

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

**DÉVELOPPEMENT D'UNE PLATEFORME INTERNET
DES OBJETS À CONNECTIVITÉ HYBRIDE**

MÉMOIRE PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DU PROGRAMME DE MAÎTRISE EN SCIENCES ET TECHNOLOGIES DE
L'INFORMATION

PAR

HAMZA TAKROUNI

OCTOBRE 2020

Jury d'évaluation

Président du Jury : Dr. Halim Boutayeb

Membre du Jury : Dr. Péricles De Lima Sobreira

Directeur de recherche : Dr. Larbi Talbi

Mémoire accepté le : 30 septembre 2020

Dédicace

À Dieu Le Tout Miséricordieux, ton amour, ta miséricorde et tes grâces à mon égard m'ont fortifiée dans la persévérance et l'ardeur au travail.

À mon Père, en vous, je vois un père dévoué à sa famille. Votre présence en toute circonstance m'a maintes fois rappelé le sens de la responsabilité.

À ma mère, en vous, je vois la mère parfaite, toujours prête à se sacrifier pour le bonheur de ses enfants. Merci pour tout.

À ma regrettée tante Safia, ton amour et ta sollicitude à mon égard me marqueront à jamais.

À mon frère, pour qui, je le sais, ma réussite est très importante. Que Dieu vous paye pour tous vos bienfaits.

À ma femme Sabine pour son aide, encouragement, sa patience, sacrifices et amour.

À ma fille Sophia que j'aime tant.

À tous mes amis.

Remerciements

Tout d'abord je souhaite adresser mes remerciements à mon directeur de recherche. Professeur Talbi Larbi, pour m'avoir donné l'opportunité de bénéficier de cette expérience, et surtout pour ces conseils et le temps qu'il a consacré à la réalisation de mon projet malgré sa charge de travail.

Mes plus vifs remerciements vont également à toute l'équipe du laboratoire pour leur bonne humeur, leur gentillesse et leur accueil chaleureux.

Je tiens aussi à remercier Monsieur Fouzar Youcef pour son aide précieuse et ces conseils.

Finalement, il m'est aussi agréable de m'acquitter d'une dette de reconnaissance envers mon université l'UQO ainsi qu'envers tous les professeurs qui m'ont enseigné et formé.

Table des matières

Dédicace	iii
Remerciements	iv
Liste des figures.....	vii
Liste des tableaux.....	viii
Liste des abréviations, sigles et acronymes.....	ix
Résumé	xi
1. Introduction	13
1.1. Contexte.....	14
1.2. Problématique.....	15
2. Revue de littérature.....	17
2.1. Présentation de l’Internet des objets	17
2.1.1. Historique.....	18
2.1.2. Évolution.....	19
2.1.3. Apports.....	20
2.1.4. Application.....	20
2.2. Infrastructure de référence d’un système Internet des objets	22
2.2.1. Les protocoles de l’Internet des objets.....	24
2.2.2. L’informatique en périphérie et l’Internet des objets.....	29
2.2.3. L’informatique en nuage et l’Internet des objets	30
2.3. Travaux connexes	31
2.4. Contribution.....	32
3. Méthodologie de la recherche.....	34
3.1. Méthodologie adoptée	34
3.2. Approche de recherche	36
3.3. Reconceptiver et affiner l'expérience	37
4. Présentation de la solution proposée	38
4.1. Technologies et composants	39
4.1.1. Systèmes sur une puce (SoC).....	39
4.1.2. Arduino Uno	40

4.1.3.	Raspberry Pi 3 Model B.....	40
4.1.4.	ESP8266.....	41
4.1.5.	ESP32.....	42
4.1.6.	SIM5320A.....	43
4.1.7.	REYAX RYLR896	44
4.1.8.	Micrologiciel (Firmware).....	45
4.1.9.	Capteurs	46
4.1.10.	Serveur MQTT (Éclipse Mosquito)	46
4.1.11.	LAMP.....	46
4.1.12.	XAMPP.....	47
4.1.13.	Technologies Web : HTML, CSS, JavaScript, PHP, MySQL	47
4.2.	Conception de la solution	49
4.2.1.	Vue d'ensemble du circuit	49
4.2.2.	Configuration des serveurs.....	52
4.3.	Architecture de la solution.....	53
5.	Validation expérimentale.....	56
5.1.	Mise en place du circuit.....	56
5.2.	Évaluation du système	60
5.2.1.	Transmission de données LoRa	60
5.2.2.	Lancement des alertes	61
5.2.3.	Réception des données.....	63
5.2.4.	Interaction entre l'utilisateur et le système via les SMS.....	64
5.2.5.	Vue d'ensemble des fonctionnalités	64
5.3.	Résultats.....	66
5.3.1.	Surveillance et contrôle par SMS.....	66
5.3.2.	Surveillance et contrôle via l'interface Web	68
6.	Conclusion.....	72
	Bibliographie	74
	Annexe A.....	79
	Annexe B.....	86

Liste des figures

Figure 1. L'évolution de l'Internet des objets	20
Figure 2. Les domaines d'application de l'Internet des objets.....	21
Figure 3. Infrastructure de référence d'un système Internet des objets	22
Figure 4. L'informatique en périphérie et l'Internet des objets	29
Figure 5. L'informatique en nuage et l'Internet des objets.....	30
Figure 6. Les cycles de la méthodologie de recherche basée sur la conception	35
Figure 7. Les composants du circuit	49
Figure 8. Vue d'ensemble du circuit.....	50
Figure 9. Premier module du circuit	51
Figure 10. Deuxième module du circuit.....	51
Figure 11. Troisième module du circuit.....	52
Figure 12. Architecture client / serveur de la solution.....	52
Figure 13. Architecture générale de solution proposée.....	53
Figure 14. Vue réelle du premier module du circuit.....	56
Figure 15. Vue réelle du deuxième module du circuit.....	58
Figure 16. Vue réelle du troisième module du circuit.....	59
Figure 17. Contrôle du système par SMS	66
Figure 18. Alertes reçues par SMS	67
Figure 19. Rapports des états	68
Figure 20. Rapports des actions	68
Figure 21. Surveillance et contrôle en temps réel	69
Figure 22. Surveillance des alertes en temps réel	69
Figure 23. Contrôle des actions.....	70
Figure 24. Configuration du système.....	70

Liste des tableaux

Table 1. Tableau récapitulatif des protocoles de communication pour l'Internet des objets	25
Table 2. Avantages et inconvénients du protocole MQTT	28
Table 3. Caractéristiques de l'Arduino Uno.....	40
Table 4. Caractéristiques du Raspberry Pi 3 Model B	41
Table 5. Caractéristiques de l'ESP8266.....	42
Table 6. Caractéristiques de l'ESP32.....	42
Table 7. Caractéristiques du SIM5320A	44
Table 8. Caractéristiques du REYAX RYLR896.....	44
Table 9. Résultat expérimental pour la transmission de données LoRa.....	60
Table 10. Résultat expérimental pour le lancement des alertes	61
Table 11. Résultat expérimental pour la réception des données	63
Table 12. Résultat expérimental pour l'interaction avec l'utilisateur via les SMS	64
Table 13. Vue d'ensemble des fonctionnalités.....	64

Liste des abréviations, sigles et acronymes

Application Programming Interface	API
Bluetooth Low Energy	BLE
Central Processing Unit	CPU
Cascading Style Sheets	CSS
Enhanced Data rates for GSM Evolution	EDGE
Graphics Processing Unit	GPU
General Purpose Input/Output	GPIO
General Packet Radio Service	GPRS
Global Positioning System	GPS
Global System for Mobile Communications	GSM
High Speed Downlink Packet Access	HSDPA
Hypertext Transfer Protocol	HTTP
Hypertext Markup Language	HTML
Inter Integrated Circuit	I2C
Internet Protocol	IP
Internet des Objets	IdO
Internet Of Things	IoT
JavaScript Object Notation	JSON
Long Range	LoRa
Linux Apache MySQL PHP	LAMP

Message Queuing Telemetry Transport	MQTT
My Structured Query Language	MYSQL
Hypertext Preprocessor	PHP
Representational state transfer	REST
Random Access Memory	RAM
System On a Chip	SoC
Structured Query Language	SQL
Subscriber Identity Module	SIM
Transmission Control Protocol	TCP
User Datagram Protocol	UDP
Universal Mobile Telecommunications System	UMTS
Universal Asynchronous Receiver Transmitter	UART
X (cross) Apache MariaDB Perl PHP	XAMPP

Résumé

L'Internet des objets (IdO) concerne essentiellement la connexion des objets entre eux et à travers l'Internet permettant ainsi l'échange ou la transmission de données afin de bâtir une base de connaissance pour la prise de décision et l'exécution d'actions en temps réel. Le défi majeur pour de tels systèmes est le maintien de la connectivité sur le réseau Internet en tout temps. Advenant la perte de cette connexion, on devrait alors trouver une alternative, basée sur des technologies existantes, qui prendra le dessus (backup).

