Université de Technologie de Compiègne
SR04 – Réseaux Informatiques

Semestre Automne 2025

Objets connectés pour l'environnement

Application expérimentale d'une architecture IoT distribuée

Groupe n°2

CHENGUEL Nourane
KHEMIRA Youssef
CHAZELAS Clément
THÉOLIER Remy

Encadrant

Abdelmadjid BOUABDALLAH

Rapport académique – UTC

Table des matières

1 Introduction

L'Internet of Things (Internet des objets) transforme progressivement notre rapport au monde physique. Il s'agit de réseaux constitués de nombreux dispositifs connectés tels que des capteurs, des balises ou des stations autonomes, qui permettent ensemble d'obtenir une grande quantité d'informations sur le monde qui nous entoure, et améliorent notre confort de vie sur de nombreux aspects. Le développement de ces réseaux s'explique par plusieurs évolutions techniques : miniaturisation des composants, démocratisation des réseaux sans fil, développement d'algorithmes capables de traiter des volumes croissants de données.

Cette évolution technologique intervient dans un contexte particulier. En effet, le dérèglement climatique s'accélère, la pollution atmosphérique touche des populations toujours plus nombreuses, et la gestion des ressources naturelles devient une question centrale pour les décennies à venir. Face à ces défis, il est nécessaire de disposer d'informations fiables et actualisées, notamment car les décisions politiques, industrielles ou agricoles reposent de plus en plus sur ces données de terrain plutôt que sur des estimations approximatives.

C'est là que l'IoT trouve une application particulièrement pertinente. Ces objets connectés permettent une surveillance continue des écosystèmes, des milieux urbains, des sols cultivés. Ils offrent une granularité d'observation que les méthodes classiques ne peuvent pas atteindre : un réseau de capteurs peut révéler des variations locales invisibles aux satellites, détecter des anomalies en temps réel, et alerter avant qu'une situation ne devienne critique. Leur déploiement dans des contextes variés tels que l'agriculture de précision, les villes intelligentes ou encore le contrôle de la qualité de l'air modifie les pratiques habituelles. Par exemple, un agriculteur peut ajuster l'irrigation selon l'humidité réelle du sol ; une municipalité peut identifier les zones où la pollution dépasse les seuils recommandés ; ou bien un gestionnaire forestier peut surveiller l'évolution d'un écosystème fragile sans intervention humaine constante.

Ce rapport examine comment l'IoT peut se mettre au service de la surveillance environnementale. Il détaille les principes qui régissent ces systèmes, les composants techniques qu'ils mobilisent, et propose une réflexion critique sur leur empreinte écologique, car ces outils de mesure environnementale ont eux-mêmes un impact qui ne peut être négligé. Enfin, un cas d'application illustrera ces concepts à travers un exemple concret de déploiement.

2 État de l'art

L'Internet des Objets destiné à la surveillance environnementale repose sur une architecture en plusieurs couches. À la base se trouvent les capteurs, qui mesurent les grandeurs physiques du milieu observé. Ces données brutes sont ensuite transmises par des passerelles de communication, qui constituent des ponts entre le terrain et l'infrastructure numérique. Une fois transmises, elles arrivent vers des unités de traitement qui les filtrent et les stockent. Enfin, la couche applicative permet aux gestionnaires, chercheurs ou décideurs d'exploiter ces informations sous forme visuelle grâce à une interface utilisateur, de mettre en place des alertes, ou encore d'automatiser des tâches en fonction de l'état des données.

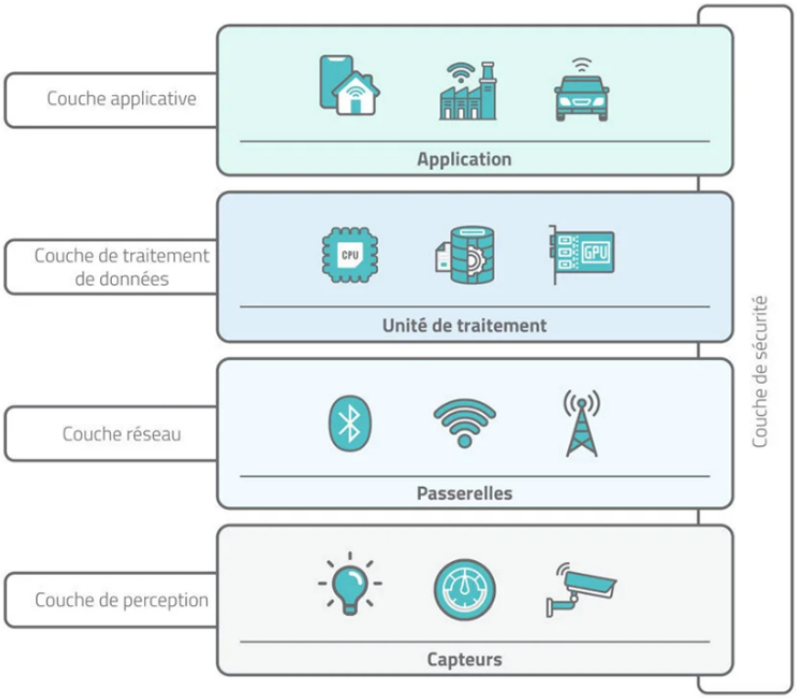

FIGURE 1 – Architecture en couches d'un système IoT.

Dans le cadre environnemental, ces architectures servent à mesurer et analyser des paramètres tels que la température, l'humidité, la qualité de l'air ou la pollution de l'eau. Chaque couche de l'architecture répond à des contraintes particulières : la robustesse pour les capteurs exposés aux intempéries, une bonne portée et une efficacité énergétique pour les réseaux, une capacité de calcul suffisante pour le traitement, et l'ergonomie pour l'interface utilisateur. Aussi, la fiabilité des données est un enjeu crucial pour ces systèmes : un capteur mal calibré ou une latence réseau trop importante peut fausser les mesures et compromettre les décisions qui en découlent.

Il y a plusieurs stratégies de déploiement qui ont émergé au cours de ces dernières années. On distingue principalement deux méthodes qui ont chacune leurs avantages et inconvénients. D'une part, on trouve les systèmes qui privilégient une architecture centralisée, dans laquelle les données convergent vers une plateforme unique de traitement, ce qui permet de faciliter la supervision du système mais crée une dépendance forte au réseau. D'autre part, certains optent pour une approche distribuée, où l'on effectue le traitement au plus près des capteurs afin de réduire cette dépendance et les besoins en bande passante ; on a ainsi une meilleure réactivité face aux événements critiques, mais les coûts et la maintenance d'un système de ce type peuvent représenter une contrainte. Le choix entre ces deux modèles dépend principalement de la géographie, de la criticité des mesures et des infrastructures existantes.

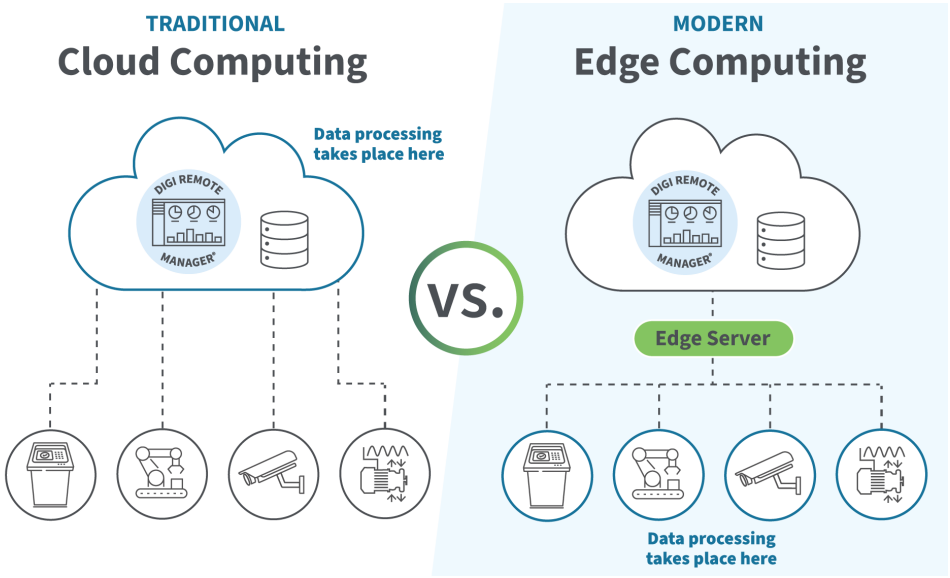

FIGURE 2 – Comparaison des architectures Cloud Computing (centralisée) et Edge Computing (distribuée).

