la sécurité alimentaire d'une population en pleine croissance tout en prenant en compte l'évolution des modes de consommation, ainsi que pour faire face aux changements climatiques tels que l'augmentation des températures moyennes, la modification des régimes de précipitation, etc.

CHAPITRE 2

RÉSEAUX DE CAPTEURS SANS FILS

2.1. Introduction : Ce chapitre définit les capteurs, les réseaux de capteurs sans fils, leurs architectures, ainsi que les paramètres de connectivité en leur sein.

2.2. Définition d'un capteur : Un capteur est un dispositif électronique qui permet de recevoir des données de certaines grandeurs physiques (température, humidité, son, mouvement, etc.) à partir de l'entourage de son emplacement, de les convertir en données de différente grandeur physique (souvent électrique) et de les envoyer enfin vers une station de contrôle pour leurs traitement et stockage à des fins de mesure et/ou de commande.

2.3. Les réseaux de capteurs sans fil : Un réseau de capteurs sans fils (RCSF) ou wireless sensors network (WSN) en anglais, est un type particulier des réseaux "Ad Hoc" dont les capteurs peuvent être dispersées dans une zone précise et communiquer entre eux d'une manière autonome et dynamique sans le besoin d'une infrastructure particulière ou de routage central. Le choix de routage est basé sur les algorithmes de connectivité établis par les fabricants et parfois aussi par les administrateurs du réseau, selon les exigences des utilisateurs.

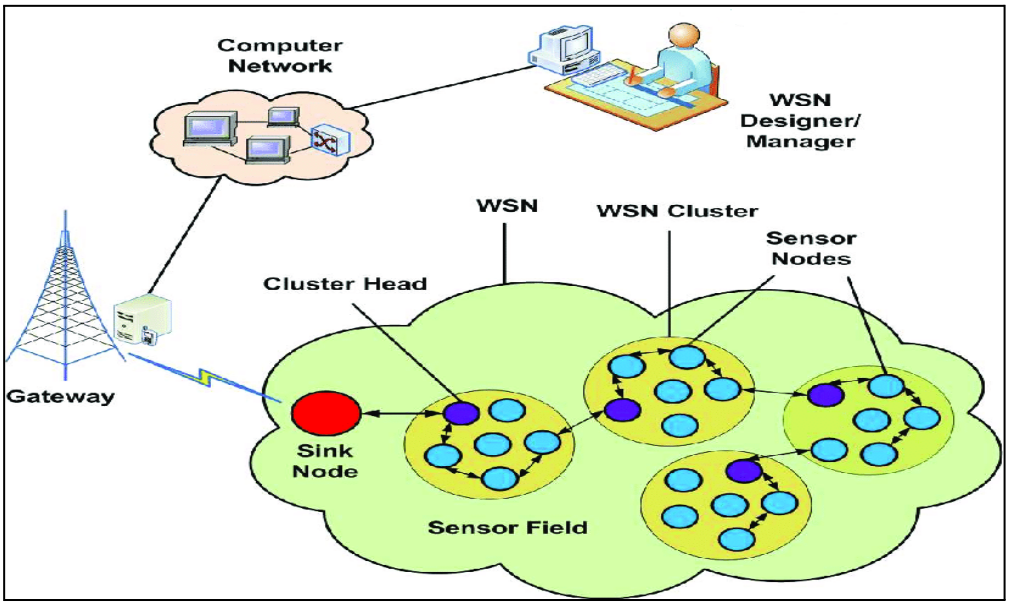


Figure 1.1. Réseau de capteurs sans fils

2.4. Architecture d'un réseau de capteurs sans fils :

2.4.1. Anatomie des RCSF : Le voyage des données dans un RCSF commence quand un capteur, autrement appelé **nœud-capteur**, collecte des données depuis sa zone de détection. Ces données sont ensuite communiquées aux nœuds-capteurs voisins via différents routages possibles jusqu'à ce qu'elles arrivent aux **nœuds-puits** ou **"Sink"**, des points de collecte de données. Ces derniers les transmettent par la suite à une **passerelle** qui permet l'interconnexion du réseau des nœuds-capteurs et nœuds-puits à **un autre réseau (internet, satellite, serveur cloud, etc.)**, souvent loin de la zone de captage et de caractéristiques matérielles et protocoles de transmissions différents, auquel est connectée la **station de contrôle** qui permet aux **utilisateurs** une interaction directe avec les données.

2.4.2. Anatomie d'un nœud : La caractéristique capitale des RCSF réside dans les nœuds. Ce sont des dispositifs électroniques contenant, selon les besoins, différentes unités fondamentales (unité d'acquisition, unité de traitement, unité de stockage et unité de communication et unité d'énergie) et d'autres supplémentaires (ex. unité de commande).

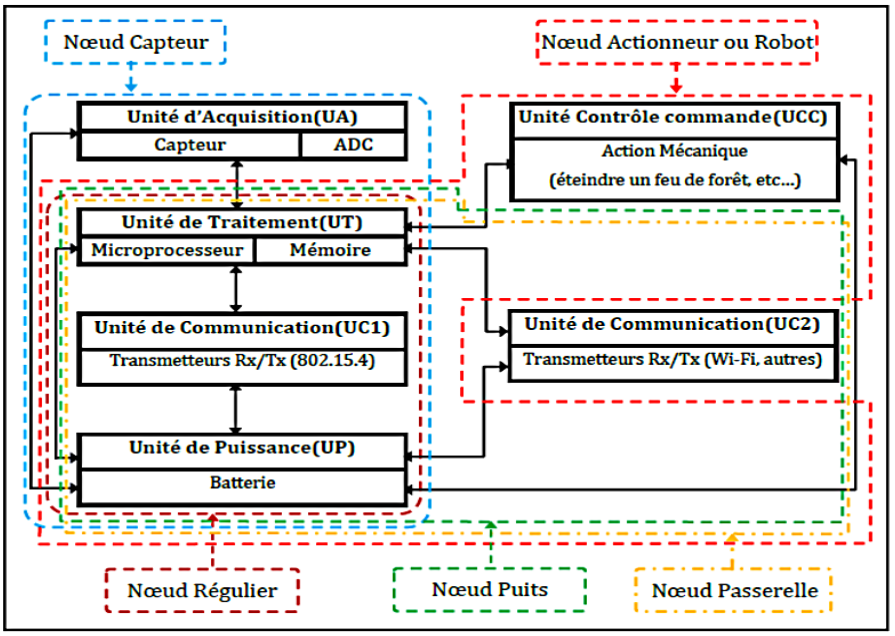


Figure 1.2 : Composants de différents types de nœuds

La différence majeure entre les nœuds-capteurs et les nœuds-puits est la présence d'une **double unité de communication** dans ces derniers, et de **plus grandes capacités de traitement et de stockage**, vu leur rôle plus complexe.

- **Unité de captage :** C'est l'unité qui est chargée de capter des grandeurs physiques (Chaleur, humidité, vibrations, rayonnement, etc..) et de les transformer en grandeurs numériques (un signal électrique). Cette unité peut incorporer de un jusqu' à plusieurs capteurs plus une unité ADC (Analog to Digital Converters). Le rôle de cette dernière consiste à convertir le signal analogique produit par les capteurs, qui est basé sur les données échantillonnées, en un signal numérique.
- **Unité de traitement :** Unité de traitement peut être considérée comme l'organe intelligent du capteur. Elle inclut un processeur qui est généralement associé à une petite unité de stockage. Elle gère des programmes et des logiciels, stockés en mémoire les paramètres métrologiques et fonctionnels (dont la datation est permise par l'horloge interne) ; elle assure les traitements des données reçues de l'unité de captage. Généralement l'unité de traitement commande les autres unités. Les processeurs utilisés dans les réseaux de capteurs sont à faible consommation d'énergie et à faible fréquence. Moins de 10 MHz pour une consommation de 1 mW. Aussi la mémoire de stockage est très limitée, elle est de l'ordre de 10 Ko de RAM pour les données et 10 Ko de ROM pour les programmes. Cette mémoire consomme la majeure partie de l'énergie de l'unité du traitement. Dans la plupart des cas on lui adjoint une mémoire flash moins coûteuse en termes d'énergie compréhensible par l'unité de traitement.

- **Unité de communication :**
  Cette unité assure la connexion entre les nœuds du réseau. Un module radio (émetteur/récepteur) est intégré à cette unité qui permet la communication entre différents nœuds du réseau. La communication peut être de type optique ou radio fréquence. Elle est responsable de la transmission-réception des données captées et traitées via un canal de communication sans fil. Le module radio c'est le module qui consomme le plus d'énergie.

- **Unité d'énergie :** Pour les réseaux de capteurs, l'unité d'énergie est la composante la plus importante, qui représente généralement une batterie. Cette batterie est de petite taille et de capacité énergétique limitée. Souvent, les capteurs sont placés dans des environnements hostiles,

inaccessibles par l'être humain et elle n'est généralement pas remplaçable. Dans ce genre de situation, il est pratiquement impossible de recharger ou de remplacer la batterie. Pour cela, l'énergie représente la contrainte principale lors de la conception d'un réseau de capteurs sans fil puisqu' elle influe sur la durée de vie du nœud capteur et donc la durée de vie du réseau. Cependant, il est possible d'utiliser des systèmes de rechargement d'énergie à partir de l'environnement via des cellules photovoltaïques, par exemple, pour étendre la durée de vie de la batterie. En fonction des besoins de l'application du réseau de capteurs le nœud capteur peut intégrer d'autres unités tels que : - Système de localisation : pour pouvoir déterminer la position du nœud. - Mobilisateur : permet le changement de position du nœud.

2.5. Connectivité dans les réseaux de capteurs sans fils :

2.5.1. Pile protocolaire : Les nœuds capteurs sont généralement dispersés sur un champ de surveillance d'une manière arbitraire, chacun de ces nœuds a la capacité de collecter les données, les router vers le nœud puits (sink), et par la suite vers l'utilisateur finale via une communication multi-sauts. Le nœud puits peut communiquer avec le nœud coordinateur de tâches (utilisateur) par Internet. Chapitre II : Présentation des réseaux de capteurs 19 La pile protocolaire utilisée par le nœud puits ainsi que tous les autres capteurs du réseau est illustrée par la **figure 2.1**. Cette pile prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs . Elle est composée de la couche application, transport, réseau, liaison de données, physique, ainsi que de trois niveaux qui sont : le niveau de gestion d'énergie, de gestion de tâches et le niveau de gestion de mobilité.

- **Couche physique :** contrôle de puissance , codage ,filtrage ,circuit RF, modulation et la démodulation.

- **Couche liaison de donne :** contrôle de puissance, retransmission, méthode d'accès au canal.

- **Couche réseau :** allocation de ressource , routage, découverte du voisin.

- **Couche transport :** retransmission , contrôle du flux.
- **Couche application :** collecte , compression, agrégation, codage.

- **Plan de gestion d'énergie** Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs, dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de captage.

- **plan de gestion de mobilité** Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs, d'une manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les nœuds voisins, cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie.

- **Niveau de gestion des tâches** Lors d'une opération de captage dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme, cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau.