Considérant ce risque, nous proposerons dans cette mémoire une approche hybride combinant différentes technologies de connectivité pour assurer la disponibilité et le bon fonctionnement du système IdO et en garantissant la collecte sécuriser, le traitement et le stockage des données en cas d'événements imprévus.

Dans le but de démontrer la viabilité de notre solution dans un contexte réel, où plusieurs capteurs, actionneurs et microcontrôleurs sont déployés, la méthode proposée se concentre sur l'analyse, la conception et la réalisation d'un système IdO, intégrant différentes technologies de connectivité, à savoir, LoRa, Réseau cellulaire, Wifi et Bluetooth.

Les résultats obtenus indiquent la bonne performance de la solution proposée.

Mots clés : Internet des objets, informatique en nuage, informatique en périphérie, protocoles Internet des objets.

Abstract

The Internet of Things (IoT) essentially concerns the connection of objects to each other and across the Internet, thus allowing the exchange or transmission of data to build a knowledge base for decision-making and the execution of actions in real time. The major challenge for such systems is always to maintain connectivity on the Internet. In the event of the loss of this connection, we should then find an alternative, based on existing technologies, which will take over (backup).

Considering this risk, we will propose in this thesis a hybrid approach combining different connectivity technologies to ensure the availability and proper functioning of the IoT system and by guaranteeing the secure collection, processing and storage of data in the event of unforeseen events.

In order to demonstrate the viability of our solution in a real context, where several sensors, actuators and microcontrollers are deployed, the proposed method focuses on the analysis, design and realization of an IoT system, integrating different technologies of connectivity, namely, LoRa, Cellular network, Wi-Fi and Bluetooth.

The results obtained indicate the good performance of the proposed solution

Keywords: Internet of Things, Cloud Computing, Edge Computing, Internet of Things Protocols.

Chapitre 1

Introduction

L'évolution rapide des technologies de l'information et de la communication a fait d'Internet un élément essentiel de notre vie. En effet, mis à part d'être un moyen simple et rapide d'accéder à des informations abondantes en temps réel et disponibles dans le monde entier, Internet nous offre également une solution de communication solide et un moyen de connecter des personnes de partout dans le globe.

Au fur et à mesure de cette évolution, l'Internet n'est plus considéré comme une simple solution de communication, mais également utilisé pour connecter des éléments physiques impliquant une communication de machine à machine, d'où la nomination de l'Internet des objets.

L'Internet des objets (IdO/IoT) est un concept basé sur l'intégration de divers dispositifs et méthodes de communication pour échanger des informations afin de contribuer à l'amélioration de la vie quotidienne, des affaires et de l'environnement. Cet échange d'informations provenant de différentes sources dépend de la connectivité, ce qui rend possibles les flux de données nécessaires à des décisions intelligentes.

À mesure que l'IdO se profile, l'importance des systèmes de communication et de contrôle qui connectent les appareils IdO dans le but de fonctionner et de donner un sens à toutes les données générées a également augmenté. En effet, avec les appareils IdO connectés, une signalisation bidirectionnelle fiable est essentielle pour la collecte et le routage des données entre les appareils.

Pour cette raison, la mise en œuvre d'une stratégie efficace est nécessaire pour la connexion de composants à travers tous les emplacements sur la base de nombreuses technologies et réseaux différents, tels que le Wi-Fi, le réseau étendu à faible puissance ou la technologie cellulaire mobile pour assurer la continuité de service entre les types de connectivité ainsi que la cohérence du

contrôle et de la sécurité pour garantir que les données importantes ne sont pas perdues de manière permanente lors d'une panne.

1.1. Contexte

La base de l'IdO est de connecter des entités physiques sur le Web et de permettre l'extraction, l'analyse et le traitement des données pour conduire à des actions ou des prédictions.

Cette révolution a non seulement contribué à améliorer la vie, mais a également ajouté de la valeur aux particuliers et aux entreprises [1]. En effet, de nos jours, il existe de plus en plus de systèmes capables de surveiller l'environnement tel que la température, l'humidité et la pression et de collecter des informations sur leurs variations dans le temps. Par la suite, grâce à la transmission de ces données qui seront traitées et stockées, des prévisions peuvent être faites.

Un défi majeur pour de tels systèmes, bien que leurs applications couvrent différents domaines et ciblent différents clients, reste la dépendance à la connectivité [2]. En effet, les appareils peuvent communiquer avec un serveur pour collecter des données, ou le serveur peut communiquer avec des appareils, ou ces appareils communiquent entre eux. Quel que soit le cas d'utilisation, les données doivent se déplacer d'un point A à un point B de manière rapide et fiable. Par conséquent, il est de plus en plus impératif que ces systèmes soient protégés contre une menace très réelle telle qu'une perte d'alimentation qui peut entraîner une perte de connexion.

De tels problèmes peuvent perturber l'ensemble de la chaîne de communication, ce qui peut entraîner une perte de données potentielle pouvant entraîner des conclusions et des résultats erronés en raison de la non-conformité et du manque d'informations en temps réel, ce qui peut être un grave problème affectant la fiabilité des informations.

Pour faire face à un tel problème, il est important de proposer une solution complémentaire, voire alternative, qui puisse être utilisée comme sauvegarde en prenant en considération le besoin de base dans la majorité des systèmes IdO, qui est la nécessité de transmettre les données générées au nuage (Cloud) pour analyse, traitement et stockage [3] afin de donner suite à des informations utiles, des actions et des prédictions.

1.2. Problématique

Les données créées par un appareil IdO passent par trois étapes. La première étape est la création initiale, qui a lieu sur l'appareil, puis est envoyée sur Internet. La deuxième étape consiste à savoir comment le système central collecte et organise ces données. La troisième étape est l'utilisation continue de ces données pour l'avenir.

Pour les appareils intelligents et les capteurs, chaque événement peut créer et créera des données. Ces informations peuvent ensuite être envoyées sur le réseau à l'application centrale. Selon l'appareil, le réseau et les contraintes de consommation d'énergie, les données peuvent être envoyées en temps réel ou par lots à tout moment. Cependant, la valeur réelle est dérivée de l'ordre dans lequel les points de données sont créés.

En effet, des données chronologiques peuvent être créées à mesure que des événements se produisent autour de l'appareil, puis envoyées. Cette utilisation des informations en temps réel fournit un enregistrement complet pour chaque appareil, au fur et à mesure. Alternativement, il peut être collationné lorsque les données sont envoyées par lots, l'enregistrement des données historiques sera là, il n'est tout simplement pas disponible en temps réel. Chaque entrée doit être considérée comme reçue de l'appareil lui-même et placée dans la base de données. Une fois que vous avez ce magasin de séries de données chronologiques, la prochaine opportunité est de rechercher les tendances au fil du temps. L'analyse de séries des données chronologiques offre la possibilité de créer plus de valeur pour les propriétaires des appareils concernés ou d'effectuer des tâches automatisées selon un certain ensemble de conditions remplies. D'où l'importance que ces données chronologiques soient exactes pour les systèmes IdO. Sinon, cela risque de compromettre les objectifs mêmes des systèmes eux-mêmes.

Cependant, nous devons faire face à la réalité que de nombreux facteurs puissent provoquer une connexion sans fil intermittente ou interrompue, comme des routeurs qui ne fonctionnent pas, des interférences qui se produisent en cas de conflit avec certains appareils sans fil qui utilisent également la même fréquence dans le réseau, l'appareil qui s'éteint, une perte d'alimentation et des perturbations d'alimentation. De tels problèmes peuvent perturber toute la chaîne de communication, entraînant une perte potentielle de données et des interruptions de la production et de la continuité des activités. C'est pourquoi, pour s'assurer que des données importantes ne sont

pas perdues de manière permanente lors d'une panne, il sera nécessaire de repenser les stratégies de sauvegarde qui permettraient à ces systèmes de rester opérationnels pendant une période donnée tout en maintenant une transmission et une collecte d'informations continues.