Selon la littérature étudiée, ce ne sont pas les seuls défis rencontrés. Une autre préoccupation majeure est l'autonomie des dispositifs, particulièrement lorsqu'une intervention humaine peut représenter un coût et un risque élevés, par exemple dans des environnements difficiles. Il y a également l'interopérabilité entre les équipements de différents constructeurs à cause des standards propriétaires, ou encore la difficulté organisationnelle lors du passage à une échelle plus importante.

Les villes représentent un bon terrain d'expérimentation pour l'IoT environnemental. Par exemple, plusieurs métropoles ont déployé des réseaux de surveillance de la qualité de l'air, qui révèlent des variations significatives entre différents quartiers. Ces variations ne pouvaient être détectées par les stations officielles car l'échelle est plus petite que ce que la station peut fournir comme précision. Cela est rendu possible par la granularité

des réseaux IoT, qui ont permis à ces municipalités d'adapter localement leurs politiques, notamment en termes de circulation de véhicules. Mais les capacités des réseaux IoT ne se limitent pas à la pollution. En effet, certaines initiatives concernent également la réduction du bruit urbain, l'optimisation de la collecte des déchets ou encore le suivi de la consommation énergétique des bâtiments publics. L'intérêt de ces systèmes est la capacité à croiser les différentes sources d'informations pour obtenir une vision détaillée et fiable de la situation.

L'IoT peut également être exploitée pour la surveillance des milieux naturels, qui posent des contraintes spécifiques liées à l'isolement et aux conditions environnementales difficiles. Plusieurs projets ont déjà démontré la faisabilité technique du suivi en continu de lacs, rivières ou zones forestières, qui ont permis d'anticiper des phénomènes comme la prolifération d'algues toxiques ou l'évolution du risque d'incendie. Les gestionnaires d'espaces protégés y trouvent un moyen de documenter l'impact du dérèglement climatique sur les écosystèmes et d'adapter leurs stratégies de conservation. Les données collectées alimentent également la recherche scientifique, les données collectées étant des séries temporelles longues et détaillées, exploitables pour des études et de l'analyse. Toutefois, la pérennité de ces installations n'est pas garantie : les équipements se détériorent à cause des intempéries, de la faune ou même parfois du vandalisme, et leur maintenance dans des sites difficiles d'accès est coûteuse. Le modèle économique de ces déploiements n'est pas toujours stable, variant entre financements publics de recherche et expérimentations portées par des organismes de gestion.

Ces différentes expériences montrent que l'IoT environnemental n'est plus un simple concept mais une réalité opérationnelle déjà sérieusement employée. Les architectures techniques convergent vers des principes qui ont fait leurs preuves, même si chaque contexte impose ses propres spécificités. Les bénéfices mesurés tels que les économies de ressources, la détection précoce d'anomalies ou l'aide à la décision justifient de poursuivre les investissements, malgré les limites actuelles en termes de coût, de robustesse et de simplicité d'usage.

3 Composants matériels (Hardware)

L'infrastructure matérielle d'un système IoT environnemental repose sur une combinaison cohérente de capteurs, d'unités de traitement, de solutions d'alimentation et de réseaux de communication. Chaque composant joue un rôle essentiel dans la fiabilité, la précision et la durabilité du dispositif.

Cette image illustre la transmission des informations depuis les capteurs répartis sur le terrain, qui collectent des données environnementales (par exemple la qualité de l'air,

FIGURE 3 – Schéma du flux de données depuis le matériel (Hardware) jusqu'au Cloud.

température, humidité), jusqu'aux serveurs de traitement centralisés, via des réseaux de communication. Ces données sont stockées puis analysées afin de produire des visualisations, alertes et indicateurs destinés aux décideurs.

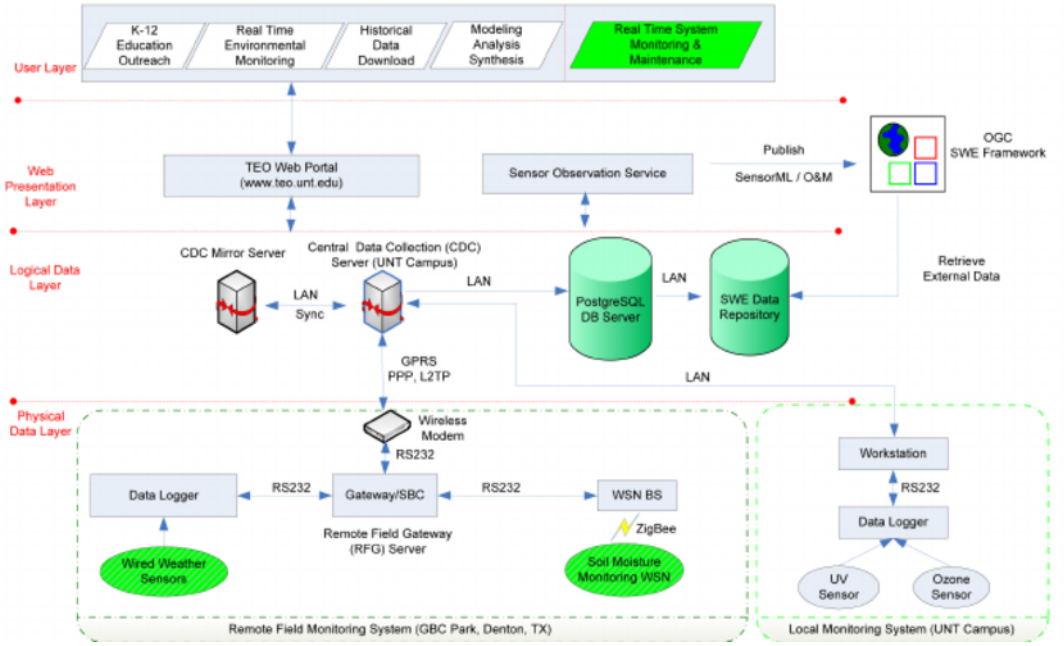

FIGURE 4 – Exemple d'architecture matérielle et logicielle multi-couches pour un système de surveillance environnementale.

3.1 Capteurs

Dans une architecture IoT environnementale, les capteurs représentent la couche de perception en permettant la conversion d'une grandeur physique comme la température, l'humidité, la pression atmosphérique ou les propriétés chimiques de l'eau, en données numériques pouvant être traitées par la suite sur des machines. Dans le modèle OSI, ils opèrent principalement au niveau de la couche physique, car interagissent directement avec l'environnement réel ; mais leur intégration implique aussi la couche liaison afin d'assurer la communication avec le microcontrôleur auquel ils sont connectés.

En pratique, on utilise différents types de capteurs en fonction des grandeurs à mesurer.

— Si l'on cherche des informations sur le climat local, on peut s'orienter vers des modules tels que le DHT22 ou le BME280 qui permettent la mesure de température et d'humidité.

— Il existe aussi d'autres dispositifs, comme le MQ-135 ou le CCS811, qui analysent la qualité de l'air grâce aux gaz détectés.

— Pour la pollution atmosphérique, il y a le SDS011 ou le PMS5003 qui sont des capteurs de particules fines (ils mesurent les concentrations de PM2.5 et PM10 dans l'air).

— Enfin, on peut se servir de sondes de pH, de turbidité ou de conductivité pour analyser l'état chimique et biologique de l'eau.

On peut aussi distinguer des capteurs actifs, qui émettent ou consomment leur propre énergie pour collecter l'information, et des capteurs passifs, qui mesurent simplement un signal ou un phénomène existant.