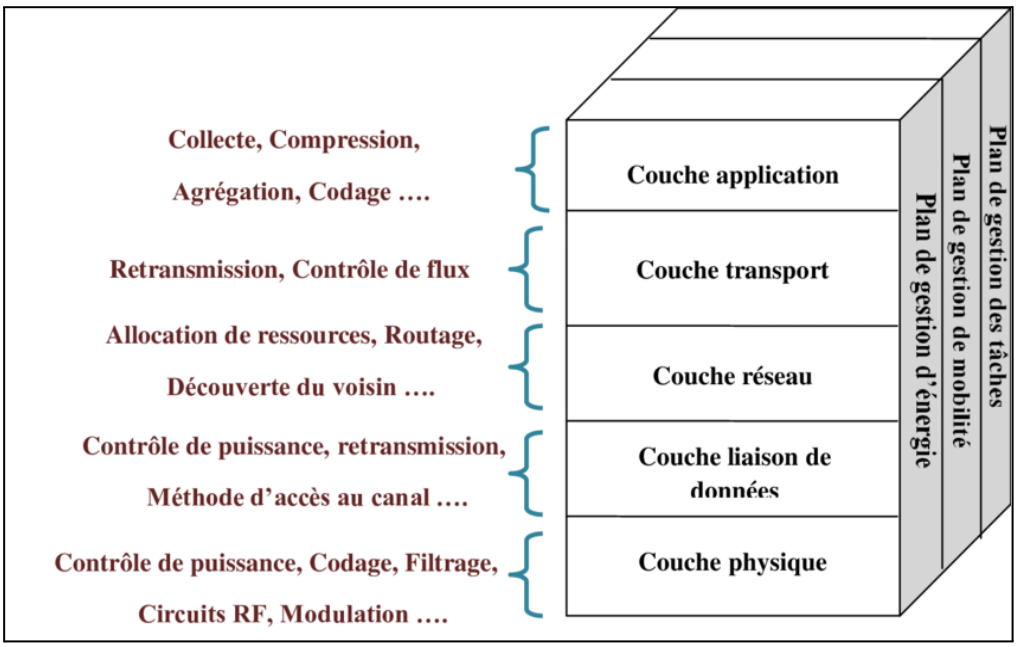


Figure 2.1. Pile protocolaire

2.5.2. Topologies : Le déploiement des capteurs peut se faire de façon prédéfinie ou, plus souvent, de façon complètement aléatoire. Généralement, les capteurs ne connaissent pas la topologie et donc la position de leurs voisins, ce qui affecte directement les performances du routage. Les protocoles de routage doivent donc supporter des méthodes de découverte de voisinage et de son entretien afin de fournir à chaque nœud une connaissance de la topologie qui l'entoure; surtout s'il existe des nœuds mobiles, ce qui est le cas pour certains réseaux sans fil.

2.5.2.1. Topologie Hiérarchique : Pour appliquer cette topologie il doit y avoir au minimum trois niveaux ou groupe de nœuds ,le nœuds central le sink il appartient au niveaux supérieur et il est reliée à un ou plusieurs nœuds qui appartiennent au deuxième niveaux jouent le rôle des passerelles entre ceux de troisième niveau et le le premier niveau de la hiérarchie par une liaison point à point , chaque nœud de deuxième niveaux est relié à un ou plusieurs nœuds plus bas troisième niveau avec une liaison point à point chaque nœud a une responsabilité particulier, le cluster est un nœuds qui est responsable de capte les données et les transmettre au nœud chef (cluster head) ce dernier il est responsable de la transmissions de l'information ver la stations de base un nœud ayant une petite puissance énergétique peut être sélectionné comme un nœuds qui a pour tâche le captage de donne et renvoyer les donne senti au nœuds chef (cluster head) , alors qu'un nœud ayant une énergie plus élevée puisse être choisi comme nœud chef (cluster head) pour traiter les données et les transmettre à la station de base.

2.5.2.2. Topologie plate : Dans cette topologie tous les nœuds sont égaux et on le même rôle et peuvent communiquer entre eux sans devoir passer par une passerelle ou un nœuds particulier ,cette topologie est applicable dans les petit réseaux Le sink est chargé de la collecte des données issue de différents nœuds de capteurs afin de les envoyées vers les centres de traitements En cas d'une panne d'un nœud les donnée seront transmise en utilisant les sauts multiple à travers les nœuds intermédiaires.

2.5.2.3. Topologie point par point La topologie point à point a aussi un coordinateur de pas Dans ce cas (dit « communication multi-sauts »), tout nœud peut échanger avec n'importe quel autre nœud du réseau (doivent être à la portée les un les autres). Un nœud voulant transmettre un message à un autre nœud hors de sa portée de transmission, peut utiliser un nœud intermédiaire pour envoyer son message au nœud destinataire. Cette topologie permet de formation les réseaux plus complexes à mettre en

œuvre telle que la topologie maillé La topologie maillée permet le routage des paquets et les saut multiple (ses fonctions peuvent être ajoutés à la couche réseau grâce à l'alliance de zigbee Avantage : Possibilité de passer à l'échelle du réseau, avec redondance et tolérance aux fautes. Inconvénient : une consommation d'énergie plus importante est induite par la communication multi-sauts. Une latence est créée par le passage des messages des nœuds par plusieurs autres avant d'arriver à la station de base.

2.5.2.4. Topologie en Maille : Une topologie avec une connectivité complète entre les nœuds est une topologie maillée (Mesh en anglais). Dans ce cas (dit « communication multi-sauts ») La topologie maillé transmis l'information en suivant divers chemins à travers plusieurs sauts tous les nœuds sont connecté entre eux Ce type de topologie nécessite un protocole de routage efficace Les nœuds de routeur ou de passerelles dans le réseau maillé reste réveillé de manière permanente pour renvoyer l'information au nœuds d'extrémité ce qui consomme beaucoup d'énergie L'avantage d'utiliser la topologie en maille est la possibilité de passer à l`échelle du réseau, avec redondance et tolérance aux fautes et une bonne couverture. Par contre, les inconvénients d`une telle topologie sont l`importante consommation d`énergie induite par la communication multi-sauts ainsi que la latence créée par le passage des messages à travers plusieurs nœuds avant d`arriver au nœud destinataire.

2.5.2.5. Topologie en Étoile : La topologie en étoile est basé sur le nœud centrale (station de base) , il est le seule à pouvoir transmettre et recevoir un message via un certain nombre de nœud , ces nœuds ne peuvent pas communiquer entre eux mais seulement avec le nœud centrale Le nœud central joue un rôle de relais entre les différents nœuds , ce type de topologie est appliqué plus pour des applications d'avertissement Cette topologie présente des avantages qui peuvent être résumés par la simplicité, la faible consommation d'énergie des nœuds et la moindre latence de communication entre les nœuds et le nœud central. Par contre, son inconvénient majeur est la vulnérabilité du nœud central car tout le réseau est géré par un seul nœud.

Topologies	Point par point	Maillée	Etoile	Hiérarchique
Latence		✓	✓	

Redondance et tolérance aux fautes	✓	✓		✓
Consommation d'énergie		Élevée	Réduite	Élevée
Bonne couverture		✓		✓
Vulnérabilité du noeud			✓	
Bande passante			Élevée	Élevée
Echelle	✓			

Tableau 2.1. Tableau comparatif entre différentes topologies

2.5.3. Protocoles : Selon le mode d'établissement des routes Suivant la manière dont les routes sont créées et maintenues pendant le routage, nous distinguons trois catégories de protocoles de routage, les protocoles réactifs, les protocoles proactifs et les protocoles hybrides et aussi d'autres protocoles.

2.5.3.1. Routage basé sur la localisation Dans le routage basé sur la localisation, le routage est effectué en utilisant l'emplacement des nœuds. Selon la force des signaux entrants, il est possible de calculer la distance du nœud voisin le plus proche. Ils permettent la transmission directionnelle de l'information en évitant l'inondation d'information dans l'ensemble du réseau. Par conséquent, le coût de contrôle de l'algorithme est réduit et le routage est optimisé. De plus, avec la topologie réseau basée sur des informations de localisation de nœuds, la gestion du réseau devient simple.

2.5.3.2. Protocoles de routage multi-chemin Ces protocoles sont efficaces pour la gestion de plusieurs chemins. Les nœuds envoient les données collectées sur plusieurs liens au lieu d'un seul. Cela permet une bonne fiabilité et une bonne tolérance aux pannes dans le réseau car il existe un chemin alternatif lorsque le chemin primaire échoue.

2.5.3.3. Protocoles de routage basés sur les interrogations Le routage basé sur les interrogations propage les requêtes émises par la station de base. La station de base envoie des requêtes demandant certaines informations des nœuds du réseau, le nœud, qui est responsable de la détection et de la collecte des données, lit ces requêtes et s'il y a égalité avec les données demandées dans la requête commencent à envoyer les données à la station de base.

2.5.3.4. Protocoles de routage basés sur la négociation Ces protocoles utilisent des descripteurs de haut niveau codés en haut niveau afin d'éliminer la transmission des données redondantes. Chaque nœud effectue un suivi de la consommation de ses ressources ce qui influence ses décisions lors des négociations. Les négociations se font à travers des paquets d'avertissement, de requête et de données.

2.5.3.5. Protocoles de routage basés sur la qualité de service Dans ce type de protocole de routage, la qualité de service et l'énergie consommée doivent être maintenues dans le réseau. Chaque fois que le puits demande des données des nœuds dans le réseau, la transmission des données doit satisfaire certains paramètres de qualité de service, tels que, la latence (les données doivent être envoyées dès qu'elles sont détectées) et la bande passante consommée. Le protocole SAR (Sequential Assignment Routing) est l'un des premiers protocoles qui assure la qualité de service dans les réseaux de capteurs.

2.5.3.6. Protocoles de routage proactifs Les protocoles de routage proactif essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles (qui peuvent représenter l'ensemble de tous les nœuds du réseau), dans ce type de routage, les nœuds établissent les routes avant la demande, cette catégorie donne la plus grande importance à l'optimisation des chemins. Au niveau de chaque nœud du réseau les routes sont sauvegardées même si elles ne sont pas utilisées. La sauvegarde permanente des chemins de routage est assurée par un échange continu des messages de mise à jour des chemins.

2.5.3.7. Protocoles de routage réactifs Dits aussi protocoles de routage à la demande, créent et maintiennent les routes selon les besoins. Lorsqu'un nœud a besoin d'une route, une procédure de découverte globale est lancée. Cette procédure s'achève par la découverte de la route ou lorsque toutes les permutations de routes possibles ont été examinées. La route trouvée est maintenue par une procédure de maintenance de routes jusqu'à ce que la destination soit inaccessible à partir du nœud source ou que le nœud source n'ait plus besoin de cette route. Comparés aux protocoles proactifs qui conservent les routes vers l'ensemble des stations du réseau dans leur table de routage, les protocoles réactifs ne conservent que les routes qui ont une utilité. Par conséquent, la taille des tables de routage contenues en mémoire est moins importante que pour les protocoles proactifs.

2.5.3.8. Protocoles de routage hybrides Ce type de protocoles combine les mécanismes des protocoles proactifs et réactifs. Dans cette approche, les protocoles hybrides utilisent les méthodes proactives (messages périodiques de contrôle) pour découvrir les routes dans un voisinage prédéfini . Au-delà de la zone de voisinage, le protocole hybride fait appel à un protocole réactif pour obtenir les routes vers les nœuds destination.