Ces stratégies pourraient être basées sur une connexion réseau redondante. En effet, pour gérer une perte de connectivité Internet pendant quelques heures, une connexion redondante est une option. Mais une connexion redondante ne signifie pas seulement de se baser sur une même technologie ou des connexions sans fil et câblées si elles passent par le même fournisseur. Au lieu de cela, une connexion redondante utilise différentes technologies, fournisseurs et opérateurs de réseau. Cela signifie que même si un réseau rencontre des difficultés, l'autre réseau peut prendre le relais. Une autre stratégie consiste à utiliser une connexion d'alimentation de secours. Une panne de courant inattendue pourrait entraîner la perte d'une énorme quantité de données. Même s'il ne faut que quelques secondes aux générateurs pour se mettre en ligne, cette brève interruption pourrait entraîner la perte de centaines de données. Ce scénario peut être facilement évité en utilisant une alimentation alternative qui fournit une alimentation de secours lorsque l'alimentation régulière est coupée.

Dans un scénario idéal, la meilleure solution est d'adopter plusieurs stratégies pour garantir la continuité et la cohérence des données. Dans notre solution, nous aborderons et mettrons l'accent sur une stratégie où plusieurs technologies de connectivité seront utilisées avec un système de sauvegarde pour le stockage des données.

Chapitre 2

Revue de littérature

Les recherches et développements continuent activement sur le concept d'Internet des objets. Plusieurs applications ont vu le jour dans différents domaines et plusieurs autres sont en cours d'élaboration. Les études et les recherches montrent des limitations et des faiblesses qui sont encore présentes cependant, ça n'empêche pas la demande croissante pour de tels systèmes.

Dans ce chapitre, nous allons aborder quelques notions et concepts fondamentaux pour la suite de notre travail. Tout d'abord, nous commencerons par présenter l'Internet des objets. Par la suite nous passerons à la description d'une infrastructure typique d'un système Internet des objets et l'introduction de certaines ressources typiques et principales technologies. Nous nous limiterons aux technologies qui seront impliquées dans notre travail. Puis, un petit résumé des articles qui abordent les systèmes de suivi et contrôle Internet des objets. Pour finir, nous présenterons la contribution de notre mémoire.

2.1. Présentation de l'Internet des objets

Pour l'Internet des objets, on ne trouve pas une définition exacte du concept, cependant, elle est présentée pour décrire la mise en place de systèmes et des solutions technologiques impliquant la mise en réseau d'objets physique doté de capacités différentes et qui partagent un mode de communication pour permettre la collecte, l'échange et le transfert de données provenant de différents capteurs permettant ainsi l'interaction avec l'environnement externe [4].

Ces données caractérisées par leurs volume, variété et vitesse sont utilisées pour améliorer les processus en les automatisant ou bien en tirant de l'information constructive et utile qui peut être la base d'action en temps réel ou dans le futur et de sorte à offrir des niveaux de services avancés et par conséquent changer pratiquement la façon de faire.

Les acteurs présents dans de tels systèmes peuvent être physique tels qu'appareils électroniques et capteurs ou bien virtuels tels que protocoles et technologies de transmission des signaux qui permettant la génération et la transmission des données qui sont la base d'information.

Ces données collectées, traitées et stockées puis utilisées pour le suivi et le contrôle dans différents cas d'utilisations permettront des progrès dans différents domaines tels que la médecine, l'énergie, l'agriculture, les villes intelligentes et les maisons intelligentes [5].

2.1.1. Historique

Le terme Internet des objets a été utilisé en 2009 par Kevin Ashton [6] pour décrire l'interconnexion des périphériques physiques sur l'Internet pour échanger les données ou être contrôlé. Cependant, depuis 1999, ce même terme a été présent plusieurs fois avec des différentes interprétations dans différents articles de recherches ce qui a introduit de différentes présentations du concept.

En effet, pour les entreprises l'Internet des objets représente un moyen et un outil commercial très important pour l'amélioration et l'efficacité de la production et l'innovation.

Pour Cisco, le concept est bâti sur l'interconnexion des objets à Internet dans le but d'amélioration du quotidien. Ils considèrent l'Internet des objets comme une phase qui s'achèvera dès que tous les objets seront connectés et se transformera en Internet de toutes les choses [7].

Pour IBM, ils insistent sur l'interconnexion des systèmes dans leur définition que sur les objets dans le but de créer un système des systèmes et définissent l'Internet des objets comme un moyen pour créer une planète plus intelligente [8].

Pour les chercheurs, l'IdO a été introduit par le Dr John Barrett, responsable des études académiques pour la recherche sur les systèmes embarqués au Cork Institute of Technology [9] dans un exposé TEDx sur la nécessité de l'Internet des objets, il l'a défini comme un concept basé sur un identifiant unique, communication et contrôle avec une capacité d'autoapprentissage et prise de décision grâce aux données collectées.

Pour ce qui est de Kevin Ashton l'Internet des objets est une solution fondamentale pour les problèmes futurs et définit comme dès la mise en place de machines détectant des éléments de l'environnement qui les entourent de façon autonome et mettant à disponibilité ces informations

via l'Internet. La problématique pour lui n'ont pas le déploiement des objets connectés, mais plutôt de bâtir des systèmes capables d'exploiter les données mises à disposition et extraire de l'information utile de façon automatique.

Malgré la diversité les divergences dans la description et présentation de l'Internet des objets, néanmoins des éléments communs sont présentent dans toutes les définitions. En effet, l'interconnexion des objets est un point présent dans toutes les descriptions et présente le cœur même de tel système pour permettre la transmission et l'échange des données. Les quantités immenses de données générées, la collecte des données, le traitement et l'extraction d'informations utiles pour la prise de décision ainsi que l'automatisation des processus sont les autres aspects communs faisant partie des différentes définitions.

2.1.2. Évolution

L'une des bases du concept de l'Internet des objets était disponible depuis plusieurs années. En effet, la collecte des données à partir des capteurs et leurs stockages dans un emplacement centralisé existaient avant même de l'introduction de l'IdO [10].

Cependant, l'évolution, l'amélioration et l'émergence des nouvelles technologies et systèmes extrêmement puissants avec une diminution des coûts d'appareil et des services [11] ont donné un autre élan à l'évolution et l'apparition de l'Internet des objets.

En effet, le monde est en train de vivre une aire où les coûts de stockage des données sont en diminution continue, les capteurs sont de plus en plus de taille extrêmement minime avec des précisions de haut niveau. En plus, l'apparition et l'évolution des technologies de traitement de données et apprentissage machine [12] ont eux aussi favorisés l'évolution de l'IdO qui continue avec le temps.

La popularité des systèmes Internet des objets est de plus en plus sentie et leur mis en place ne cesse d'augmenter [13]. En effet, en 2011 Cisco IBSG [14] a estimé que 50 milliards d'appareils seront connectés à Internet de 2011 à 2020 (objets physiques).