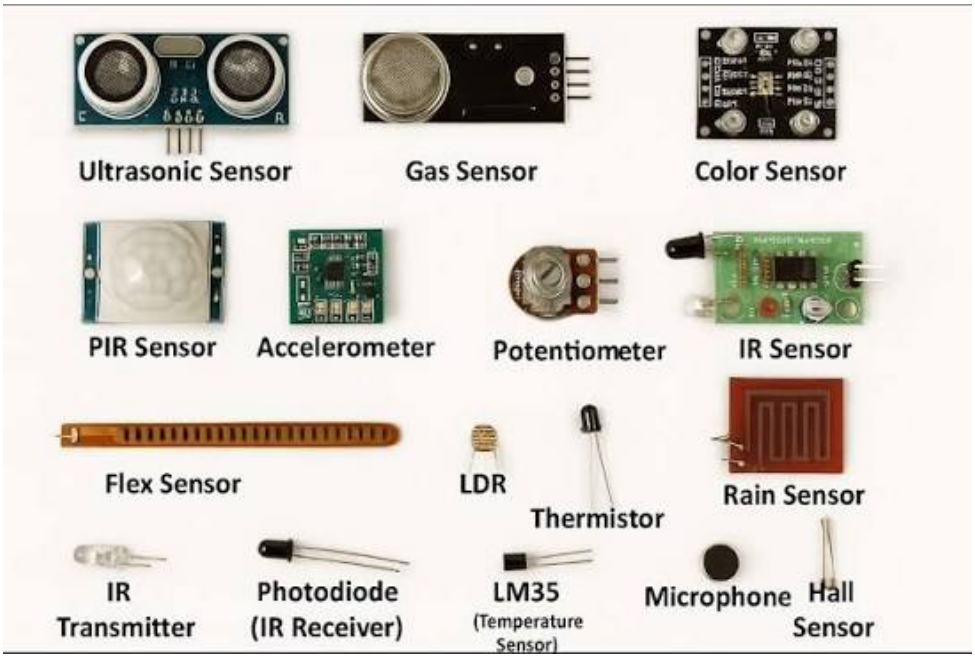

FIGURE 5 – Exemples de divers types de capteurs.

Le choix du capteur dépend directement du contexte de déploiement et impose un certain nombre de contraintes techniques.

— La précision attendue peut varier selon que l'objectif est d'obtenir une mesure indicative destinée à observer des tendances ou des valeurs destinées à des usages scientifiques nécessitant une rigueur de mesure accrue.

— La consommation énergétique est un facteur particulièrement déterminant, surtout dans le cas de déploiements autonomes en extérieur où l'alimentation repose sur

une batterie ou un panneau photovoltaïque, et où les dispositifs doivent souvent fonctionner en mode veille prolongée avec réveils intermittents.

— La robustesse mécanique et environnementale est également essentielle, les capteurs étant exposés aux intempéries, à l'humidité, à la poussière ou à des variations de température importantes.

— Enfin, l'interface de communication utilisée influence directement l'intégration au système : les protocoles I^2C, SPI, UART ou encore les sorties analogiques correspondent à la couche liaison du modèle OSI et conditionnent la manière dont la donnée brute sera transmise vers l'unité de traitement.

Une fois la mesure effectuée, l'information est encapsulée puis transmise dans la pile réseau. Selon l'architecture choisie, la transmission peut se faire via des technologies basse consommation et longue portée comme LoRaWAN ou Sigfox, adaptées aux environnements isolés, ou via des réseaux plus classiques tels que le Wi-Fi ou la 4G lorsque le débit et la continuité de service sont prioritaires. À ce stade, les couches réseau et transport du modèle OSI interviennent, assurant l'acheminement fiable des données jusqu'à la plateforme de traitement. Au niveau applicatif, des protocoles comme MQTT, conçu pour les communications légères entre machines, ou HTTP dans des solutions plus classiques, permettent l'exploitation directe des mesures pour la visualisation ou l'activation de mécanismes automatiques.

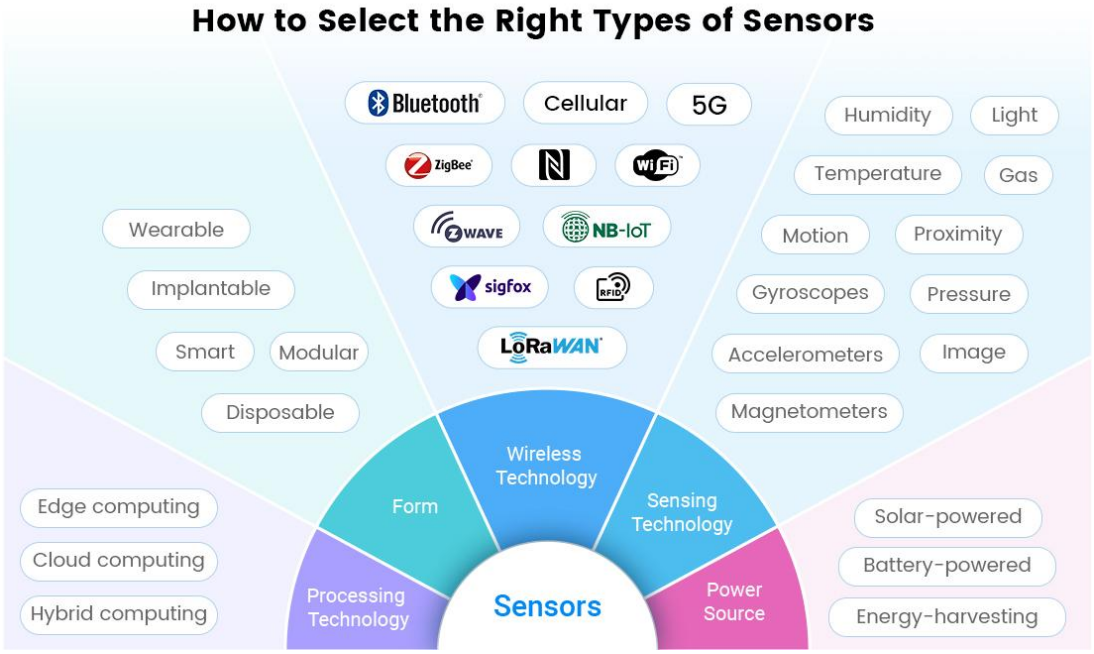

FIGURE 6 – Facteurs clés pour la sélection des capteurs : technologie de détection, alimentation, traitement, forme et technologie sans fil.

La fiabilité des capteurs et leur calibration régulière sont déterminantes pour garantir

la qualité des données transmises. Une dérive lente ou une défaillance d'un capteur peut entraîner des analyses incorrectes, voire des prises de décision inadaptées si les données alimentent des systèmes d'alerte automatique. C'est pourquoi la conception d'un système IoT environnemental nécessite une réflexion approfondie dès la sélection des capteurs, en tenant compte non seulement de leur précision et de leur coût, mais également de leur endurance, de leur stabilité dans le temps, et de leur compatibilité avec l'architecture réseau et logicielle du déploiement.

3.2 Unités de traitement

Les unités de traitement constituent le cœur d'un système IoT. Elles récupèrent les données provenant des capteurs, et effectuent tout le travail de traitement : opérations de filtrage, agrégation, détection d'événements... puis transmettent les informations traitées à l'infrastructure réseau ou cloud. Elles assurent donc le passage de la couche physique vers les couches réseau, transport et applicative du modèle OSI, en structurant les données de manière exploitable.

Pour exécuter les tâches simples, les microcontrôleurs tels que l'Arduino, l'ESP32 ou les STM32 sont très efficaces grâce à leur faible consommation énergétique. Cela les rend adaptés aux systèmes autonomes qui fonctionnent sur batterie. Ils interviennent lorsque le traitement demandé concerne des opérations légères, comme le calcul d'une moyenne, la comparaison avec un seuil ou l'envoi périodique d'une mesure vers une passerelle. Le choix d'un microcontrôleur dépend du nombre de ports GPIO recherchés (E/S), en fonction du nombre de capteurs et actionneurs à connecter ; mais il faut aussi prendre en compte la puissance de calcul et la mémoire disponibles, la tension d'alimentation, et la compatibilité avec les protocoles utilisés. Les communications avec les capteurs se font généralement via UART, SPI ou I^2C, correspondant aux couches physiques et liaison du modèle OSI. Les transmissions vers le réseau peuvent mobiliser LoRa, Wi-Fi ou Bluetooth selon l'architecture choisie.

Lorsque le traitement local doit être plus avancé, on se tourne vers des micro-ordinateurs tels que le Raspberry Pi ou le BeagleBone, capables d'exécuter un système d'exploitation complet basé sur Linux. Ceux-ci peuvent jouer le rôle de passerelles IoT en assurant la liaison entre les capteurs connectés par microcontrôleur et les serveurs distants. Grâce à leur capacité de calcul il est possible d'exécuter des algorithmes plus poussés, d'analyser des séries temporelles, ou encore gérer des bases de données locales. Ils interviennent dans une logique d'edge computing, comme vu précédemment, qui vise à rapprocher le traitement des données de leur source afin de réduire la latence, la dépendance au réseau et le volume d'informations à transmettre. Le choix entre microcontrôleur et micro-ordinateur dépend donc d'un choix à effectuer entre efficacité énergétique et puissance de traitement.