2.6. Gestion d'énergie dans un RCSF :

2.6.1. Consommation d'énergie : Les nœuds capteurs, étant des dispositifs microélectroniques, peuvent être seulement équipés d'une batterie faible en énergie (< 0.5 Ah, 1.2V). Dans la plupart des applications, cette batterie est irremplaçable. La durée de vie du nœud capteur dépend fortement de la durée de vie de sa batterie. Cette énergie est consommée par les différentes unités du capteur afin de réaliser les tâches de collecte de données (captage), le traitement de ces derniers et leurs communications. La transmission de données est la tâche qui consomme le plus d'énergie. Par ailleurs, la plupart des applications des réseaux de capteurs sont basés sur la communication multi-sauts, chaque nœud joue à la fois un rôle d'initiateur de données et de routeur également, la défaillance d'un certain nombre de nœuds entraîne un changement significatif sur la topologie globale du réseau, et peut nécessiter un routage de paquets différent et une réorganisation totale du réseau. C'est pour cela que le facteur de consommation d'énergie est d'une importance primordiale dans les réseaux de capteurs.

2.6.2. Economie d'énergie : Dans le réseau de capteur sans fil la durée de vie est importante même primordiale c'est l'un des principaux élément vulnérable dans un réseau de capteur pour optimiser de l'énergie on doit maîtriser certain élément de la topologie (contrôle de la topologie ,maintenance de la topologie,constructions de la topologie ,contrôle de la puissance de transmission de la topologie).

2.6.2.1. Contrôle de la topologie Le contrôle de topologie est une technique utilisée dans les réseaux de capteurs sans fil. Son principe est de modifier, reorganiser et gérer certains paramètres et modes de fonctionnalités des nœuds dans la topologie initiale du réseau. Cette technique consiste à éliminer du réseau les nœuds inutiles (par la mise en sommeil des nœuds redondants) et les liens inutiles (par l'ajustement de la puissance de l'émetteur radio, et donc de la portée de communication), dans le but de diminuer la dépense d'énergie dans le réseau et prolonger sa durée de vie. Le contrôle de topologie regroupe deux contraintes: la

construction et la maintenance. La première contrainte est responsable de la réduction de topologie initiale, tandis que la seconde s'occupe de l'entretien de la topologie réduite tout en préservant les caractéristiques importantes du réseau comme la connectivité et la couverture. Cette topologie réduite doit être maintenue à jour de temps en temps car le réseau a besoin d'évoluer au fur et à mesure que des nœuds actifs arrivent à épuisement.

2.6.2.2. Construction de topologie Cette phase comporte trois principaux mécanismes, le contrôle de la puissance de transmission, les approches hiérarchiques et hybrides. Notre cas s'intéresse beaucoup plus au contrôle de la topologie qui permet l'ajustement de la puissance de transmission et le regroupement des nœuds capteurs (hiérarchisation).

2.6.2.3. Contrôle de la puissance de transmission Ce mécanisme n'a pas seulement un effet sur la durée de vie de la batterie d'un nœud capteur, mais aussi sur la capacité de charge du trafic qui est caractérisée par le nombre de paquets transmis avec succès vers une destination. En outre, il influe sur la connectivité et la gestion de la densité (le nombre de nœuds voisins). Ainsi, il peut conserver l'énergie à deux niveaux: explicitement par l'application de puissances faibles d'émissions et implicitement en réduisant la contention avec d'autres nœuds transmetteurs. Le contrôle de la puissance de transmission dans le cas homogène (c'est-à-dire le cas où tous les nœuds du réseau ont les mêmes caractéristiques), de manière centralisée est peut-être la technique la plus mature de la construction de topologie.

2.6.2.4. Maintenance de la topologie dans le but d'augmenter la durée de vie du réseau, l'ensemble des nœuds actifs ceux de la topologie réduite, ne peuvent pas être actifs tout le temps. Par conséquent, des mécanismes de maintenance de la topologie doivent être mis en place pour reconstruire une nouvelle topologie réduite avec la collaboration des nœuds inactifs de sorte que tous les nœuds appartenant au réseau consomment leur énergie de façon équitable et donc entraîne l'augmentation de la durée de vie du réseau. La maintenance de la topologie est définie comme étant le

processus qui restaure, fait pivoter, ou recrée la topologie du réseau de temps en temps lorsque le réseau réduit existant n'est plus optimal. Il consiste à faire tourner le rôle des nœuds autant que possible pour prolonger la durée de vie du réseau.

2.7. Conclusion : Nous remarquons la complexité de chaque couche de communication des données dans un réseau de capteurs sans fils. Les protocoles varient grandement selon les besoins des utilisateurs, le processus de leur sélection est délicat, encore plus leur processus de fonctionnement.

Alors, selon les exigences de notre projet, nous avions choisi le protocole ZigBee pour la réalisation d'un réseau physique de capteurs sans fils, étant le meilleur choix -surtout- en termes de consommation d'énergie (voir annexe p.).

Toutefois, pour cause de difficultés lors de l'acquisition du matériel nécessaire (capteurs ZigBee de température et d'humidité, coordinateur ZigBee, etc.), nous nous étions tournés vers une simulation d'un RCSF comme alternative, qui sera discuté dans le chapitre suivant.

Chapitre 3

Simulation d'un Réseau de Capteurs dans une Ferme à Serres

3.1. Introduction : Nous présenterons dans ce chapitre, une simulation d'un réseau de capteurs sans fils dans le but de surveiller les changements de températures et d'humidité dans différentes serres agricoles, utilisant les outils : MQTT, Python, Docker et InfluxDB.

3.2. Outils utilisés : Avant de comprendre le principe de cette simulation, on doit d'abord comprendre les outils utilisés dans sa programmation. Au total, ils sont quatre : **MQTT**, **Python**, **Docker** et **InfluxDB**.

3.2.1. MQTT : MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie qui permet la transmission de données entre des appareils de technologies différentes. Il est l'un des protocoles les mieux adaptés aux exigences du domaine de l'Internet des Objets grâce à sa moindre consommation d'énergie et sa bande passante minimale. Il permet de créer un serveur, autrement appelé "**Broker**", médiateur entre différents appareils nommés "**clients**". Ces clients peuvent se souscrire à des "**topics**" particuliers, ceux sont une forme d'adressage qui permet de catégoriser et d'acheminer les données entre les clients. Cependant, un client peut publier des données sur un topic particulier, pour qu'un autre client souscrit à ce topic les reçoive, c'est le principe **Publish/Subscribe**.

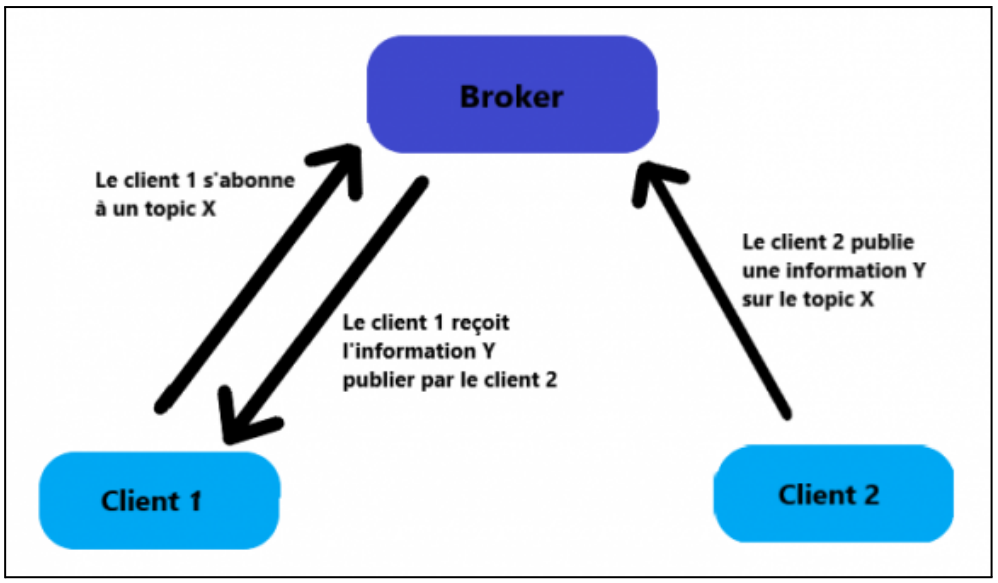


Figure 3.1. Principe du protocole MQTT

Figure 3.2. Logo de MQTT

3.2.2. Python : Python est un langage de programmation open source multi plateformes. Il est connu pour son adaptation à tous les niveaux de programmeurs : des débutants aux professionnels; et pour son multi usage : programmation d'applications, génération de codes, création de services web, etc.

Figure 3.3. Logo de Python

3.2.3. Docker : Docker est une espace virtuel de conteneurisation où plusieurs logiciels peuvent y être exécutés tout en étant isolés les uns des autres afin d'éviter les interférences entre les données qu'ils génèrent.

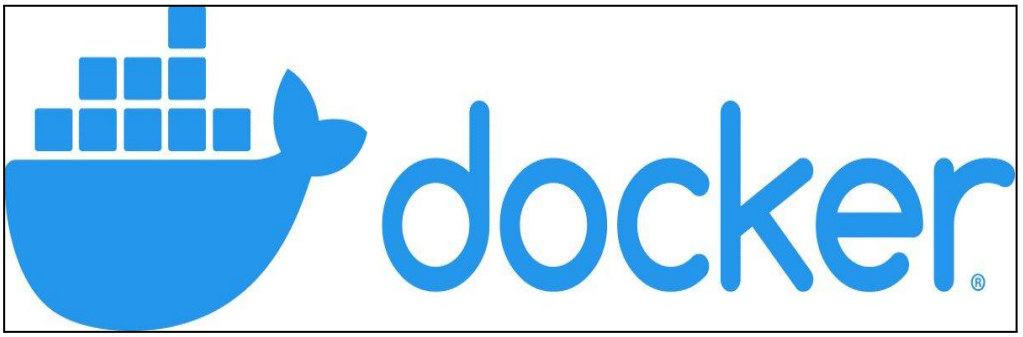


Figure 3.4. Logo de Docker

3.2.4. InfluxDB : InfluxDB est une TSDB (Time Series DataBase), autrement dit, un système de gestion de base de données qui permet la collecte, le stockage, le suivi et l'affichage du suivi de données chronologiques.