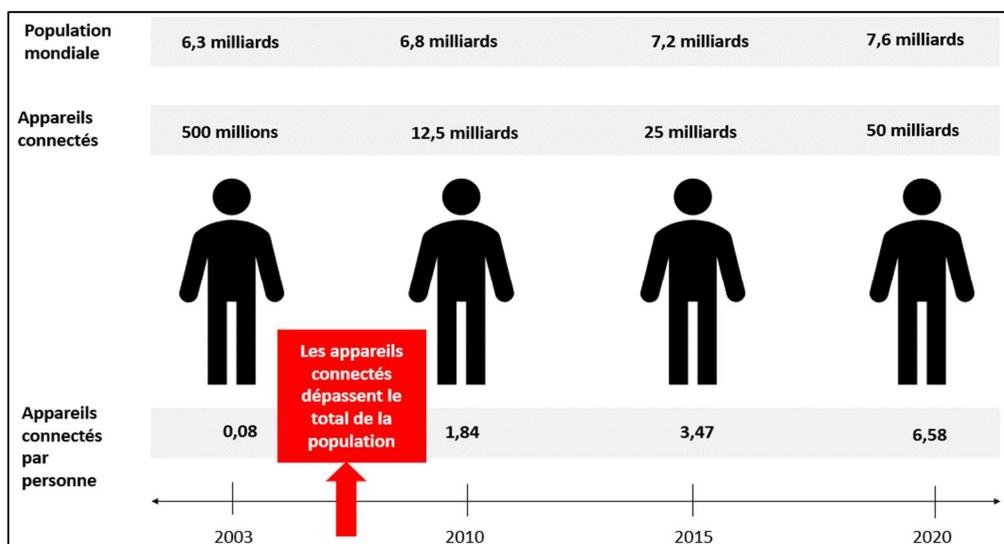


Figure 1. L'évolution de l'Internet des objets [14]

2.1.3. Apports

L'évolution, les avancements et les découvertes technologiques qui ont abouti à des systèmes mis en place ont toujours été alimentés par le besoin des personnes et dans le but de faciliter la vie de ces derniers ainsi que présenter une alternative aux limites des êtres humains pour bâtir une société durable.

C'est dans ce contexte que l'Internet des objets vient pour faire face aux lacunes et limitation dans le processus de collecte, traitement et transmission des données [15] tout en gardant un niveau acceptable d'exactitude, de fiabilité et de pertinence.

En effet, Il y a de nombreuses années, ces données numériques été dactylographiées, enregistrées ou créées par des personnes ce qui imposait une limitation étant donné que la vitesse de traitement des données par les êtres humains est inférieure comparé aux machines qui en plus permettent d'automatiser le processus et ainsi accroître la vitesse, augmenté la quantité de données, réduire les pertes et les coûts inutiles avec une grande amélioration de la précision [16].

2.1.4. Application

Les contextes de l'utilisation de système d'Internet des objets peuvent différer pour plusieurs domaines allant du corps humain, maison et ville jusqu'aux industries et l'environnement.

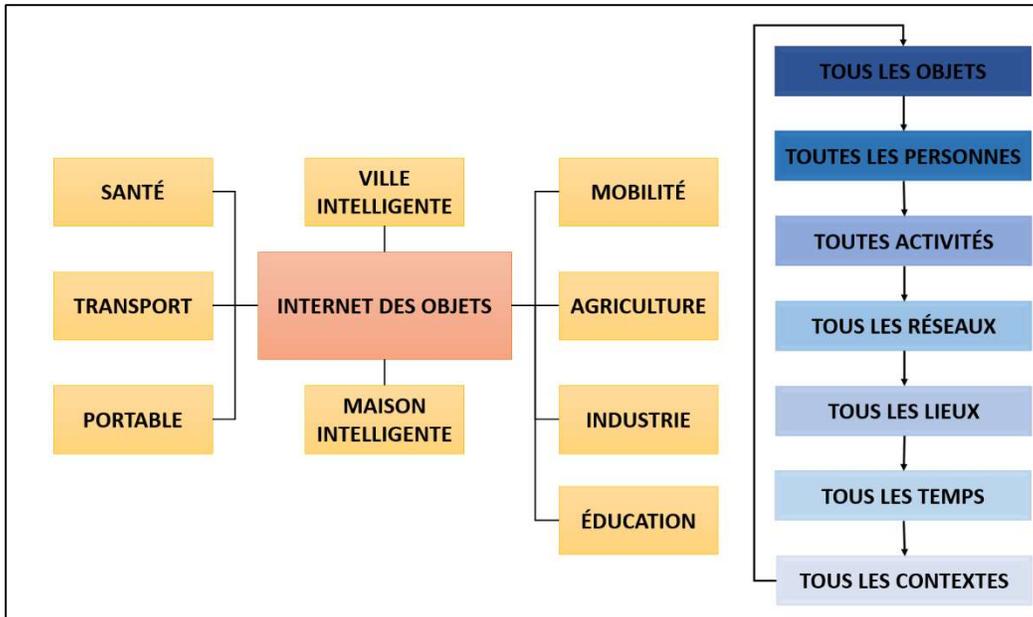


Figure 2. Les domaines d'application de l'Internet des objets

En effet, l'implication de l'Internet des objets dans des systèmes sont utilisés pour l'amélioration de la vie humaine par exemple un système de détection et suivi de l'état de santé des personnes, en collectant des informations sur leurs habitudes qui par la suite sont traitées, stockées puis utilisées pour identifier les mauvaises habitudes et les risques afin de prédire des problèmes de santé et ainsi prendre des mesures et les actions nécessaires [18].

Dans d'autres cas tels que les maisons intelligentes [19], l'Internet des objets peut être utilisé dans des systèmes de surveillance, le suivi et contrôle de différents appareils électroniques ou la commande à distance tel que le déclenchement du système d'arrosage des plantes.

Autre situation, pour les villes intelligentes [20], l'Internet des objets peut être un outil de collecte, traitement et stockage et un moyen d'avoir une source d'informations sur les infrastructures pour permettre aux municipalités ou organismes responsables d'offrir des services d'amélioration de la vie des citoyens ou en améliorant les interventions par exemple pour la collecte des déchets, pour qu'ils puissent être ciblés selon le besoin et ainsi optimiser les interventions et diminuer les coûts et les frais de déplacement.

En plus pour l'industrie [21], l'Internet des objets permet le suivi des actifs des entreprises, améliorer la sécurité environnementale, maintenir la qualité et la cohérence du processus de production. Ce qui entraîne une amélioration de la productivité et la réduction des

coûts inutiles tels que pour la maintenance des systèmes qui sera plus ciblée et selon le besoin grâce aux suivis des états.

Pour finir, l'impact de l'Internet des objets est non négligeable étant donné que cette dernière permet d'appréhender les changements et mieux gérer les ressources. En effet, les capteurs peuvent aider à protéger la faune, à suivre l'utilisation et les débits d'eau, à surveiller les conditions météorologiques locales, à surveiller l'utilisation des ressources naturelles ou à donner des avertissements avant et après les catastrophes naturelles afin de préparer les populations à l'avenir.