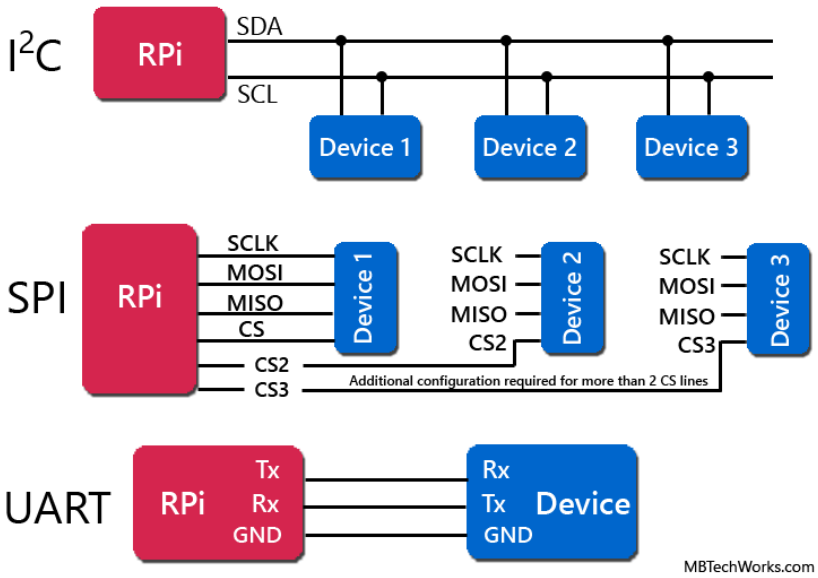

FIGURE 7 – Représentation schématique des protocoles de communication I²C, SPI et UART.

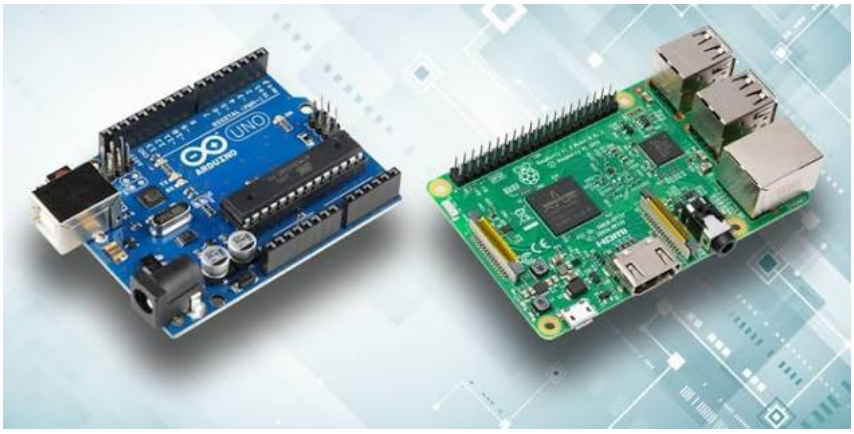

FIGURE 8 – Exemples d'unités de traitement : Microcontrôleur Arduino (à gauche) et Micro-ordinateur Raspberry Pi (à droite).

3.3 Alimentation et boîtiers

L'autonomie énergétique représente une contrainte majeure pour le déploiement de systèmes IoT dans des environnements isolés. Les dispositifs peuvent être alimentés par des batteries rechargeables, des piles longue durée ou par des panneaux photovoltaïques associés à des régulateurs de charge. Le système doit être optimisé pour réduire au maximum la consommation, par exemple grâce à des modes de veille profonde ou à des fréquences d'échantillonnage adaptées. Les boîtiers assurent la protection mécanique et environnementale des modules électroniques. Leur conception tient compte de l'exposition au vent, à l'humidité, à la poussière ou aux variations de température, et s'appuie généralement sur des certifications telles que IP65 ou IP67 pour garantir la durabilité du système sur

le terrain.

3.4 Réseaux de communication

La connectivité est le lien entre le monde physique (les capteurs) et le monde numérique (le cloud). Elle permet la transmission des données vers des serveurs d'analyse ou des plateformes IoT. Le choix du réseau de communication dépend de la portée, de la consommation et du volume de données :

3.4.1 Réseaux filaires

Les réseaux filaires représentent une catégorie de connectivité très employée pour sa robustesse et sa constance. L'Ethernet, via son interface physique RJ45, est la technologie la plus présente pour les applications fixes qui nécessitent à la fois une haute vitesse de transmission et une grande fiabilité. C'est une technologie qui est systématiquement déployée dans les infrastructures des bureaux, les centres de données et les systèmes de contrôle industriels car elle offre une connexion stable et sécurisée, essentielle pour les opérations critiques. Il existe aussi parallèlement des protocoles sériels tels que le RS-232 et le bus CAN qui restent utilisés dans certains domaines niches où leurs caractéristiques spécifiques sont indispensables. Le RS-232 reste une solution simple pour des liaisons point à point, tandis que le bus CAN est optimisé pour les communications en temps réel dans des systèmes embarqués complexes, notamment dans les domaines de l'automobile et de l'automatisation.

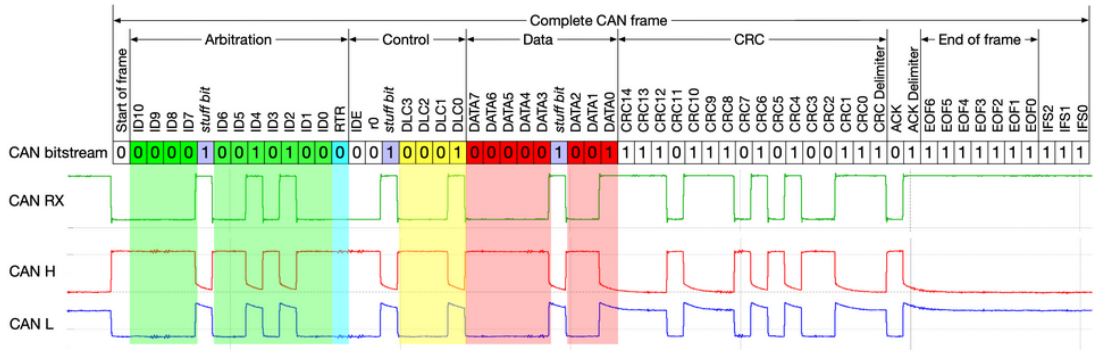

FIGURE 9 – Déroulement temporel d'une trame CAN avec ses champs logiques et les signaux physiques sur les lignes CAN H et CAN L.

Le schéma ci-dessus illustre le déroulement complet d'une trame CAN et la manière dont chaque champ logique se traduit physiquement sur les lignes CAN H et CAN L. La trame débute par le Start of Frame, un bit dominant qui sert de synchronisation pour tous les nœuds du réseau. Il y a ensuite le champ d'arbitrage, constitué de l'identifiant (ID0–ID10) et du bit RTR, qui permet de déterminer la priorité du message : un identifiant plus

dominant (donc comportant davantage de bits à 0) prend le bus en cas de concurrence. On y voit également des bits de bourrage (stuff bits) insérés automatiquement après cinq bits consécutifs identiques pour garantir la synchronisation du codage NRZ. Il y a ensuite le champ de contrôle, qui contient notamment le DLC (Data Length Code), qui indique le nombre d'octets de données qui vont suivre. Le champ de données, représenté en rouge dans le schéma, transporte l'information utile (qui peut être représentée jusqu'à 8 octets). Il y a ensuite le CRC permettant de vérifier l'intégrité de la trame à la réception. Le champ ACK matérialise la confirmation de la réception. Enfin, les bits End of Frame (EOF) marquent la fin de la trame et sont suivis d'un Interframe Space, un intervalle obligatoire permettant de séparer deux transmissions successives.

3.4.2 Réseaux sans fil

En contrepartie de la stabilité physique, les technologies de réseau sans fil offrent une flexibilité qui étend considérablement le champ des applications possibles. Le Wi-Fi par exemple se distingue par sa capacité à supporter une large bande passante, ce qui le rend adapté au transfert de données volumineuses comme les flux vidéo ou les mises à jour logicielles. Cependant, cette performance nécessite une consommation énergétique significative, ce qui limite son emploi pour les dispositifs dont l'autonomie sur batterie est une contrainte majeure comme les systèmes embarqués. Pour des portées plus faibles, il y a le Bluetooth et son extension maillée, Bluetooth Mesh, qui sont conçus pour interconnecter des appareils de proximité. Enfin, pour les projets où la portée et la faible consommation énergétique sont les priorités absolues, comme pour l'IoT, des protocoles comme Zigbee, LoRa ou Sigfox sont privilégiés. Ils permettent de déployer des réseaux de capteurs sur de vastes zones, y compris dans des endroits isolés, en assurant une longévité convenable pour les équipements sur le terrain.