Logo 3.5. Logo de InfluxDB

3.3. Principe de la simulation : L'infrastructure de cette simulation est divisée en trois éléments : **Client 1**, **Serveur Cloud** et **Client 2**, qui sont illustrés et expliqués ci-dessous :

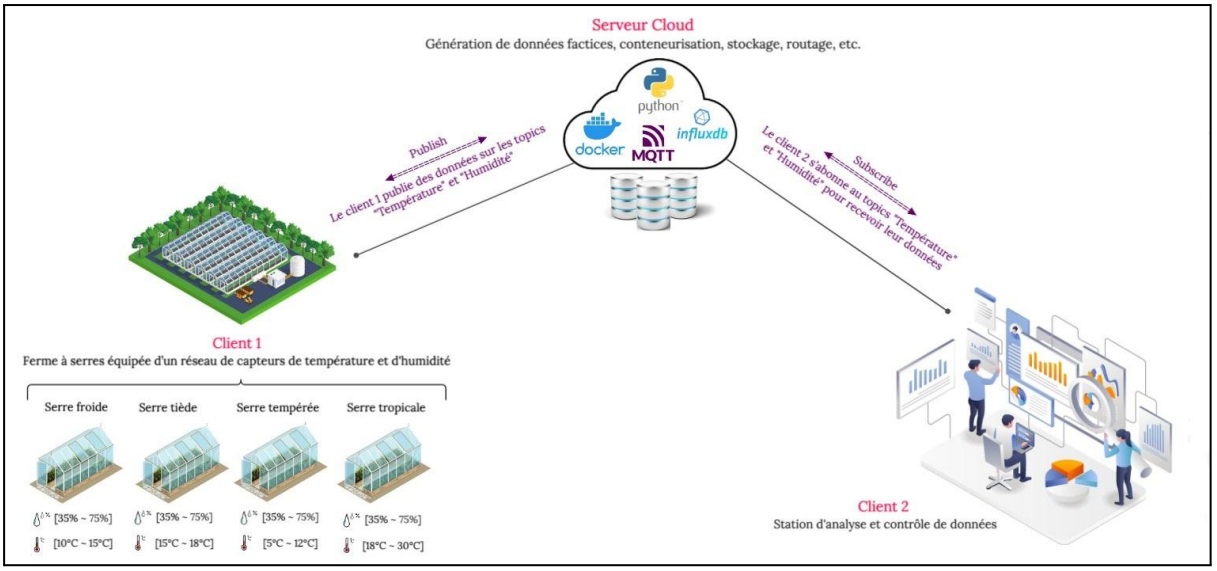


Figure 3.6. Principe de la simulation d'un RCSF dans une ferme à serres utilisant MQTT, Python, Docker et InfluxDB

- **Client 1 :**

  Virtualisons une ferme à quatre serres différentes : une serre froide, une serre tiède, une serre tempérée et une serre tropicale. La différence principale entre elles réside au niveau de leurs intervalles de températures idéales. Quant au taux de l'humidité idéal, il est approximativement le même pour la majorité des serres.

Serre	Température idéale	Taux d'humidité idéal
Froide	[10°C ~ 15]	[35% ~ 75%]
Tempérée	[5°C ~ 12°C]	[35% ~ 75%]
Tiède	[15°C ~ 18°C]	[35% ~ 75%]
Tropicale	[18°C ~ 30°C]	[35% ~ 75%]

Tableau 3.1. Températures et taux d'humidité idéaux de chaque type de serre

  Ces serres sont équipées d'un réseau de capteurs de température et d'humidité pour assurer l'entretien de l'environnement optimal des plantes de chaque serre, ce réseau de capteurs représente le Client 1.

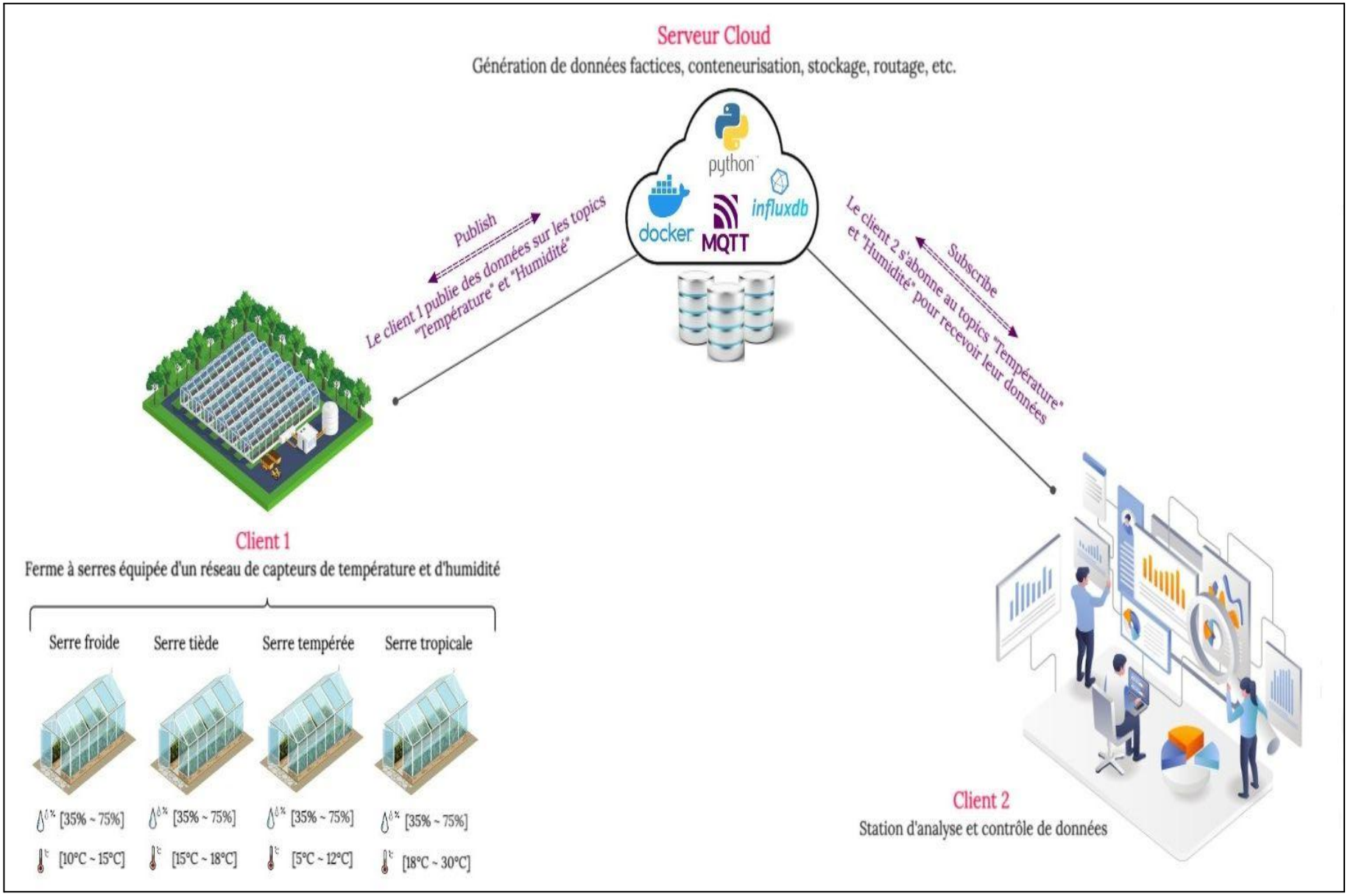
- **Serveur Cloud :**

  Le serveur Cloud est la plateforme virtuelle qui permet l'exécution des programmes en leur fournissant la puissance de calcul et l'espace de

stockage nécessaire. C'est sur cette plateforme que des données factices sont générées par un premier programme Python appelé **`smart_sensor.py`**, conteneurisées par Docker grâce à un programme YAML nommé **`docker-compose.yml`**, jusqu'à en arriver, à travers un deuxième programme Python appellé **`influxdb_consumer.py`** , à la base de données InfluxDB où elles sont temporairement stockées. Ces programmes Python sont dotés de la bibliothèque Eclipse Paho qui implémente un Broker MQTT. Les données factices générées représentent les données publiées par le Client 1 sur les Topics "Température" et "Humidité" que le Broker MQTT transmet au Client 2 déjà abonné à ces Topics.

- **Client 2 :**

  Le Client 2 représente la station d'analyse et de contrôle de données, ou les données reçues par le Broker MQTT sont enfin affichées en forme de graphes grâce à la plateforme InfluxDB. Il en résulte huit graphes au total : Quatre représentant les variations de températures idéales en fonction du temps, pendant trois heures, dans chaque type de serres et les autres quatre représentant les variations de taux d'humidité idéaux en fonction du temps, pendant trois heures, dans chaque type de serre.


Serveur Cloud
Génération de données factices, conteneurisation, stockage, routage, etc.
python
docker
MQTT
influxdb
Publish
Le client 1 publie des données sur les topics "Température" et "Humidité"
Subscribe
Le client 2 s'abonne au topics "Température" et "Humidité" pour recevoir leur données
Client 1
Ferme à serres équipée d'un réseau de capteurs de température et d'humidité
Serre froide
Serre tiède
Serre tempérée
Serre tropicale
[35% ~ 75%]
[35% ~ 75%]
[35% ~ 75%]
[35% ~ 75%]
[10°C ~ 15°C]
[15°C ~ 18°C]
[5°C ~ 12°C]
[18°C ~ 30°C]
Client 2
Station d'analyse et contrôle de données

3.4. Étapes de la programmation :

1. Tout d'abord, on génère des données factices par un programme Python qu'on nomme **`smart sensor.py :`**

```
"""
MQTT Smart temperature Sensor
"""
import time
import paho.mqtt.client as mqtt
from faker import Faker

# let's connect to the MQTT broker
MQTT_BROKER_URL     = "mqtt.eclipseprojects.io"
MQTT_PUBLISH_TOPIC = "temperature"
mqttc = mqtt.Client()
mqttc.connect(MQTT_BROKER_URL)

# Init faker our fake data provider fake = Faker()

# Infinit loop of fake data sent to the Broker
while True:
    temperature = fake.random_int(min=5, max=12)
    mqttc.publish(MQTT_PUBLISH_TOPIC, temperature)
    print(f"Published new temperature measurement:
{temperature}")
    time.sleep(1)
```

Les voici affichées sur le terminal après son exécution :

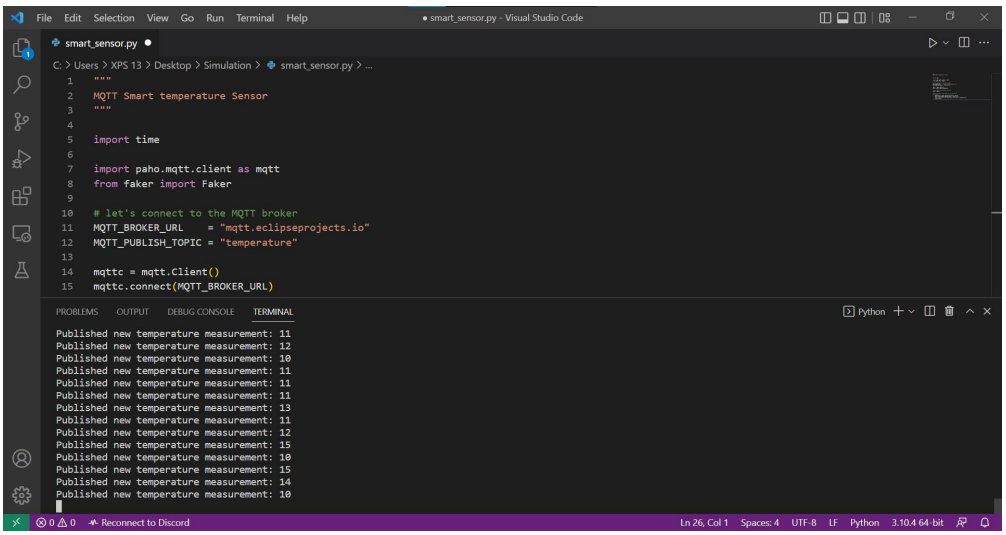


Figure 3.7. Terminal du programme smart sensor.py

2. Ensuite, on combine les plateformes Docker et InfluxDB :

   En définissant les variables de l'environnement requises dans le programme YAML, dans un fichier **.env** :

```
INFLUXDB_USERNAME=admin
INFLUXDB_PASSWORD=admin1234
INFLUXDB_TOKEN=F-QFQpmCL9UkR3qyoXnLkzWj03s6m4eCvYgDl1ePf
HBf9ph7yxaSgQ6WN0i9giNgRTfONwVMK1f977r_g71oNQ==
INFLUXDB_URL="http://localhost:8086/"
INFLUXDB_ORG=iot
INFLUXDB_BUCKET=temperature
```