2.2. Infrastructure de référence d'un système Internet des objets

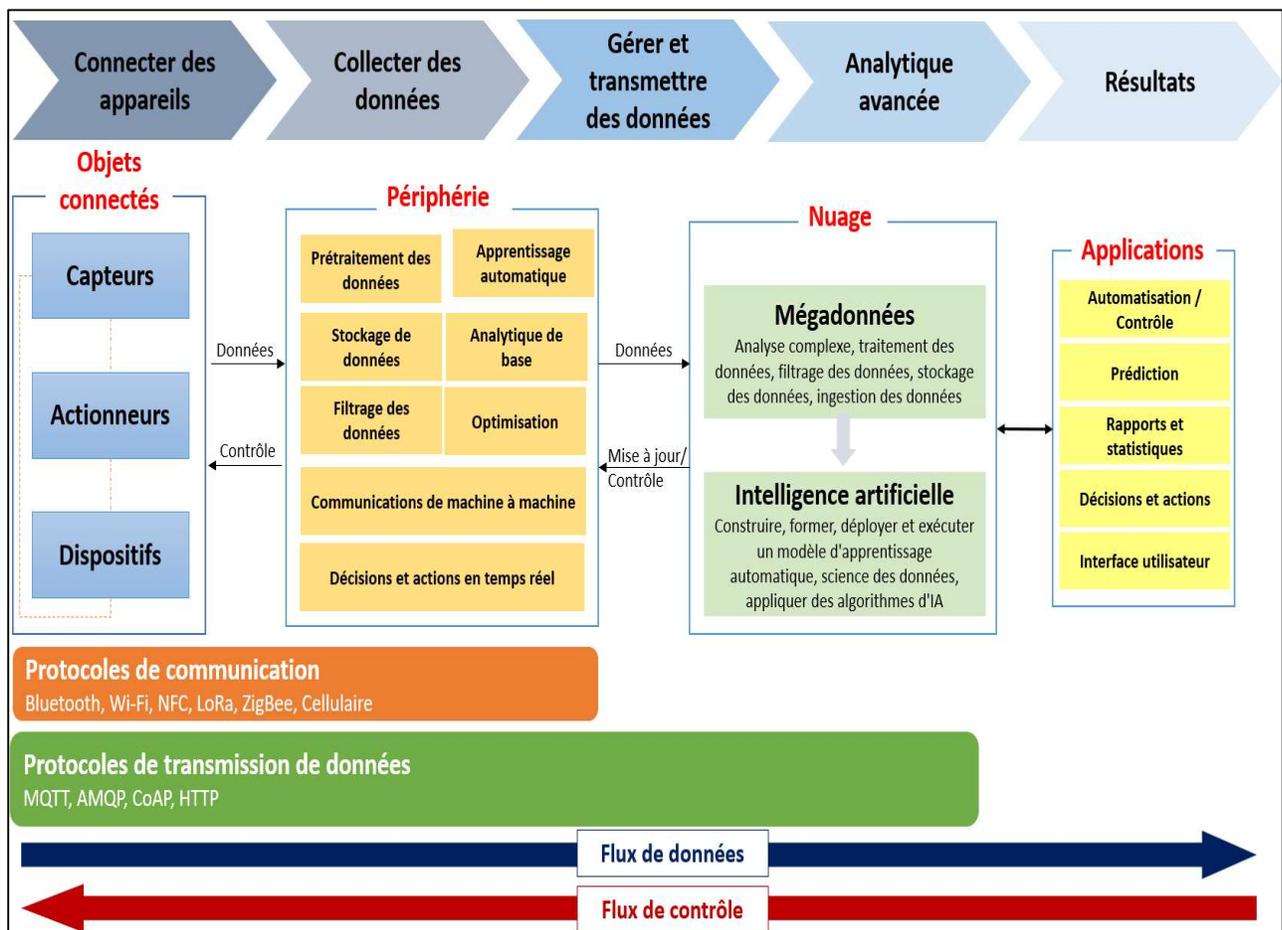


Figure 3. Infrastructure de référence d'un système Internet des objets

Nous remarquons à partir de la figure 3 que l'infrastructure typique d'un système IdO peut être séparés en quatre modules principaux. Ces modules sont les générateurs de données et

exécuteur d'action (objets connectés), la connectivité et le support de transport de données (protocole de communication et transmission de données), le traitement et le stockage (informatique en périphérie ou dans le nuage) et enfin les applications [21, 22, 23, 24].

Tout d'abord, les objets qui sont les capteurs, actionneurs ou tous autres dispositifs qui jouent le rôle d'entrée en alimentant le système par des données collectées sur leurs environnements tels que des capteurs de température ou les caméras vidéo et les actionneurs qui acceptent des commandes pour exécuter une fonction telles qu'un moteur.

Ensuite, les données collectées doivent être transmises, d'où le deuxième module qui est la connectivité qui va s'assurer du transfert des données dans le nuage (Cloud) ou autres pour leur traitement. On trouve plusieurs formes de technologies capables d'effectuer les transferts de données tels que les réseaux cellulaires, le Wi-Fi ou la connexion directe via les câbles Ethernet et des protocoles IdO tel que le MQTT.

Suivit de la collecte et le transfert de données qui sont caractérisées par leurs quantités qui est immenses et leurs diversités, une étape importante et qui est le troisième module qui comprend l'informatique en périphérie (Edge Computing) et l'informatique en nuage (Cloud Computing) pour le traitement des données qui est une tâche très importante et très complexe. En effet, étant donné que le but majeur de cette collecte de données est l'extraction d'informations pertinentes, il est primordial de procéder au traitement de ces données qui doivent dans certains cas, donner suite à des actions qui doivent être prises en temps réel, tel que dans les systèmes de suivi des températures où le dépassement de seuil d'une température donnée doit enclencher automatiquement des actions incluant des alertes.

Pour finir, le quatrième module qui est les applications ou les interfaces utilisateur. Étant donné qu'au final, bien que nous tendions à automatiser le plus possible les systèmes, il est important de permettre aux utilisateurs d'effectuer le suivi, avoir des comptes rendus, des résultats et des rapports par le biais d'interfaces graphiques telles que des applications Web ou mobiles. En plus, de permettre l'accès et le suivi il est nécessaire de permettre à l'utilisateur d'avoir la possibilité de contrôler et effectuer des actions.

2.2.1. Les protocoles de l'Internet des objets

La communication est l'élément essentiel de la mise en place du système IdO permettant l'interconnexion de différents objets puisque l'idée derrière le concept est que tout doit être connecté. En effet, en l'absence de tels composants, le système ne peut pas être configuré et la collecte, la détection, le stockage et le traitement deviennent inutiles. C'est pourquoi la sélection du bon protocole de connectivité est la clé du succès du projet [25]. Il est donc important d'analyser les exigences du système, telles que les intervalles de transmission de données, le flux de données, la sécurité et la bande passante [26]. Ces protocoles pour l'IdO pourraient être divisés en deux groupes. Protocoles de communication tels que Bluetooth, Wi-Fi, LoRa et protocoles de transmission de données telles que MQTT [26, 27].

La mise en place d'un système de surveillance et contrôle intelligent peut être un défi avec l'évolution des normes de connectivité sans fil pertinentes pour les systèmes IdO. Nous présenterons dans cette section une partie des protocoles de communications les plus répandues et qui seront utilisés dans notre solution.

Wi-Fi

Le Wi-Fi est une technologie de transmission sans fil haute vitesse qui utilise les ondes radio pour permettre à deux appareils de communiquer entre eux. Cette technologie est la technologie la plus utilisée et est couramment utilisée pour connecter des routeurs à des ordinateurs, des tablettes et des smartphones et pour connecter des objets.

Cette technologie utilise une large bande passante et nécessite une grande quantité d'énergie. Elle consomme beaucoup d'énergie et les composants nécessaires sont chers, ce qui signifie que la plupart des concepteurs d'objets connectés évitent cette technologie.

Bluetooth

Le Bluetooth est une technologie utilisée pour envoyer des informations sur de courtes distances avec l'établissement d'une connexion de proximité sécurisée, principalement utilisée sur les appareils connectés aux téléphones et tablettes. Il consomme moins d'énergie que le Wi-Fi et est pris en charge par la plupart des smartphones, ordinateurs et systèmes d'exploitation. Cela lui

confère un avantage notable dans le contexte des objets connectés compte tenu de son intégration généralisée dans les smartphones et autres appareils mobiles.

Le problème avec cette technologie est que les objets connectés doivent être proche, à une distance réduite et ne permet pas de transférer des fichiers volumineux.

LoRa

LoRa est un protocole à bas débit qui permet le transfert de données sur des distances de 2 à 5 kilomètres en zone urbaine et jusqu'à 15 kilomètres en zone rurale. LoRa cible un large éventail d'applications et est conçue pour offrir une connexion à faible puissance avec des caractéristiques favorables pour prendre en charge la communication mobile bidirectionnelle sécurisée pour les objets connectés et la communication machine à machine. LoRa est optimisée pour une faible consommation électrique et prend en charge les grands réseaux avec des millions d'appareils et de données.

Ce réseau est intéressant pour transférer une petite quantité de données, mais ne peut transiter que de 0,3 à 0,5 kilobit par seconde.

Cellulaire

Le principal avantage des réseaux cellulaires est qu'ils permettent le transfert de grandes quantités de données avec une possibilité de transfert des données à travers l'Internet.

Néanmoins, reste son coût qui est très élevé étant donné la dépendance aux fournisseurs de service et la disponibilité du réseau dans des emplacements éloignés. En plus de ça, la grande consommation d'énergie.

Table 1. Tableau récapitulatif des protocoles de communication pour l'Internet des objets

	Norme (la fréquence)	Portée	Vitesse de transmission	Avantages	Inconvénients
Bluetooth	Bluetooth 4.2 (2,4 GHz)	20 m	1 Mbit/s	- Grande capacité à diffuser de grandes quantités de données.	- Courte portée. - Petites portions de données.
Wi-Fi	802.11n (2,4 / 5 GHz)	50 m	600 Mbit/s	- Bonne qualité de signal. - Haute vitesse.	- Grande consommation d'énergie.