Le choix du réseau dépend des besoins spécifiques du projet. Si la contrainte principale est la distance de transmission, il est préférable d'utiliser LoRa ou Sigfox, tandis que le Wi-Fi et Bluetooth sont plus adaptés pour du local. Si la bande passante doit être importante, on privilégiera Wi-Fi, sinon Sigfox. Enfin, si le dispositif doit respecter une faible consommation énergétique, on pourra utiliser Zigbee ou LoRa. L'interopérabilité entre ces protocoles reste un défi, mais les solutions hybrides (passerelles multi-protocoles) se développent rapidement. Pour relier ces éléments clés on utilisera une passerelle IoT (gateway). On peut prendre l'exemple d'une passerelle pour la surveillance de cultures agricoles : elle agrège les données issues de plusieurs capteurs, les traite localement, puis les envoie à une base de données distante. Cette approche réduit la consommation énergétique et améliore la fiabilité du système.

3.5 Vers le Cloud environnemental

Une fois collectées, les données sont transmises à un Cloud IoT où elles sont stockées, analysées et visualisées. Le matériel joue ici un rôle fondamental : il constitue la base de l'écosystème IoT, permettant de relier le monde physique (capteurs) au monde virtuel (intelligence artificielle, visualisation, décision). Les passerelles et micro-ordinateurs assurent la transmission vers le cloud, où les outils d'analyse, d'apprentissage automatique et de visualisation transforment ces données en connaissances utiles pour la préservation de l'environnement.

4 Composants logiciels (Software)

La couche logicielle constitue le cœur de l'intelligence d'un système IoT. C'est elle qui permet de collecter, transmettre, stocker, traiter et visualiser les données issues des capteurs, tout en garantissant leur sécurité et leur intégrité. Si la partie hardware constitue le corps d'un objet connecté, le software en représente le cerveau et le système nerveux. Il permet la collecte, la transmission, le traitement, la visualisation et la sécurisation des données issues des capteurs. Dans le cadre d'un projet d'objets connectés pour l'environnement, le software joue un rôle essentiel pour transformer les données brutes en informations exploitables, utiles à la surveillance, la prévention ou la gestion durable des ressources naturelles. Cette partie peut se décomposer en quatre grands volets.

4.1 Acquisition et transmission des données

La première étape d'un système IoT consiste à collecter et transmettre les données générées par les capteurs via les microcontrôleurs ou micro-ordinateurs. Les données issues des capteurs sont d'abord traitées localement par le microcontrôleur, puis envoyées à une plateforme distante via un protocole léger comme MQTT (Message Queuing Telemetry Transport) ou CoAP (Constrained Application Protocol). Ces protocoles sont conçus pour fonctionner sur des connexions instables ou à faible bande passante, ce qui les rend idéaux pour les environnements isolés. Des bibliothèques logicielles comme Arduino IoT Cloud, Node-RED ou ThingsBoard facilitent la configuration et la communication entre les objets. Pour assurer cette communication, il faut définir un protocole d'échange et une plateforme logicielle qui serviront d'intermédiaires entre les objets et les services cloud.

4.1.1 Choix de la plateforme IoT

Les plateformes IoT jouent un rôle central dans la gestion logicielle du système. Elles assurent la connexion, la supervision, la collecte et parfois le contrôle à distance des

dispositifs. Parmi les solutions les plus répandues, on trouve :

— **ThingsBoard** : open-source, adaptée aux projets de recherche et de prototypage. Elle offre des tableaux de bord interactifs et une gestion souple des règles d'automatisation.

— **Blynk** : simple à utiliser, orientée vers les projets éducatifs ou de petite échelle.

— **AWS IoT Core, Google Cloud IoT ou Azure IoT Hub** : plateformes professionnelles, hautement scalables, intégrant des outils d'analyse, de sécurité et de gestion de flottes d'appareils.

Ces plateformes facilitent la visualisation en temps réel, la configuration d'alarmes, et l'automatisation des actions en fonction des données environnementales (par exemple : déclencher une alerte si la température dépasse un seuil critique).

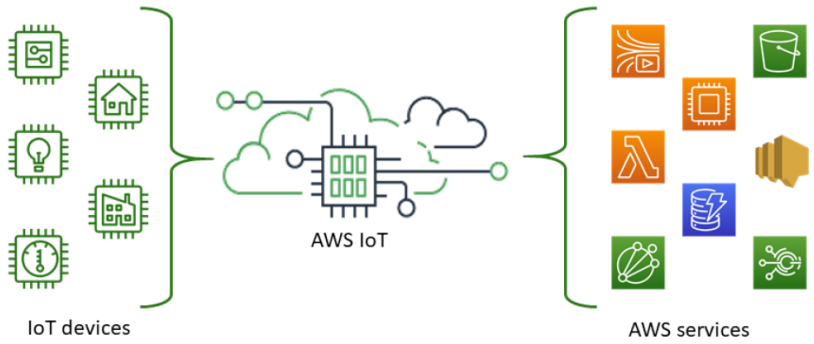

FIGURE 10 – Architecture d'un système IoT basé sur AWS IoT, connectant les dispositifs aux services AWS.

4.1.2 Protocoles de communication

Les protocoles déterminent comment les objets échangent les données à travers le réseau. Les plus utilisés dans les systèmes IoT sont :

— **MQTT (Message Queuing Telemetry Transport)** : un protocole léger et efficace pour les réseaux à faible bande passante. Il fonctionne selon un modèle *publish/subscribe*, idéal pour des capteurs environnementaux dispersés.

— **CoAP (Constrained Application Protocol)** : similaire à HTTP, mais optimisé pour les systèmes à ressources limitées. Il convient bien aux réseaux contraints comme LoRa ou ZigBee.

— **HTTP/REST** : plus standard et compatible avec le web, utile pour des communications simples avec des serveurs ou API, mais moins adapté aux environnements basse énergie.

Le choix du protocole dépend donc du type de réseau, de la fréquence d'envoi des données et des contraintes énergétiques du système.

4.2 Stockage et traitement : Cloud, Edge et bases de données

Une fois les données acquises, elles doivent être stockées et traitées pour en extraire des informations pertinentes. Deux approches principales coexistent : le cloud computing et le edge computing.

4.2.1 Cloud IoT

Le cloud permet de centraliser et de stocker d'importants volumes de données. Les solutions comme AWS IoT Core, Google Cloud IoT ou Microsoft Azure IoT Hub offrent des services de stockage, de traitement en temps réel et d'analyse prédictive grâce à des outils de machine learning intégrés. Ces plateformes permettent aussi d'établir une communication fluide entre les capteurs, les utilisateurs et les applications de gestion.

4.2.2 Edge computing

Dans certains cas, notamment pour les systèmes environnementaux isolés ou à connectivité limitée, une partie du traitement est effectuée en local, directement sur le microcontrôleur ou la passerelle (Raspberry Pi, ESP32). Cette approche réduit la latence et la consommation de bande passante, tout en garantissant la continuité du service même en cas de coupure réseau.

4.2.3 Bases de données

Les données collectées peuvent être stockées dans des bases de données locales ou distantes selon les besoins du projet :

— **InfluxDB** : spécialisée dans les séries temporelles, idéale pour suivre l'évolution de paramètres environnementaux (température, humidité, pression).

— **Firebase Realtime Database** : adaptée aux petits projets nécessitant une mise à jour instantanée des données.

— **MySQL / PostgreSQL** : bases relationnelles classiques, utilisées pour les projets nécessitant des analyses plus structurées.

4.3 Visualisation et exploitation : tableaux de bord et alertes

La visualisation permet de donner du sens aux données collectées. Grâce à des dashboards interactifs, les utilisateurs peuvent surveiller les paramètres environnementaux en

temps réel, analyser les tendances et détecter les anomalies. Les plateformes IoT comme ThingsBoard ou AWS IoT intègrent des outils de visualisation graphique (courbes, jauges, cartes géographiques) et permettent la configuration de tableaux de bord dynamiques affichant les mesures en direct, d'alertes et notifications envoyées par e-mail, SMS ou via une application mobile en cas de dépassement de seuil, ou d'automatisations, comme le déclenchement d'un actionneur (ex. : activer un système d'irrigation si le taux d'humidité est trop faible). Ces interfaces sont essentielles pour exploiter efficacement les données environnementales et faciliter la prise de décision.