   En écrivant le programme YAML nommé **`docker-compose.yml`** :

```yaml
version: '3.3'
services:
    influxdb:
        image: influxdb:2.0.7
        environment:
            DOCKER_INFLUXDB_INIT_MODE: setup
            DOCKER_INFLUXDB_INIT_USERNAME:
${INFLUXDB_USERNAME}
            DOCKER_INFLUXDB_INIT_PASSWORD:
${INFLUXDB_PASSWORD}
            DOCKER_INFLUXDB_INIT_ORG: ${INFLUXDB_ORG}
            DOCKER_INFLUXDB_INIT_BUCKET: ${INFLUXDB_BUCKET}
            DOCKER_INFLUXDB_INIT_ADMIN_TOKEN:
${INFLUXDB_TOKEN}
        ports:
        - "8086:8086"
```

   et le programme Python nommé **`influxdb_consumer.py`** :

```python
"""
MQTT subscriber - Listen to a topic and sends data to
InfluxDB
"""
import os
from dotenv import load_dotenv
from influxdb_client import InfluxDBClient, Point
import paho.mqtt.client as mqtt
import struct
load_dotenv()
```

```
# take environment variables from .env.
# InfluxDB config
BUCKET = os.getenv('INFLUXDB_BUCKET')
client = InfluxDBClient(url=os.getenv('INFLUXDB_URL'),
                token=os.getenv('INFLUXDB_TOKEN'),
org=os.getenv('INFLUXDB_ORG'))
write_api = client.write_api()

# MQTT broker config
MQTT_BROKER_URL    = "mqtt.eclipseprojects.io"
MQTT_PUBLISH_TOPIC = "temperature"

mqttc = mqtt.Client()
mqttc.connect(MQTT_BROKER_URL)

def on_connect(client, userdata, flags, rc):
    """ The callback for when the client connects to the
broker."""
    print("Connected with result code "+str(rc))

    # Subscribe to a topic
    client.subscribe(MQTT_PUBLISH_TOPIC)

def on_message(client, userdata, msg):
    """ The callback for when a PUBLISH message is received
from the server."""
    print(msg.topic+" "+str(msg.payload))

    # We received bytes we need to converts into something
usable
    measurement = int(msg.payload)

    ## InfluxDB logic
    point = Point(MQTT_PUBLISH_TOPIC).tag("location", "Serre
froide").field("temperature", measurement )
    write_api.write(bucket=BUCKET, record=point)

## MQTT logic - Register callbacks and start MQTT client
mqttc.on_connect = on_connect
mqttc.on_message = on_message
mqttc.loop_forever()
```

Voici ce qu'on voit sur le terminal après son exécution :

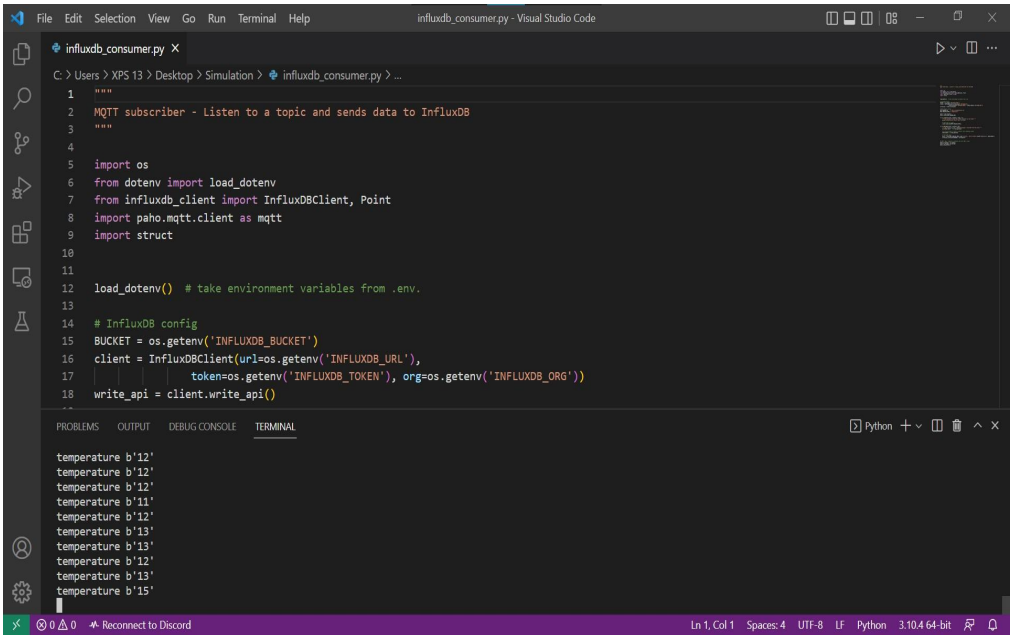


Figure 3.8. Terminal du programme influxdb_consumer.py

3. On met ces quatre fichiers dans un même dossier qu'on copie sur une invite commande, puis, on exécute la commande :

```
docker-compose --env-file .env up
```

Voici le conteneur qui se crée par la suite sur Docker :

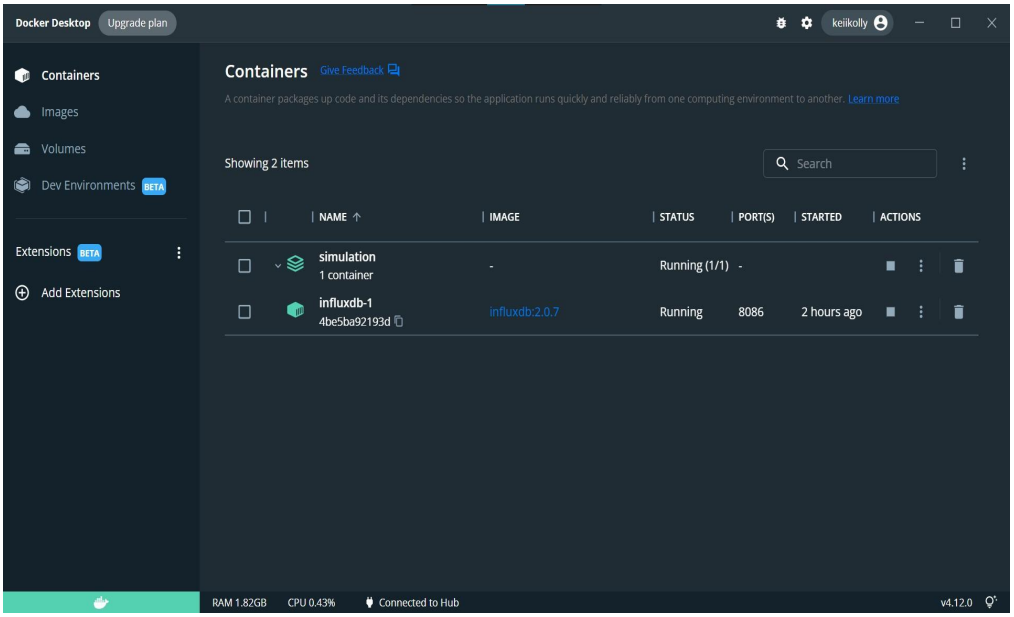


Figure 3.9. Conteneur Docker après l'exécution des programmes

Et voici ce que s'affiche sur l'invite de commande :

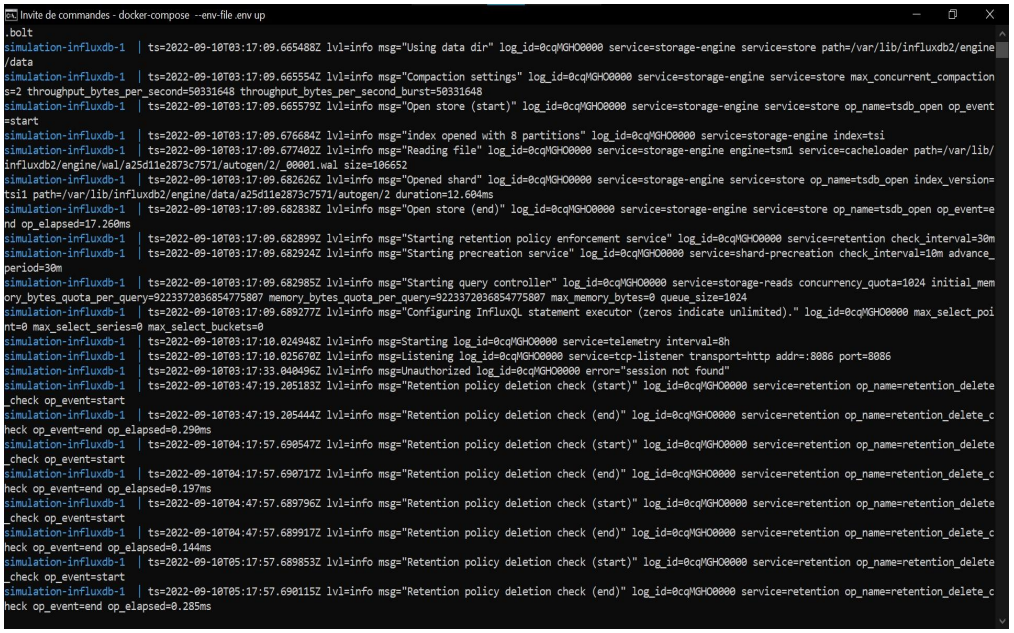


Figure 3.10. Invite de commande après l'exécution des programmes

4. On se connecte à la plateforme InfluxDB à travers le lien http://localhost:8086/ en introduisant les données :

   Nom d'utilisateur : **admin**
   Mot de passe : **admin1234**

5. On affiche enfin les graphes sur InfluxDB, en attendant la collecte de données pendant trois heures pour chaque graphe.