Cellulaire	GSM/GPRS/EDGE (2G)	35 km for GSM	35-170 kbit/s (GPRS)	- Haute sécurité. - Bonne qualité de signal.	- Faible portée dans les zones rurales
	UMTS/HSPA (3G)	200 km for HSPA	120-384 kbit/s (EDGE) 384 kbit/s-2 Mbit/s (UMTS)	- La haute disponibilité. - Bonne couverture géographique.	- Dépendance opérateur. - Grande consommation d'énergie.
	LTE (4G) (900 / 1 800 / 1 900 / 2 100 MHz)		600 kbit/s-10 Mbit/s (HSPA) 3-10 Mbit/s (LTE)	- (4G) Haut débit.	- Trop cher.
LoRa	LoRa (Variable)	15 km (rural) 2-5 km (urbain)	0,3-50 kbit/s	- Faible consommation d'énergie. - Bas prix. - Longue portée.	- Faible vitesse de transmission.

Le tableau 1 présente les caractéristiques des différentes normes sans fil qui peuvent être appliquées dans un système IdO. L'émergence et la diversité des protocoles de communication nous donnent un choix plus large pour le type de connectivité, élément fondamental de tout projet IdO, adopté pour la transmission et la réception des données.

Cependant, chaque technologie a ses avantages et ses inconvénients et peut être efficace dans un contexte et moins dans un autre. En effet, chaque technologie offre différentes capacités et qualités pour répondre aux objectifs du système à mettre en place. Par conséquent, il est important de prendre en considération les besoins des systèmes IdO tels que la transmission de données à courte et longue distance, le déploiement massif de différentes technologies intégrées et notamment les systèmes d'informatique en nuage (Cloud Computing) et d'informatique en périphérie (Edge Computing) ainsi que le besoin de communication en temps réel.

Une alternative avantageuse consiste à tirer parti de cette diversité de technologies de connectivité et à créer une solution hybride qui surmonte les limites d'une technologie avec une autre, ce qui entraîne de meilleurs résultats, une disponibilité accrue et une meilleure gestion de la connectivité pour les projets à grande échelle.

Le but de notre travail est d'intégrer différents types de connectivité dans une même solution IdO pour démontrer l'efficacité d'une telle approche. La technologie du Bluetooth, le wifi

et les réseaux cellulaires seront intégrés dans notre solution étant donné qu'ils sont les technologies les plus populaires et largement utilisées dans le monde réel.

L'élément nouveau dans notre solution et qui est apparue pour les applications IdO, c'est la connectivité LoRa qui fait partie de la catégorie de réseau étendu à basse consommation tel que Sigfox qui est un autre protocole de communication IdO qui offre des avantages presque similaires à la LoRa telle que sa faible consommation d'énergie, idéal pour les appareils simples avec une portée de 25Km. Cependant, son inconvénient est la dépendance à la compagnie propriétaire ainsi que sa non-disponibilité dans plusieurs endroits ce qui a fait basculer notre choix pour la technologie LoRa dans la réalisation de notre travail.

MQTT

Étant donné que les objets ont une forte contrainte en termes de consommation d'énergie et de capacité de calcul, les protocoles d'application Internet, tels que HTTP et REST, ne conviennent pas.

En effet, REST est un style architectural conçu comme un modèle de demande / réponse qui communique via HTTP. Il permet de gérer, identifier et manipuler des ressources par l'intermédiaire d'une API. Cette interface correspond à un ensemble d'URI accessible via les différentes méthodes (GET, PUT, POST et DELETE) des requêtes HTTP. Cependant, plusieurs de ces caractéristiques limitent de son utilisation dans des systèmes IdO tels que le besoin de réponse instantanée, car REST est une connexion unidirectionnelle. La connexion au serveur est intermittente. Le client se connecte au serveur lorsque cela est nécessaire pour pousser les données du client et extrait les données vers le client. Le serveur doit attendre que les clients se connectent pour envoyer les données destinées au client. Par conséquent, cela oblige l'utilisateur à attendre la connexion client.

Un autre de ces inconvénients est la consommation d'énergie ou dans le REST, la plupart de l'énergie est perdue sur les ressources utilisées pour la connexion et la déconnexion et le nettoyage des ressources sur le serveur et le client, ce qui limite la durée de vie des batteries.

Son autre point faible est la rapidité et la livraison, ce dernier se base sur la requête-réponse pour l'informatique client-serveur où le client doit explicitement demander des ressources, puis le serveur répond avec elles.

Finalement, il n'est pas toujours optimisé pour les appareils mobiles ce qui le rend un choix secondaire pour les systèmes IdO.

Par conséquent, nous avons vu l'émergence de nouveaux protocoles d'application spécialement développés pour l'IdO. Telle que l'MQTT qui est celui utilisé dans notre solution [28].

MQTT est un protocole de messagerie spécialement conçu pour les applications machine a machine (M2M) et IdO, principalement en raison de sa légèreté (faible consommation de bande passante et d'énergie). Il permet la publication de nouvelles données dans le temps, sans nécessiter d'interroger un serveur, ce qui permet une réactivité et une économie de bande passante supplémentaire. De plus, MQTT offre une qualité de service (QoS) à 3 niveaux :

- QoS 0 : transmission des messages sans acquittement.
- QoS 1 : transmission des messages avec accusé de réception.
- QoS 2 : transmission de message avec accusé de réception et vérification du message transmis pour éviter que le même message ne soit transmis plusieurs fois.

Le tableau 2 ci-dessous résume les avantages et inconvénients du protocole MQTT.

Table 2. Avantages et inconvénients du protocole MQTT

	Avantages	Inconvénients
MQTT	<ul style="list-style-type: none"> - Poids léger. - Supporte la publication / abonnement et le modèle de demande / réponse. - Taille réduite de l'en-tête (deux octets). - Simplicité et rapidité de mise en œuvre. 	<ul style="list-style-type: none"> Aucun support de file d'attente de messages. - Grande consommation d'énergie grâce à la connexion basée sur TCP.

2.2.2. L'informatique en périphérie et l'Internet des objets

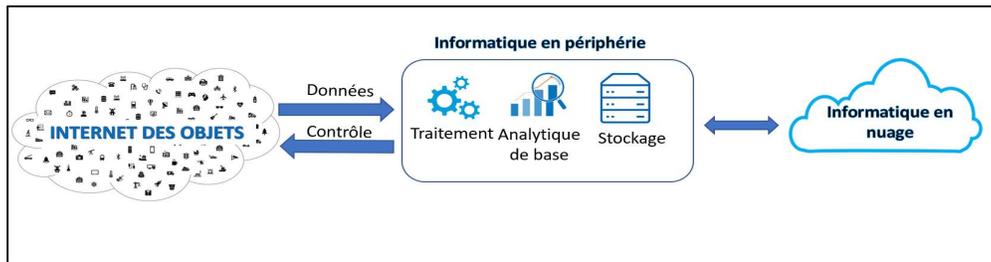


Figure 4. L'informatique en périphérie et l'Internet des objets

La figure 4, présente une architecture avec une couche d'informatique en nuage (Cloud Computing) et d'informatique en périphérie (Edge Computing) qui montre que les données provenant des appareils et capteurs sont transmises aux passerelles en périphérie à proximité des objets, où elles font l'objet d'un premier traitement avant d'être transférées. En effet, ces données sont ensuite envoyées, si nécessaire, à un ordinateur central du nuage (Cloud) chargé d'effectuer des analyses plus détaillées ou d'assurer l'archivage de ces données [30, 31].

Face à la nécessité d'une plus grande vitesse de traitement pour agir dans certains cas d'utilisation qui doivent être en temps réel et connaissant les limites du nuage (Cloud) en termes de latence puisque les informations doivent être transmises pour être traitées et renvoyées, l'informatique en périphérie (Edge Computing) est une alternative qui vient faire face à ces limitations et besoins essentiels, en rapprochant le traitement des données et les dispositifs de prise de mesures et actionneur, ce qui fait en sorte que le temps de réponse est réduit par rapport à la communication et au stockage dans le nuage (Cloud). De plus, l'informatique en périphérie (Edge Computing) contribue à une gestion des données plus efficace, car le traitement des données en périphérie permettra une bien meilleure qualité de filtrage et une priorisation des données plus efficace. En effet, en éliminant les données inutiles et en ne présentant que les données importantes de bonne qualité pour le traitement et la sauvegarde dans le nuage (Cloud) ce qui donne lieu à de meilleures analyses et prévisions.