4.4 Sécurité et confidentialité

La sécurité constitue un pilier essentiel dans tout système IoT, en particulier lorsque les dispositifs collectent des données environnementales critiques (qualité de l'air, température, pollution, etc.). Un projet IoT sans mesures de protection adaptées peut être vulnérable aux attaques informatiques, à la falsification des données ou à l'usurpation d'identité des appareils. C'est pourquoi il est indispensable de mettre en place une cybersécurité de bout en bout, depuis les capteurs jusqu'au cloud.

4.4.1 Sécurisation des communications avec TLS

L'un des mécanismes fondamentaux utilisés dans les architectures IoT modernes, notamment sur des plateformes comme AWS IoT Core, est le protocole TLS (Transport Layer Security). TLS est un protocole cryptographique qui permet d'assurer la confidentialité, l'intégrité et l'authenticité des échanges de données sur un réseau. Concrètement, AWS IoT Core impose que toutes les communications entre les objets connectés et la passerelle IoT (Device Gateway) soient chiffrées en transit à l'aide de TLS. Cela signifie que les données échangées via les protocoles MQTT, HTTP ou WebSocket sont protégées contre toute tentative d'interception ou de modification. Grâce à ce chiffrement, même si les paquets de données sont capturés par un tiers, ils restent illisibles sans la clé de déchiffrement. TLS offre trois niveaux de protection essentiels :

— **Confidentialité** : les messages sont chiffrés pour empêcher toute lecture non autorisée.

— **Intégrité** : toute altération des données est détectée.

— **Authentification** : les appareils et serveurs se reconnaissent grâce à des certificats numériques, garantissant que chaque élément du réseau est bien celui qu'il prétend être.

De plus, les données stockées dans le cloud sont également chiffrées par les services AWS eux-mêmes, afin d'assurer la sécurité complète, aussi bien en transit que au repos

(*data in transit / data at rest*).

5 Impact environnemental de l'IoT

Bien que l'IoT ait de nombreux bénéfices, il ne reste pas sans conséquence. Il nous semblait important au vu du contexte environnemental de cette étude d'inclure un passage sur les impacts environnementaux de l'IoT. On dénombre plusieurs effets négatifs de l'IoT, les principaux "hotspots" d'impact environnemental étant : la production des composants électroniques, énergivore et génératrice de CO ; l'utilisation continue, notamment la consommation énergétique des réseaux et serveurs ; la fin de vie des capteurs, souvent dotés de batteries non recyclables.

Cependant, il existe des stratégies d'atténuation : conception modulaire, récupération d'énergie (*solar harvesting*), et développement de matériaux plus durables. Ainsi, si l'IoT n'est pas totalement vert, ses apports en matière de réduction des gaspillages, suivi de la pollution et optimisation des ressources peuvent compenser ses effets négatifs, à condition d'adopter une démarche écoresponsable dans la conception des dispositifs.

L'Internet des Objets (IoT) s'impose aujourd'hui comme un levier central de la transition écologique. En connectant les équipements urbains, industriels ou domestiques à des plateformes d'analyse de données, il permet de surveiller, réguler et optimiser la consommation des ressources. Des réseaux d'eau aux systèmes d'éclairage public, en passant par la gestion énergétique des bâtiments, l'IoT aide à construire des villes plus durables. Mais cette révolution technologique a aussi un revers : fabrication énergivore, explosion du volume de données à traiter, dépendance aux centres de données... Cette étude propose un état des lieux global de l'usage de l'IoT dans la gestion environnementale et de son impact écologique réel.

5.1 L'IoT au service de la gestion environnementale

5.1.1 Surveillance environnementale et qualité de l'air

Selon France Environnement, la surveillance en temps réel des données environnementales est désormais incontournable pour évaluer la qualité de l'air et respecter les exigences européennes. Les méthodes traditionnelles (mesures ponctuelles) manquent de précision et de continuité. Les réseaux IoT, eux, reposent sur des capteurs connectés, des plateformes d'analyse et des protocoles de communication performants (ex : LoRaWAN en Seine-Saint-Denis). Cependant, les défis techniques persistent : fiabilité des capteurs, sécurité des données et consommation énergétique des réseaux. L'intégration du machine learning permettrait d'améliorer la précision des modèles et d'automatiser la détection

d'anomalies.

5.1.2 Gestion intelligente de l'eau : Veolia et Hubgrade

Veolia, via sa plateforme Hubgrade, optimise la gestion des ressources hydriques dans plusieurs villes à travers le monde. Les systèmes permettent de suivre la quantité et la qualité de l'eau potable, détecter les fuites à distance, et ajuster en temps réel la distribution selon les besoins.

Aux Émirats arabes unis, Hubgrade est utilisé pour gérer l'énergie, le trafic, les déchets et la qualité de l'air, tandis qu'à Barcelone, des capteurs régulent l'irrigation des parcs et fontaines pour éviter le gaspillage.

5.1.3 Éclairage public et villes intelligentes

D'après Requea, l'éclairage public représente jusqu'à 41% de la consommation électrique des communes. Grâce à l'IoT, les collectivités peuvent désormais adapter la luminosité selon la présence ou la luminosité ambiante, centraliser la supervision et anticiper la maintenance et réduire la pollution lumineuse.

Résultat : jusqu'à 65% d'économies d'énergie, une durée de vie des LED multipliée par 10 et une baisse nette des coûts d'exploitation. C'est un pilier fort des *smart cities*, soutenues par le Ministère de la Ville et du Logement.

5.1.4 Bâtiments intelligents et décret BACS

Entré en vigueur le 1er janvier 2025, le décret BACS impose aux bâtiments tertiaires (>290 kW de puissance thermique) d'intégrer des systèmes de gestion automatisés (GTB, capteurs, plateformes IoT). Les objectifs sont de réduire la consommation énergétique, d'améliorer la performance des bâtiments et de contribuer à la neutralité carbone 2050.

L'IoT permet un pilotage en temps réel de la consommation, tandis que la non-conformité expose les gestionnaires à des pertes financières et à une non-conformité réglementaire.

5.2 Les impacts environnementaux de l'IoT

5.2.1 Une empreinte carbone croissante

Selon l'étude ADEME x ARCEP (Mavana.earth), les objets connectés représentent environ 5% de l'empreinte environnementale du numérique français, dont 80% liés à leur fabrication. Malgré un impact unitaire faible, leur nombre explose (+21% en 2021), entraînant une pression accrue sur les ressources naturelles et les émissions carbone. L'empreinte

environnementale de l'IoT se divise en deux parties. Les appareils eux-mêmes : fabrication, transport, fin de vie ; et les données qu'ils génèrent : stockage, calcul, transfert.

5.2.2 Les données, un "géant énergétique invisible"

Comme le souligne Glooton (2025), les centres de données nécessaires au traitement des informations IoT pourraient représenter jusqu'à 10% de la consommation électrique mondiale d'ici 2025. Ces "cathédrales numériques" exigent un refroidissement constant et des infrastructures coûteuses en énergie. Les pistes de réduction sont l'usage d'énergies renouvelables pour les data centers, l'optimisation logicielle et la virtualisation et le recours à des architectures décentralisées (*edge computing*) limitant les transferts de données massifs.

5.3 Vers un IoT plus durable

Évidemment, des stratégies d'atténuation émergent : conception modulaire, récupération d'énergie (par exemple via le *solar harvesting*), et développement de matériaux plus durables.

Pour réduire l'impact global de l'IoT, l'ADEME recommande de ralentir la prolifération des équipements, de prolonger leur durée de vie, de favoriser les réseaux fixes plutôt que mobiles et de mesurer systématiquement les impacts environnementaux. Les entreprises, quant à elles, peuvent adopter des matériaux recyclés ou recyclables, concevoir des appareils réparables et modulaires et optimiser leurs algorithmes pour réduire la consommation énergétique.

Il y a aussi des technologies "IoT vert" qui émergent, comme les capteurs à basse consommation alimentés par énergie solaire ou par récupération d'énergie, les centres de données écoresponsables avec refroidissement passif, ainsi que l'intelligence artificielle utilisée pour ajuster dynamiquement la consommation des réseaux. Ces innovations, combinées à une prise de conscience collective, peuvent transformer l'IoT en véritable outil de durabilité.