3.5. Graphes résultants :

Figure 3.11. Variation de températures dans une serre froide

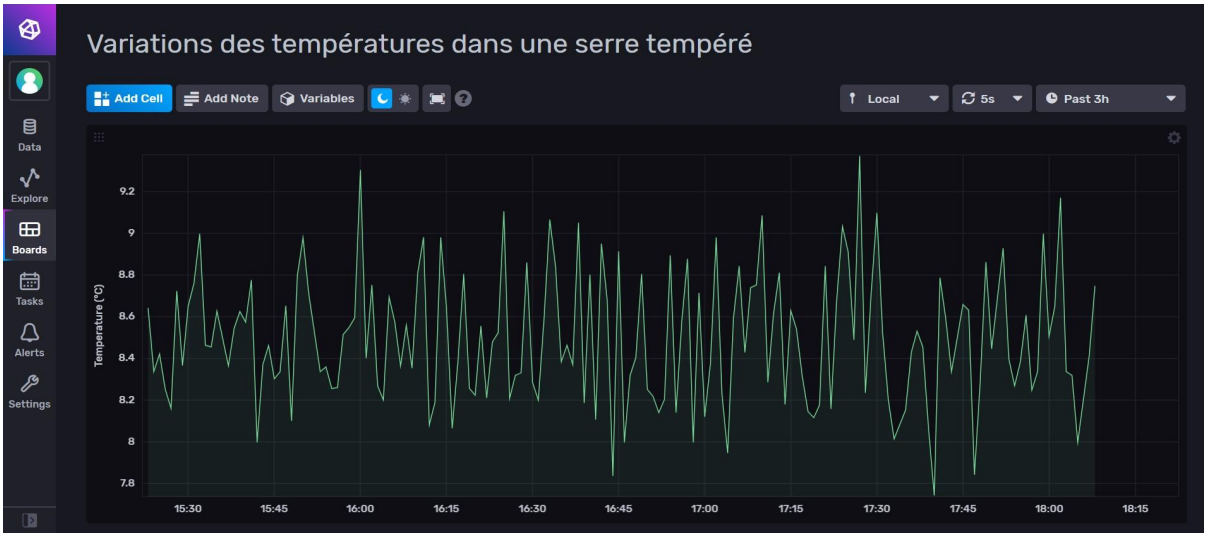


Figure 3.12. Variation de températures dans une serre tempérée

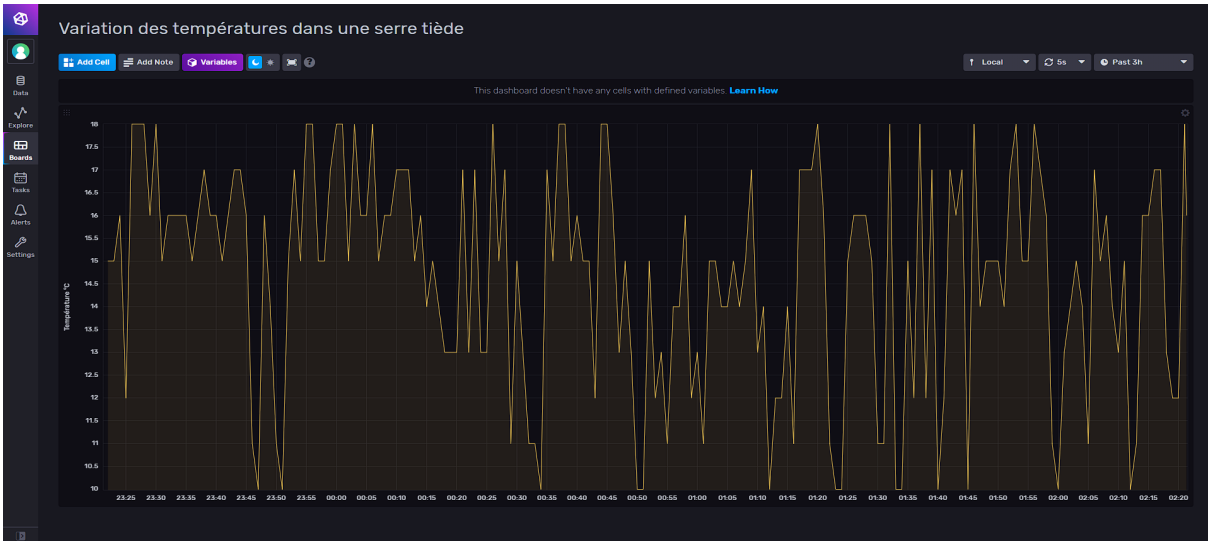


Figure 3.13. Variation de températures dans une serre tiède

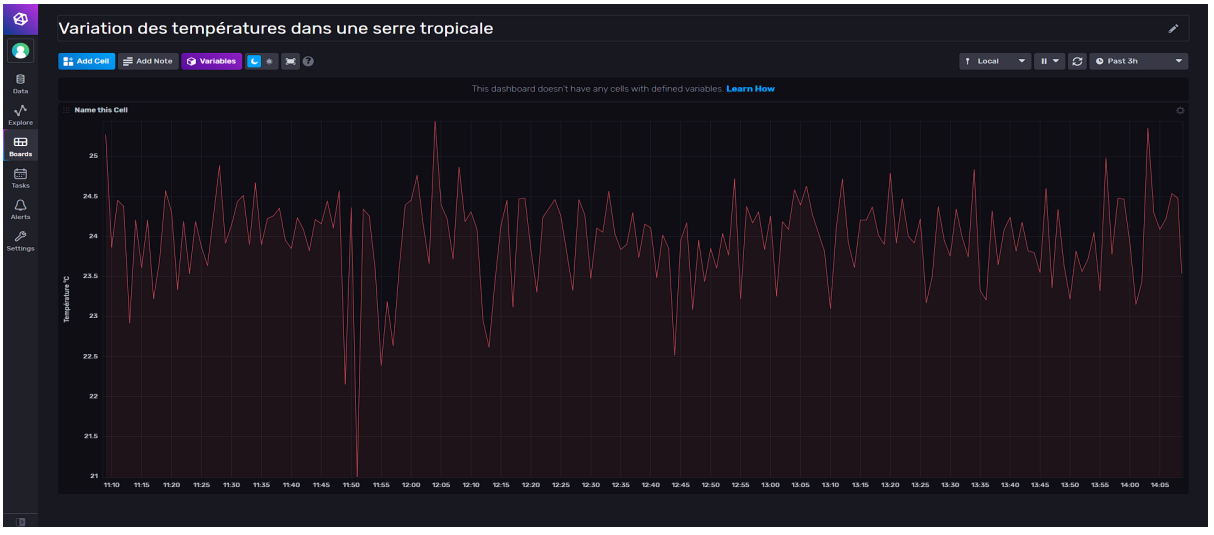


Figure 3.14. Variation de températures dans une serre tropicale

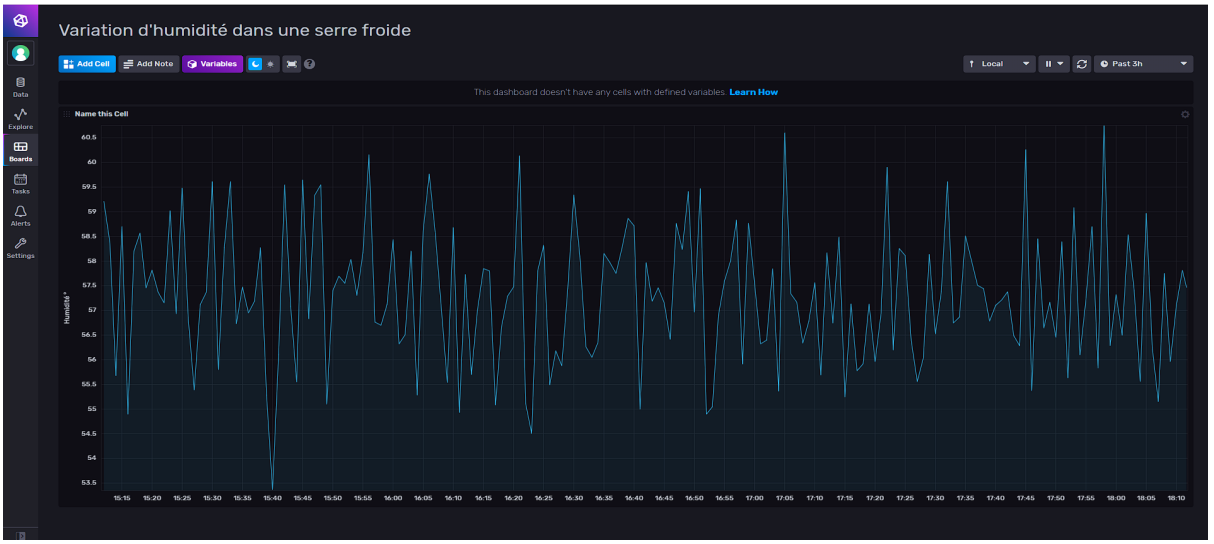


Figure 3.15. Variation d'humidité dans une serre froide

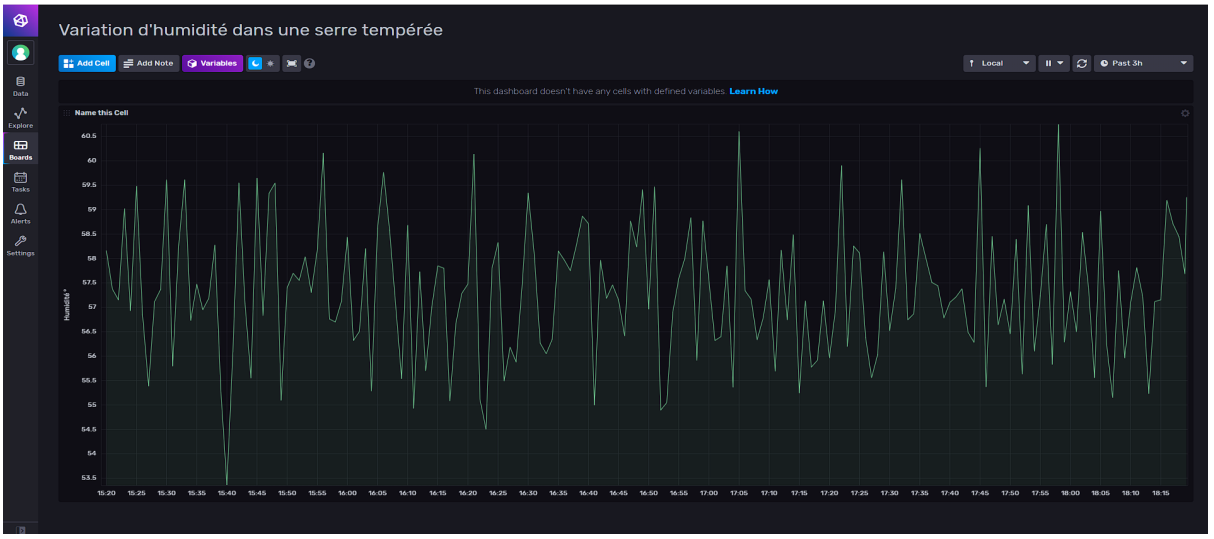


Figure 3.16. Variation d'humidité dans une serre tempérée

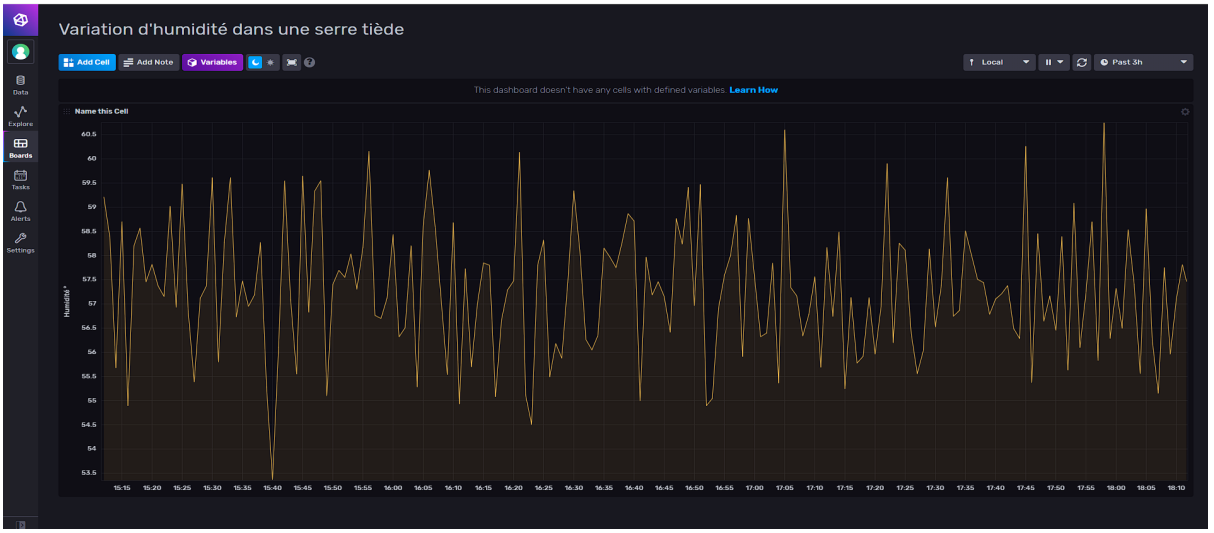


Figure 3.17. Variation d'humidité dans une serre tiède

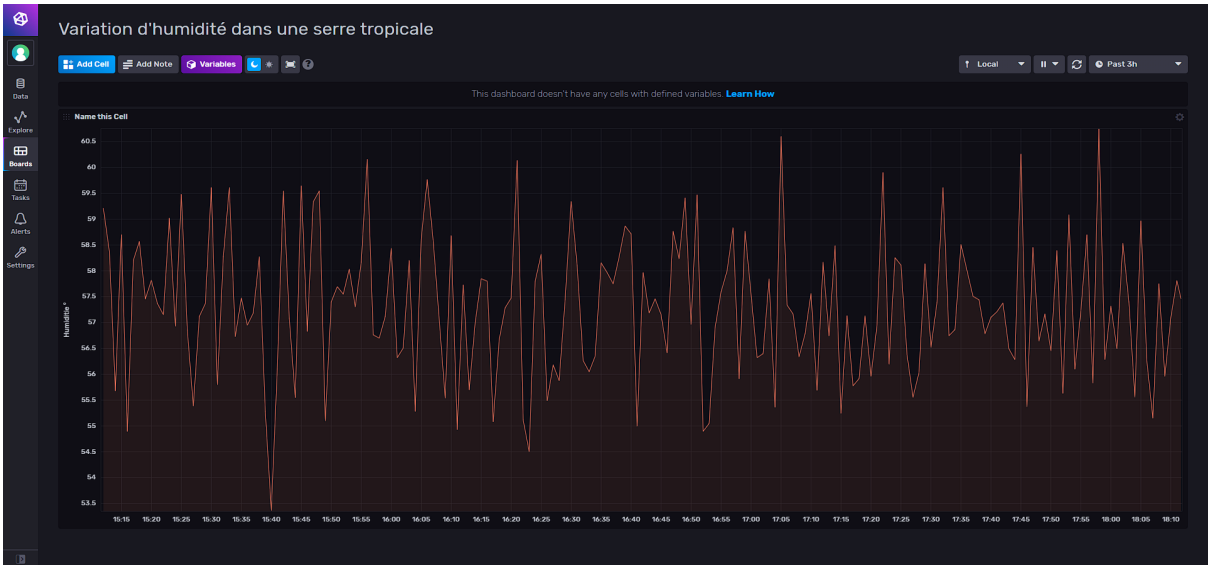


Figure 3.18. Variation d'humidité dans une serre tropicale

3.6. Analyse de la simulation : L'exécution des programmes a apporté, avec succès, le résultat attendu : Des graphes montrant le suivi de données collectées représentatives de variations de températures et d'humidité dans différents types de serres agricoles au cours de trois heures.

Par contre, on remarque quelques données erronées sur les graphes telles que l'apparition de valeurs de température non incluses dans l'intervalle de températures prédéfini dans le programme et d'autres carrément nulles. Ceci est dû à plusieurs facteurs : les capacités de calcul et de stockage des logiciels utilisés, des processeurs de PCs sur lesquels les programmes ont été exécutés, du débit d'internet lors de l'exécution des programmes, etc. À noter que la simulation correspondante à chaque serre n'a pu durer plus de trois heures pour cause de surchauffe des PCs et par la suite leur plantage. Ce type de dysfonctionnement est présent même dans la réalité, où les réseaux de capteurs et les stations de bases sont dotés de plus grandes capacités de traitement et de stockage.

Il est important d'offrir aux agriculteurs gestionnaires de réseaux de capteurs de paramètres environnementaux, des informations précises sur le fonctionnement de leurs serres afin de leur permettre de prendre les bonnes précautions afin d'éviter les maladies végétales et les pertes financières. C'est pour cela qu'il est impératif de remédier à ces problèmes de qualité de données grâce à l'existence de différents outils de nettoyage de données, implémentables dans les simulateurs de réseaux de capteurs et dans les réseaux de capteurs réelles.

3.7. Conclusion : Nous avions réussi à générer des données factices simulant les changements de températures et d'humidité dans une serre grâce à l'outil Python, les transporter par la suite, grâce à l'outil Docker, jusqu'à la plateforme de catégorisation et d'affichage de données InfluxDB, le tout en suivant le protocole de communication sans fil MQTT.

Conclusion générale

Les réseaux de capteurs sans fils sont capables de créer un langage entre les agriculteurs et leurs plantations, chose qui est révolutionnaire car elle optimise non seulement la qualité de l'environnement de travail des agriculteurs mais aussi la qualité et la productivité de l'agriculture, un domaine qui touche tous les humains.

ANNEXE

PROTOCOLE DE COMMUNICATION SANS FIL « ZIGBEE »

1. **Architecture ZigBee/IEEE 802.15.4 :** Les variations des signaux radio en fonction du temps et de la distance ont toujours été un problème majeur dans les réseaux sans fil. Ce sont des systèmes très complexes dont leurs conception, exploitation et analyse exigent de grands coûts à grande échelle, surtout en matière de consommation d'énergie. Toutefois à petite échelle, les applications de réseaux sans fils dans l'agriculture, la sécurité, la domotique… n'exigent pas autant de complexité, d'énergie ou de coûts. Les précédentes normes ne répondaient pas à ces exigences, une nouvelle norme devait être établie, c'est la norme ZigBee.

   Le ZigBee est un protocole de communication des niveaux 3/7 du modèle OSI, développé par la ZigBee Alliance (+50 compagnies), basé sur la norme 802.15.4, un protocole de communication du niveau 1/2 du modèle OSI, développé par l'IEEE (Institute of Electrical and Electronics Engineers).