2.2.3. L'informatique en nuage et l'Internet des objets

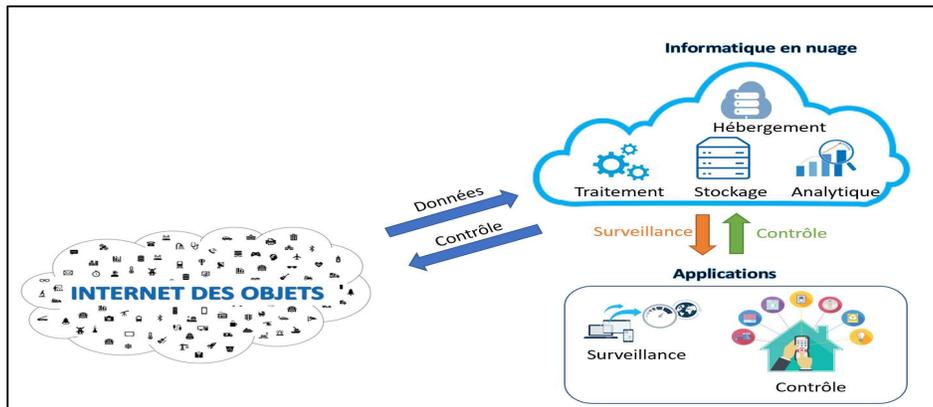


Figure 5. L'informatique en nuage et l'Internet des objets

Comme le montre la figure 5, l'IdO est responsable de la création d'énormes blocs de données tandis que le nuage (Cloud) crée un chemin pour le transfert, la maintenance et le stockage des données. En plus, le nuage (Cloud) permet le stockage et l'accès instantané aux ressources. Même en cas de catastrophe soudaine, le nuage (Cloud) conservera les données et fournira des informations en temps réel aux systèmes et applications [32,33].

Dans les systèmes IdO, d'énormes quantités de données sont générées, mis à part la nécessité de leur stockage pour une consultation préalable, elles doivent être traitées et structurées pour obtenir des informations pertinentes et utiles. Ce besoin a fait du nuage (Cloud) et de l'IdO deux technologies Internet futurs très étroitement associées et la convergence de ces deux technologies a apporté de nombreux avantages parmi eux citant, l'infrastructure tierce qui offre le nuage (Cloud) pour les systèmes IdO, les immenses espaces de stockage pour partager et enregistrer les données à utiliser, augmenter les performances avec les ressources disponibles qui permettent de meilleures performances pour le traitement et la transmission des données et pour terminer l'avantage du mode de paiement pour les plateformes nuage (Cloud), de payer que pour ce qui est utilisé (Pay-as-you-go), ce qui permet d'économiser de l'argent étant donné que les utilisateurs ne se soucient plus de l'achat d'espace de stockage, ils peuvent facilement faire évoluer le stockage à mesure que les données générées augmentent et paient pour le stockage qu'ils consomment.

2.3. Travaux connexes

Un aperçu de l'IdO est présenté dans [34], couvrant l'évolution, l'architecture, les protocoles et les applications IdO. Les auteurs de l'article soulignent l'importance d'avoir des capteurs intelligents qui communiquent directement sans intervention humaine.

Dans [35], les auteurs proposent un système basé sur un Raspberry Pi 3 qui contrôle plusieurs Arduino et en déployant des capteurs, moteurs et autres composants, pour la mise en œuvre d'un système capable de détecter les incendies en fournissant l'emplacement de la région affectée. Le fonctionnement du système présenté passe par les capteurs qui collectent des données et les envoient aux Arduino pour traitement. Par la suite, le Raspberry Pi reçoit ces données et envoie un message d'avertissement à l'administrateur à l'aide du module GSM. Pour finir, l'administrateur vérifiera la validité du message d'avertissement du système et confirmera ou niera l'existence d'une alerte.

En poursuivant avec le même type de systèmes, dans [36], la description de la conception d'un système de surveillance et de contrôle du feu est abordée où les auteurs ont utilisé divers capteurs tels que les capteurs de flammes, de la fumée et de gaz qui sont gérés par un Raspberry Pi pour détecter le feu et observer le processus d'extinction. Ils ont également utilisé le système GSM / GPS pour localiser et fournir l'emplacement de l'incendie.

Dans [37], les auteurs présentent un système d'alarme d'incendie conçu avec le module sans fil ZigBee. Le système composé d'un Arduino Uno, d'un capteur de température, d'une alarme sonore et d'un logiciel pour la mise en place d'un système capable en utilisant le capteur de température de détecter le feu et de transmettre des signaux à 10 mètres de distance.

Dans un autre type de système mis en place en [38], les auteurs présentent la réalisation d'un système de sécurité domestique à faible coût qui permet une surveillance à distance de la maison. Partant du fait qu'il existe plusieurs systèmes de sécurité à domicile ayant des problèmes tels que le retard dans les alertes par appels téléphoniques, SMS ou transferts de courriels dans des situations défavorables, ils proposent un système pour détecter un intrus ou tout autre événement lorsqu'il n'y a personne à la maison. Ce système utilise un capteur PIR pour détecter un intrus et une caméra pour capturer l'image de l'intrus. Il permet à l'utilisateur de contrôler la maison à distance de manière plus efficace et efficiente.

Pour finir, dans [39], les auteurs présentent une station météo basée sur l'IdO. Le système se compose de capteurs environnementaux pour surveiller la météo à un emplacement et télécharge les données sur un serveur en temps réel. En effet, les paramètres météorologiques sont téléchargés et enregistrés en temps réel et permettent également à l'utilisateur de définir des seuils d'alertes pour que le système transmette des avertissements à l'utilisateur. Cet article donne une idée claire des capteurs, de l'acquisition des données des capteurs, de la prise de décision automatique, du contrôle et de la façon dont les données des capteurs sont téléchargées sur un serveur.

2.4. Contribution

Après avoir parcouru la littérature sur le sujet et les systèmes de surveillance IdO mis en œuvre, nous avons observé que la réduction du temps de latence est un paramètre primordial dans les systèmes de surveillance en temps réel tels que la sécurité à domicile. De plus, nous avons remarqué l'efficacité de l'utilisation de composants tels que Arduino et Raspberry Pi pour mettre en place des systèmes IdO très efficaces.

Cependant, nous remarquerons qu'il est primordial de prêter une attention particulière au bon choix de connectivité qui est crucial pour le succès de tout projet IdO. En effet, les projets IdO varient dans leurs exigences et nombre d'entre eux utilisent différentes options de connectivité en fonction de leurs besoins, car la connectivité permet de prendre le contrôle total des flux de données.

Avec les technologies sans fil populaire existantes telles que les réseaux cellulaires, le Wi-Fi et le Bluetooth en place, de nombreux acteurs de l'IdO négligent l'importance d'obtenir une bonne connectivité IdO fiable et disponible, ce qui peut finalement conduire à un échec du projet en raison de l'incapacité à apprivoiser les immenses quantités de données en croissance rapide et de garantir la disponibilité en tout temps.

Avec la quantité et la diversité écrasantes des options disponibles, le paysage de la connectivité IdO en constante évolution se concentre actuellement sur la satisfaction des besoins des environnements gourmands en données au sein des applications client et industrielles de l'IdO. Dans un monde idéal, la solution de connectivité ultime et universelle offrirait une consommation d'énergie extrêmement faible pour les appareils tout en conservant la capacité de transmettre rapidement d'énormes morceaux de données sur de longues distances et garantir une disponibilité

en tout temps à des prix suffisamment bas pour que les entreprises restent économiquement viables. Cependant, étant donné l'hétérogénéité inhérente des cas d'utilisation au sein de l'IdO, la triste vérité est qu'aucune solution de connectivité IdO universelle parfaite n'a émergé pour répondre à tous les défis technologiques et analytiques liés à l'IdO.

Ainsi, trouver la meilleure solution pour un projet donné implique toujours de négocier un équilibre entre quatre paramètres fondamentaux pour la connectivité qui sont la disponibilité, la portée, la bande passante et la consommation d'énergie. Ce qui ne peut pas se faire en se basant sur une seule technologie de connectivité, mais par conséquent, une alternative qui peut être envisageable est d'identifier les besoins, les limites particulières de la solution IdO et adopter une approche hybride basée sur plusieurs technologies de connectivité permettant de surmonter les limites d'une technologie avec une autre, ce qui entraîne de meilleurs résultats, une disponibilité accrue et une meilleure gestion de la connectivité pour les projets IdO.