6 Application expérimentale

Dans la continuité de notre étude théorique, nous avons conçu et déployé une architecture IoT fonctionnelle dédiée à la surveillance de paramètres environnementaux. L'objectif de cette expérimentation était de simuler une chaîne complète de traitement de l'information, de la mesure physique à la visualisation, en mettant en œuvre une infrastructure distribuée capable d'intégrer des capteurs hétérogènes.

6.1 Architecture du système

Pour répondre aux problématiques de flexibilité et de scalabilité évoquées dans l'état de l'art, nous avons opté pour une architecture distribuée s'approchant du modèle de l'Edge Computing. Contrairement à une approche centralisée classique où tous les capteurs seraient reliés à une unique unité de traitement, notre réseau se compose de trois nœuds distincts interconnectés via le protocole TCP/IP. Les deux premiers nœuds agissent comme des passerelles de collecte (*Gateways*), constituées chacune d'un microcontrôleur Arduino Uno relié à un ordinateur. Le troisième nœud fait office de serveur central. Cette structure permet d'ajouter de nouveaux points de mesure sans modifier l'architecture du serveur central, validant ainsi la modularité du système.

6.2 Implémentation matérielle (Hardware)

Le choix des composants s'est porté sur deux capteurs représentatifs des enjeux de surveillance environnementale. Pour le premier nœud, nous avons utilisé un capteur DHT11, permettant la mesure numérique de la température et de l'humidité relative. Le second nœud intègre un capteur de qualité de l'air Shinyei PPD42. Ce dispositif optique fonctionne sur le principe de la diffusion lumineuse pour détecter la concentration de particules fines en suspension. Son installation a nécessité une attention particulière, notamment un positionnement vertical strict pour garantir le flux d'air par convection thermique nécessaire à la mesure. Les microcontrôleurs Arduino assurent ici le rôle de perception en convertissant les signaux physiques en données numériques exploitables.

6.3 Implémentation logicielle (Software)

La communication entre les différentes couches du système repose sur une chaîne logicielle standardisée. Les données brutes sont d'abord formatées en objets JSON par les microcontrôleurs, ce format ayant été choisi pour faciliter l'interopérabilité entre les machines. Sur chaque passerelle, un script Python assure l'interface entre le port série et le réseau. Pour le transport des données, nous avons sélectionné le protocole MQTT (*Message Queuing Telemetry Transport*). Ce choix se justifie par sa légèreté et son modèle de publication/abonnement (*Publish/Subscribe*), particulièrement adapté aux contraintes des réseaux de capteurs. Enfin, la visualisation est assurée par le logiciel MQTT Explorer sur le serveur central, offrant une lecture graphique en temps réel des flux de données via des courbes dynamiques.

6.3.1 Microcontrôleurs (acquisition)

Le choix s'est porté sur des microcontrôleurs Arduino Uno pour l'acquisition des données des capteurs. Les programmes qui ont été flashés dans chaque Arduino sont similaires, la seule différence étant liée au type de capteurs que le microcontrôleur traite. Le programme pour la qualité de l'air est le suivant :

```
int pin = 8;
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 30000; // Mesure sur 30 secondes
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pin,INPUT);
  starttime = millis();
}

void loop() {
  duration = pulseIn(pin, LOW);
  lowpulseoccupancy = lowpulseoccupancy+duration;

  if ((millis()-starttime) > sampletime_ms) {
    ratio = lowpulseoccupancy/(sampletime_ms*10.0);
    concentration =
       1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62;

    // JSON: {"id": "air_quality", "val": 120.5, "unit":
       "pcs/0.01cf"}
    Serial.print("{\"id\": \"air_quality\", \"val\": ");
    Serial.print(concentration);
    Serial.println(", \"unit\": \"pcs/0.01cf\"}");

    lowpulseoccupancy = 0;
    starttime = millis();
  }
}
```

On commence par déclarer les variables nécessaires pour la suite. Puis, on débute une communication serial via le port USB et on définit le pin 8 comme étant une entrée. Ensuite, l'acquisition est effectuée en appliquant une formule de curve fitting générale pour ce type de capteur. Enfin, le résultat est formaté en JSON est transmis à l'ordinateur en USB via un print.

Le programme pour le capteur de température est le suivant :

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (!isnan(h) && !isnan(t)) {
    // JSON: {"id": "temperature", "val": 24.0, "unit": "C"}
    Serial.print("{\"id\": \"temperature\", \"val\": ");
    Serial.print(t);
    Serial.println(", \"unit\": \"C\"}");
  }
}
```

Pour ce programme, des bibliothèques existantes ont directement été utilisées. On initialise la communication serial ainsi que le capteur DHT. Ensuite, on lit toutes les 2 secondes la valeur de l'humidité et de la température. Enfin, si ces valeurs ne sont pas nulles, on les formate en JSON et on les transmet via USB. Il est à noter que dans le programme ci-dessus, la valeur de l'humidité n'est pas transmise car elle n'est pas indispensable pour la démonstration, mais il suffit d'ajouter un champ dans le JSON transmis, et de le lire côté serveur.

6.3.2 Passerelles

Pour la transmission des données acquises par tous les capteurs, nous avons opté pour un script Python tournant sur l'ordinateur hôte du microcontrôleur. Le programme est le suivant :

```
# Transmission
import serial
import json
import time
import paho.mqtt.client as mqtt

MQTT_SERVER_IP = "192.168.1.50"
TOPIC_PREFIX = "capteurs/"
```

```
9
10 SERIAL_PORT = "COM3"
11 BAUD_RATE = 9600
12
13 client = mqtt.Client()
14
15 def run():
16     print(f"Connexion au serveur MQTT {MQTT_SERVER_IP}...")
17     try:
18         client.connect(MQTT_SERVER_IP, 1883, 60)
19         client.loop_start()
20     except:
21         print("Erreur : Impossible de joindre le serveur MQTT")
22         return
23
24     print(f"Ouverture du port série {SERIAL_PORT}...")
25     try:
26         ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
27     except:
28         print("Erreur : Impossible d'ouvrir le port USB")
29         return
30
31     while True:
32         if ser.in_waiting > 0:
33             try:
34                 line = ser.readline().decode('utf-8').strip()
35                 if not line: continue
36
37                 data = json.loads(line)
38
39                 topic = TOPIC_PREFIX + data['id']
40
41                 client.publish(topic, json.dumps(data))
42                 print(f"Envoyé : {topic} -> {data}")
43
44             except json.JSONDecodeError:
45                 pass
46             except Exception as e:
47                 print(f"Erreur : {e}")
48         time.sleep(0.1)
```