Figure 2.2. Logo de IEEE 802.15 Group

Figure 2.3. Logo de Zigbee

L'organisation	Les couches qu'elle	Leurs	Leurs

<table>
<tr><th></th><th>définit</th><th>sous-couches</th><th>protocoles</th><th rowspan="6">AES 128 bit</th></tr>
<tr><td rowspan="4">ZigBee Alliance</td><td rowspan="4">4. Couche APL</td><td>3. AF</td><td>/</td></tr>
<tr><td>2. APS</td><td>/</td></tr>
<tr><td>1. ZDO</td><td>/</td></tr>
<tr><td>3. Couche NWK</td><td>/</td><td>AODV</td></tr>
<tr><td rowspan="2">IEEE 802.15.4 Working Group</td><td>2. Couche MAC</td><td>/</td><td>CSMA/CA</td></tr>
<tr><td>1. Couche PHY</td><td>/</td><td>/</td></tr>
</table>

Tableau 2.2. Couches de Zigbee/IEEE 802.15.4

Norme IEEE 802.15.4 :

Le 802.15.4 est un protocole de communication des niveaux 1/2 du modèle OSI. Il est spécifiquement destiné aux réseaux sans fil de la famille des LR WPAN (Low Rate Wireless Personal Area Network) : Des réseaux sans fil de faible portée, à faible débit et faible consommation d'énergie.

Ce protocole définit les couches inférieures “PHY” et “MAC” :

Caractéristiques principales de la couche PHY :

- ➢ Portée radio maximale de 100 mètres, sans répéteurs.
- ➢ Quantité d'énergie nécessaire pour transmettre les données : Min = -3 dBm (0.5 mW) ; Max = 20 dBm (100 mW).
- ➢ Les fréquences définies dans cette norme sont réparties sur 27 canaux différents répartis en trois bandes principales avec leurs débits associés :

Canaux	Bande de fréquences	Type de modulation	Débits (Kb/s)
1 canal (Europe)	[868.0 – 868.6MHz]	BPSK O-QPSK ASK	20 100 250
10 canaux (États Unis, Australie)	[902.0 – 928.0MHz]	BPSK O-QPSK ASK	40 250 250
16 canaux (Monde entier)	[2.40 – 2.48GHz]	O-QPSK	250

Tableau 2.3. Bandes de fréquences des différents canaux

Modulation O-QPSK : L'O-QPSK (Offset Quadrature Phase Shift Keying) est une transmission en bande transposée (numérique ↔ analogique).

Caractéristiques principales de la couche MAC :
La couche MAC (Media Access Control) gère les accès au canal radio.
Pour cela, la norme 802.15.4 définit le protocole CSMA/CA.

- Protocole CSMA/CA :
  On nomme parmi les facteurs pouvant affecter le signal et donc les bits transmis sur le canal, les collisions. Une collision est la perte de données lorsque deux appareils d'un même réseau tentent de transmettre des données exactement au même moment, et mélangent accidentellement leurs trames. Pour réduire cette probabilité, le protocole CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) est né. Il est spécialement conçu pour les réseaux sans fils tels que le Wi-Fi. Il vérifie la disponibilité du canal pour une transmission : Si le canal est inactif, la transmission est permise, sinon, il attend jusqu'à ce que la transmission en cours soit achevée. En plus, il peut détecter et résoudre les erreurs.

- Il y a deux modes de transmission de données dans la couche MAC :

  1. Non-Beacon mode : (Protocole CSMA/CA)
     L'émetteur vérifie si canal est libre avant l'émission. Si oui, la transmission se produit, sinon, il attend une période aléatoire. Ce mode est surtout utilisé pour les capteurs qui dorment la plupart du temps, afin d'optimiser la durée de vie de la batterie des capteurs et afin d'éviter les collisions des trames également.

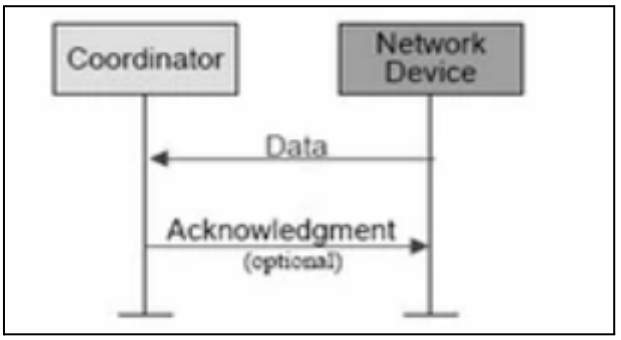


Figure 2.4. Mode Non-Beacon

  2. Beacon mode :
     Toutes les entités du réseau fonctionnent de manière indépendante, mais pour communiquer, ils doivent savoir quand se réveiller pour transmettre les données entre elles. Pour cela, il faut

qu'elles se synchronisent avec le réveil du coordinateur grâce à une balise “Beacon”. Lorsque cette balise est reçue, tous les appareils sont informés de la durée de la période d'activité du coordinateur et du moment où les appareils peuvent transmettre des données utiles. Ils recevront également une indication pour savoir quand le coordinateur entre en hibernation et pour combien de temps. Ainsi, les appareils savent quand s'allumer et quand s'éteindre.