```
49
50 if __name__ == "__main__":
51     run()
```

On commence par importer les librairies utiles : serial et json pour la lecture des données acquises par l'Arduino, time et mqtt pour la transmission vers le serveur central. On définit ensuite l'adresse IP du serveur cible et le port d'acquisition de l'Arduino sur l'ordinateur, puis le Baud Rate et le topic prefix qui est une sorte d'arborescence destinée au serveur central pour la structuration des réceptions. On tente ensuite de se connecter au serveur MQTT distant, puis en cas de réussite, au port série sur lequel est branché l'Arduino. Si toutes les connexions s'effectuent avec succès, il ne reste qu'à lire ce qui est print sur le port série, parser le JSON et le publier, ce qui va le transmettre au serveur central.

6.3.3 Serveur central

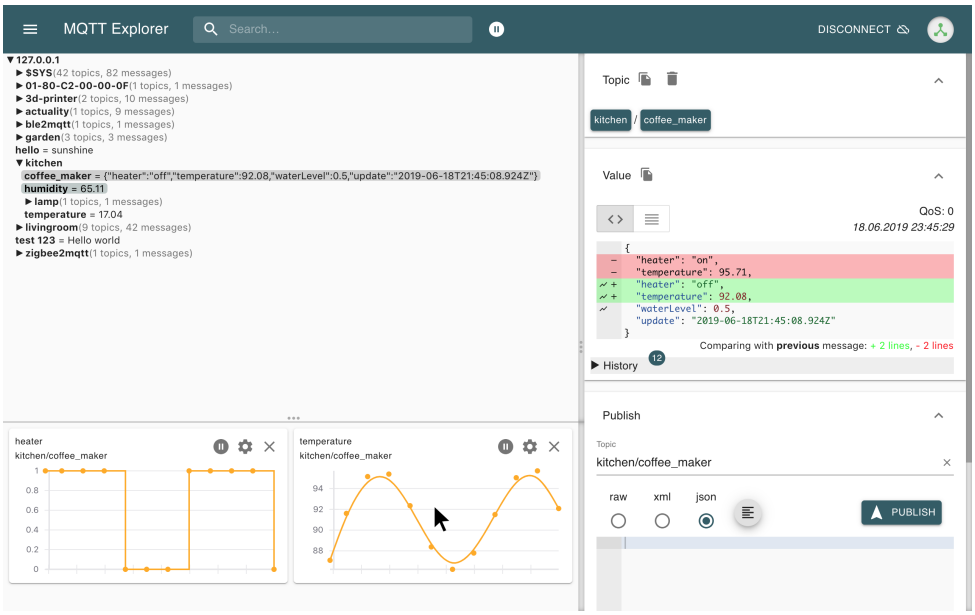

FIGURE 11 – Logiciel MQTT Explorer.

Pour le serveur central, il suffit d'exécuter le logiciel MQTT Explorer, ou tout autre programme permettant la réception des données via le protocole MQTT. La page principale montre une arborescence de dossiers. Dans notre cas, cela correspond à `/capteurs`, ce qui signifie qu'en pratique, il y aura un dossier capteurs qui une fois ouvert contient toutes les données transmises

6.4 Résultats et analyse

Les tests effectués en conditions réelles ont démontré la viabilité de l'architecture. La transmission des données s'est effectuée avec une latence négligeable, permettant un suivi en quasi-temps réel. Concernant la qualité de l'air, les mesures ont oscillé entre la valeur seuil minimale du capteur (0.62) et 781 pcs/0.01cf, témoignant d'un environnement intérieur sain lors de l'expérimentation. Pour valider la réactivité du système, nous avons simulé une source de pollution à proximité du capteur Shinyei, ce qui a entraîné une mise à jour immédiate des graphiques sur le poste de contrôle. Cette expérience confirme que l'architecture mise en place permet bien de croiser différentes sources d'informations pour obtenir une vision fiable de la situation environnementale. Aussi, notre installation est robuste et est déjà prévue pour accueillir une multitude de capteurs supplémentaires, il suffit de flasher un microcontrôleur avec le programme correspondant et d'exécuter le script python sur un ordinateur hôte pour le capteur.

Il y a également plusieurs voies d'amélioration. D'abord, il serait intéressant d'incorporer la réception MQTT directement dans un programme fait maison, qui pourra présenter les données dans un format sur-mesure et mettre en commun les données issues de différents capteurs. Par exemple, on pourrait tracer des courbes d'évolution, calculer la moyenne et d'autres informations statistiques sur les capteurs, ou bien garder en mémoire les données pour un suivi long terme de l'évolution des valeur. Aussi, il serait plus judicieux d'employer un capteur étant directement doté de capacités sans fil afin de pouvoir se passer de l'ordinateur hôte et du script python. Pour cela, on peut soit utiliser un ESP32 aux capacités sans fil et ajuster le programme C++ flashé, ou bien utiliser un Raspberry Pi Pico afin d'y exécuter directement le script Python. On pourra ainsi effectuer l'acquisition via les pins du Raspberry puis la lecture des valeurs en modifiant le script Python convenablement. Enfin, dans les 2 cas, on alimente l'ESP32 ou le Raspberry Pi via une batterie.

7 Conclusion

L'Internet des Objets appliqué à l'environnement illustre parfaitement le potentiel du numérique au service de la durabilité. Ce projet nous a permis de concrétiser les concepts théoriques en déployant une chaîne complète de valeur : de la couche de perception, assurée par les capteurs intelligents, jusqu'à la couche applicative de visualisation, en passant par un transport sécurisé et structuré des données. L'expérimentation a validé la pertinence d'une architecture distribuée pour la surveillance environnementale. Cette approche offre une grande souplesse et facilite l'intégration de technologies différentes, ce qui représente un atout majeur pour le déploiement de réseaux à grande échelle dans des contextes variés comme les villes intelligentes ou la surveillance agricole. La capacité de visualiser en temps réel des données invisibles à l'œil nu, comme la concentration de particules fines, confirme que l'IoT est un outil d'aide à la décision incontournable.

Cependant, cette mise en pratique a également mis en lumière les défis écologiques qu'impliquent ces technologies. L'utilisation de plusieurs ordinateurs pour gérer quelques capteurs, bien que nécessaire pour notre prototype, souligne le problème de la consommation énergétique des infrastructures numériques. Une évolution logique de ce projet vers un "IoT plus vert" consisterait à remplacer les passerelles PC par des microcontrôleurs autonomes à basse consommation (type ESP32) et à privilégier des protocoles radio longue portée comme LoRaWAN. Ainsi, l'équilibre entre la performance technologique et la sobriété énergétique reste l'enjeu central pour que ces dispositifs contribuent réellement à la transition écologique sans aggraver l'empreinte carbone numérique.

Bibliographie

Recherche et Études Techniques

Internet of Things (IoT) for Environmental Monitoring.
https ://www.researchgate.net/publication/382751058

Design of IoT Gateway for Crop Growth Environmental Monitoring Based on Edge-Computing Technology.
https ://pmc.ncbi.nlm.nih.gov/articles/PMC9303087/

A Standard-Based Internet of Things Platform and Data Flow Modeling for Smart Environmental Monitoring.
https ://www.mdpi.com/1424-8220/21/12/4228 ?

Comprehensive Design and Analysis of an IoT-Enabled Environmental Monitoring System for Industrial Applications.
https ://www.preprints.org/manuscript/202503.0392/v1

Généralités sur l'IoT et les Plateformes

What is AWS IoT ? - AWS IoT Core. (s. d.).
https ://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

Skandrani, Faker. « Architecture IoT : L'essentiel à savoir ». IoT Industriel Blog, 2 janvier 2022.
https ://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/

Georges. Device IoT : les 4+1 composants d'un objet connecté.
https ://www.matooma.com/fr/s-informer/actualites-iot-m2m/device-iot

M2M France. « Internet des objets (IoT) ». Dossier thématique, s.d.
https ://www.m2m.fr/iot/internet-des-objets/

Journal du Net. « IoT (Internet of things) : actualité, retours d'expérience et usages ».
https ://www.journaldunet.com/iot/1492599-iot/

« Fog Computing : Que signifie ? » IoT Industriel Blog. Consulté le 12 décembre 2025.
https ://iotindustriel.com/glossaire-iiot/fog-computing/

« Libelium» Connecting Sensors to the Cloud» IoT Solution Provider ». Libelium,
https ://www.libelium.com

Applications Spécifiques (Agriculture et Milieux Naturels)

Agriculture (Irrigation) : exemples d'irrigation intelligente et de gestion des cultures.

https ://www.mutualia.fr/nos-conseils/agriculture-et-sante/linternet-des-objets-iot-pour-une-agriculture-plus-intelligente

Milieux Naturels (Incendies) : projet "BurnMonitor" d'Inria pour la détection de feux de forêt.
https ://www.inria.fr/fr/burnmonitor-iot-detection-feux-foret

Smart City et Bâtiments Intelligents

Veolia. « Smart city : comment rendre la ville plus intelligente et durable ». Page web, s.d. (Inclut la gestion de l'eau).
https ://www.veolia.com/fr/ressources/smart-city.

Requea. « Décret BACS – Piloter la consommation des bâtiments grâce à l'IoT ». Article en ligne, s.d.
https ://www.requea.com/decrets-bacs-consommation-batiment-iot.html

Requea. « Gestion de l'éclairage public : l'IoT dans les smart cities ».
https ://www.requea.com/gestion-eclairage-public.html

Surveillance Environnementale Générale

Guide : tout savoir sur l'IoT environnemental.
https ://www.requea.com/iot-environnemental.html

L'IoT en faveur de l'environnement : 7 exemples concrets.
https ://objenious.com/blogpost/iot-au-service-de-lenvironnement/

France Environnement. « Optimisation de la gestion des données environnementales en temps réel via les réseaux IoT ». Article en ligne, s.d.
https ://www.franceenvironnement.com/article/optimisation-gestion-donnees-environnementales-temps-reel-via-reseaux-iot-400000183.

Impact Environnemental et Sobriété

Assessing the Environmental Impact of IoT Devices - Hotspots and Guidelines for a Better Understanding.
https ://www.researchgate.net/publication/394620124

Mavana. « Évaluation de l'impact environnemental de l'IoT en France ». Article en ligne, s.d. (ou article lié « Ce que dit le GIEC à propos des objets connectés »).
https ://mavana.earth/impact-iot-france/

Glooton. « L'empreinte carbone cachée de l'IoT ». Article en ligne, s.d.
https ://www.glooton.com/lempreinte-carbone-cachee-de-liot/