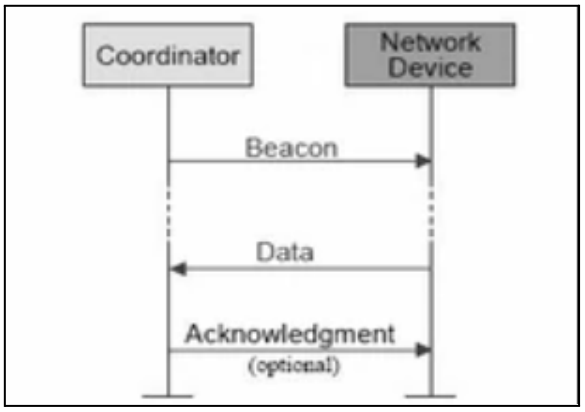


Figure 2.5. Mode Beacon

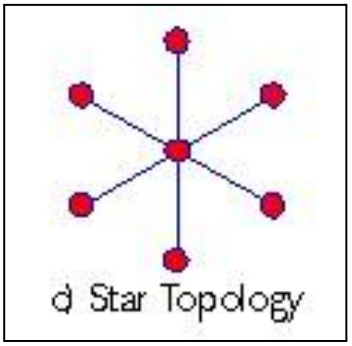
Norme ZigBee :

Le ZigBee est un protocole de communication des niveaux 3/7 du modèle OSI, basé sur la norme 802.15.4. Il y est souvent combiné pour définir les couches supérieures “NWK” (Network=Réseau) et “APL” (Application).

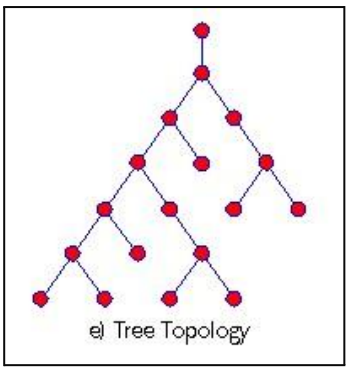
Caractéristiques principales de la couche NWK :

- Multiplexage de données selon 3 types de topologies :

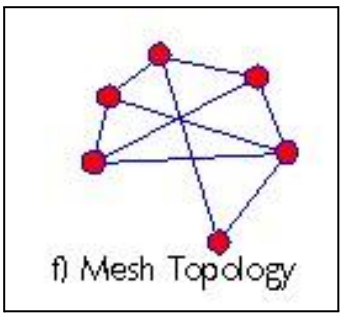
  - Topologie Étoile :



  - Topologie Hiérarchique :



- Topologie Maillée (Point à Point) :



➢ Deux types d'entités réseau :.

1. FFD : Les FFD (Full Function Device) ont trois rôles possibles :
   - ➔ Coordinateurs PAN (Personal Area Network) (=Sink)
   - ➔ Routeur (=Noeuds de capteurs)
   - ➔ Dispositif terminal (End-Device) (=Capteurs)

2. RFD : Les RFD (Reduced Function Device) n'ont que le rôle de End Device.

Les FFD peuvent communiquer avec des FFDs et des RFDs, mais les RFDs ne peuvent communiquer qu'avec des FFDs.
Exemple représentant les FFD et RFD dans une topologie Étoile :

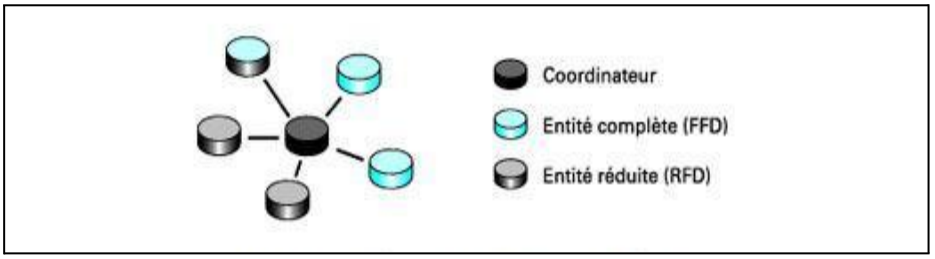


Figure 2.6. FFDs et RFDs

➢ Deux modes d'adressage :

1. Adresse à 16 bits, attribuée par l'administrateur du réseau à chaque nœud du réseau, locale et donc altérable d'un réseau à un autre.
2. Adresse à 64 bits, attribuée par le fabricant de l'appareil, unique à chaque appareil du réseau et fixe.

➢ Protocole AODV (Ad hoc On Demand Distance Vector) :
C'est un protocole de routage réactif (Il achemine les données entre les nœuds du réseau à la demande seulement et donc élimine tout trafic inutile), conçu pour les réseaux "ad hoc" (Aussi appelés "MANET"), les RCSF sont un exemple de ce type de réseaux. Il crée une table de routage, à la demande, pour chaque nœud et maintient seulement les routes actives pour une certaine durée.

Caractéristiques principales de la couche APL :

➢ Contient 3 sous-couches :

1. Zigbee Device Object (ZDO) :
Cette couche est responsable de la liaison entre la couche NWK et la couche APL (Spécifiquement la sous-couche APS).

2. Application Support Sublayer (ASP) :
Cette couche est responsable de la liaison entre la couche APL et la couche AF. C'est sur cette couche qu'ait lieu la dernière étape de conversion de données de langage machine (haut niveau) vers un langage assembleur (bas niveau) qui seront ensuite transmises vers la couche AF où les données sont enfin converties en langage lisible par l'utilisateur.

3. Application Framework (AF) :
Cette sous-couche correspond aux applications qui permettent la maintenance du réseau à l'utilisateur. Ces applications sont spécifiées par le vendeur des appareils du réseau.
On en cite : Nordic, TinyOs, etc.

N.b : Pour la sécurité du réseau, la norme 802.15.4 propose l'algorithme de cryptage "AES 128 bit", sur lequel sont basées toutes les couches de l'architecture 802.15.4/ZigBee.

2. **Zigbee2MQTT :** Il s'agit d'un serveur passerelle capable de réceptionner les informations d'un coordinateur ZigBee et de les envoyer vers notre broker MQTT sous forme de topics, auxquels différents clients sont abonnés pour les recevoir. Ce même concept sera simulé dans le chapitre suivant.

Figure 2.7. Logo de Zigbee2MQTT

Les prérequis d'installation de Zigbee2MQTT sont définies sur son officiel comme étant 3 éléments indispensables : Capteurs Zigbee, Coordinateur Zigbee et Raspberry Pi.

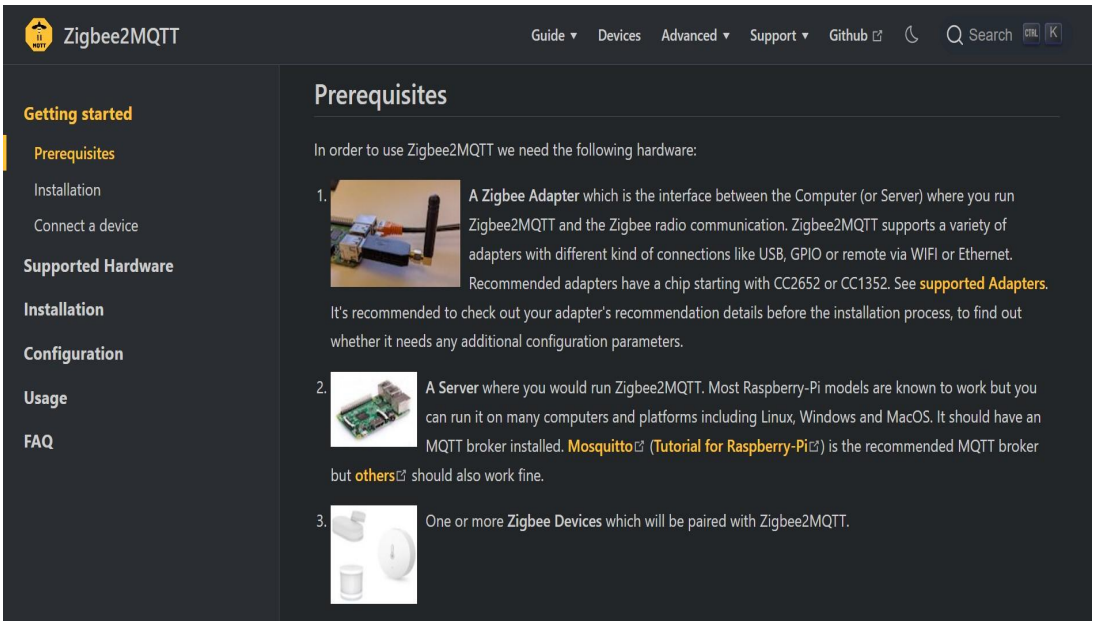


Figure 2.8 Tutoriel d'installation Zigbee2MQTT

Figure 2.9. Capteur de température et d'humidité Zigbee

Figure 2.10. Coordinateur Zigbee

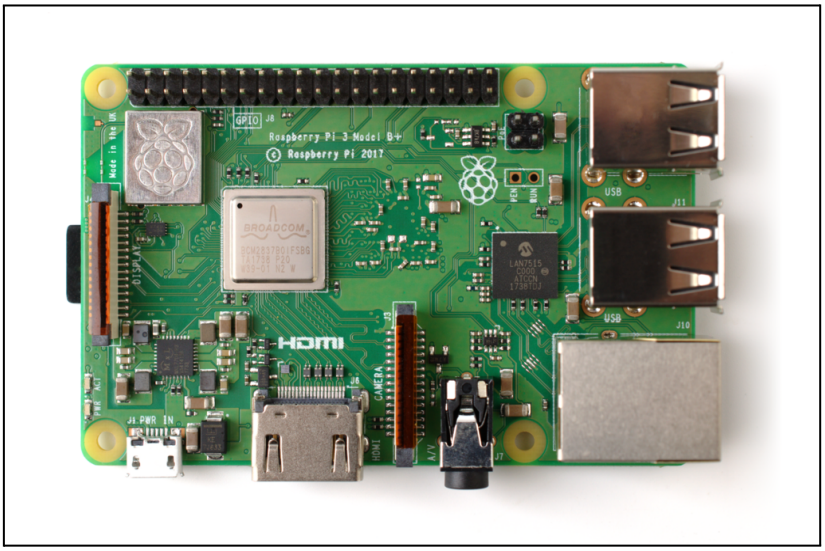


Figure 2.11. Raspberry Pi

Bibliographie

1. https://thenewstack.io/python-mqtt-tutorial-store-iot-metrics-with-influxdb/
2. https://github.com/xNok/python-influxdb-mqtt-tutorial
3. https://www.docker.com/https://qkzk.xyz/docs/nsi/cours_premiere/programmation/outils/libraries_faciles/mqtt/#mosquitto
4. https://www.influxdata.com/
5. https://docs.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package
6. http://www.steves-internet-guide.com/into-mqtt-python-client/
7. https://standards.ieee.org/ieee/802.15.4/7029/
8. https://www.zigbee2mqtt.io/
9. https://csa-iot.org/all-solutions/zigbee/