La contribution et la nouveauté de ce document est que le système mis en place offre une surveillance, un contrôle et des alertes en temps réel avec l'adoption d'une stratégie basée sur plusieurs technologies de connectivité et un système de sauvegarde local (backup) pour prévenir le risque de perte de données et assurer le bon fonctionnement du système en cas de perte de la connexion Internet.

Chapitre 3

Méthodologie de la recherche

Étant donné que l'objectif de la présente recherche est d'aboutir à la conception, implémentation, réalisation et mise en place d'un système Internet des objets fonctionnels, tout en passant par l'étude et la combinaison des connaissances dans le domaine. Dans ce chapitre nous présenterons notre choix de méthodologie qui est sur la méthode de recherche basée sur la conception (le design) [40].

3.1. Méthodologie adoptée

La méthode de recherche basée sur la conception (le design) est une méthodologie connue par sa flexibilité basée sur l'amélioration du processus d'étude par les analyses itératives, la conception, le développement et l'implantation dans un contexte réel en vue d'établir des principes et des propositions théoriques, elle permet la résolution d'un problème réel et actuel tout en construisant des principes de conception pouvant servir lors de futures prises de décision.

Trois caractéristiques principales découlent de cette méthodologie qui est sa facilité à permettre au chercheur de comprendre les phénomènes étudiés et à ajuster la conception. Sa flexibilité qui permet de modifier le protocole et la conception en cours d'expérimentation. Et finalement son adaptation à l'intégration des méthodes et techniques quantitatives et qualitatives.

Les phases suivies lors de l'adoption de la méthodologie de recherche basée sur la conception sont la prise de conscience du problème, la suggestion, le développement, l'évaluation et la conclusion.

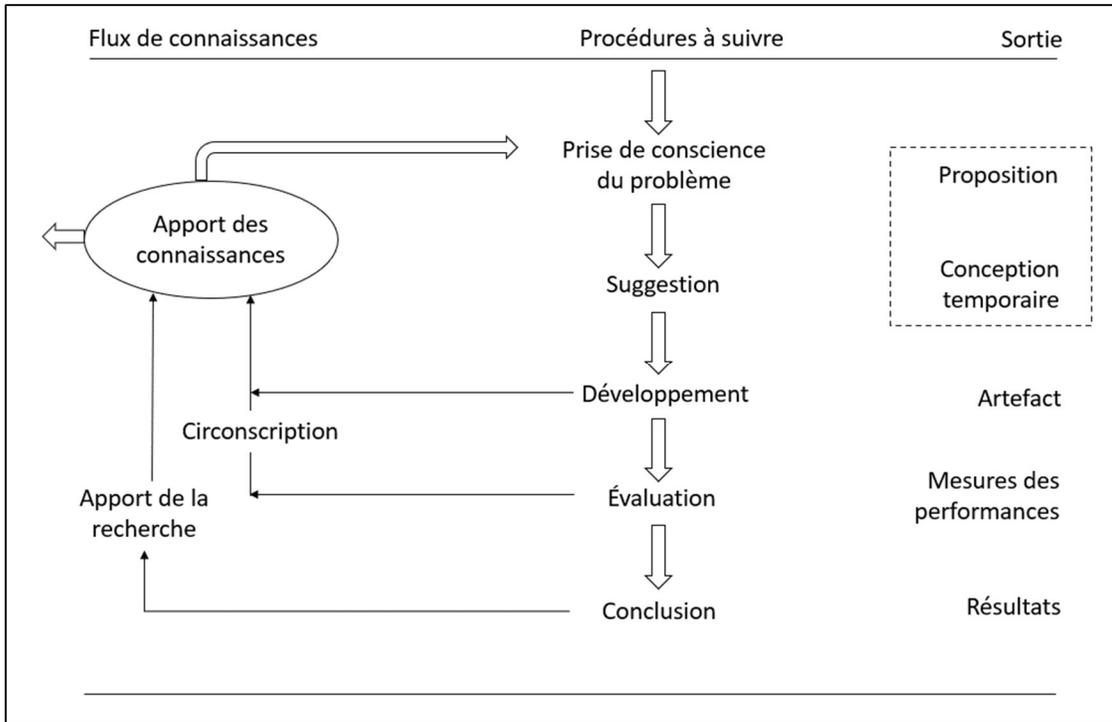


Figure 6. Les cycles de la méthodologie de recherche basée sur la conception

Prise de conscience du problème

Cette phase consiste à étudier un problème de recherche pertinent en fouillant dans plusieurs sources incluant les besoins des industries. Une prise de conscience des critères pour l'évaluation du résultat final de la recherche est nécessaire. Cette phase s'achève par une proposition de solution à un problème donné.

Suggestion

Les efforts mis dans la prise de conscience d'un problème et l'étude effectuée des connaissances actuelles ou de la base théorique, une conception temporaire d'un prototype de la solution pour résoudre le problème est proposée.

Développement

Au cours de cette phase, il y aura la mise en œuvre de la conception qui se termine par la création d'un artefact en utilisant les outils nécessaires tels que développement logiciel, création d'algorithmes, etc.

Évaluation

Une fois l'étape de développement est achevée et l'artefact créé, une évaluation de la solution selon les critères d'acceptation énumérée sera effectuée. Une analyse des écarts quantitatifs et qualitatifs entre les attentes et ce qui a été fait va démontrer soit l'acceptation soit le rejet ou le besoin de modification, d'où la suggestion d'une nouvelle conception et dans ce cas des recherches supplémentaires doivent être faites dans le but de comprendre les raisons de l'écart entre l'artefact et les critères théorique attendu.

Conclusion

Les résultats consolidés et les connaissances acquises au cours de l'effort sont résumés lors de cette phase qui marque la fin du cycle de recherche et la satisfaction du résultat soit en ayant éliminé les écarts entre la solution et les critères d'acceptation ou bien en étant arrivée à un résultat avec un écart acceptable. En plus, il y aura énumération des connaissances acquises lors de cette phase.

3.2. Approche de recherche

La présente recherche est orchestrée par la méthodologie de recherche basée sur la conception (le design). En effet, après avoir étudié et présenté un besoin dans un domaine technologique qui est l'Internet des objets en mentionnant les défis suivis de la présentation d'une proposition, une étude de littérature sur l'IdO a été menée.

Une fois la prise de conscience du problème est faite qui est le risque de perte des données due à un imprévu, et la suggestion d'une solution temporaire, nous sommes passés à la phase de développement. Au début, nous avons implémenté une solution IdO basée sur des capteurs et des actionneurs qui offrait le suivi d'état en temps réel à travers une interface graphique pour l'utilisateur et en utilisant un seul protocole de connectivité qui est le Wi-Fi.

Ce pendant lors de la phase d'évaluation, nous avons remarqué des limitations au niveau du système dû à l'utilisation d'un seul moyen de connectivite qui est le Wi-Fi, ce qui nous a amenés à une nouvelle itération lors de laquelle nous avons proposé d'améliorer notre solution en intégrant plusieurs types de connectivité, ajouter de nouvelles composantes, intégration du stockage des données, synchronisation en local et en nuage.

Finalemment, la dernière étape est l'évaluation et l'étude de l'efficacité de la solution proposée en passant à la conclusion et la présentation des résultats

3.3. Reconcevoir et affiner l'expérience

L'aspect itératif pour la recherche basée sur le design vise à affiner continuellement les solutions implémentées. En effet, grâce aux connaissances acquises auparavant, les produits d'essais et d'erreurs qui sont renforcés par une analyse systématique des performances dans des environnements réels, de nouvelles propositions apparaissent pour continuer à faire avancer et évoluer les travaux.

L'évolution et les améliorations du produit débutent par un nouveau cycle de recherche qui commence en passant par les différentes étapes de notre méthodologie qui sont la prise de conscience du problème (peut être sous forme d'amélioration, ajout, suppression, etc.), la suggestion, le développement, l'évaluation et la conclusion.

À partir de cette nouvelle itération de conception, des tests dans un environnement réel, de l'évaluation et en réfléchissant au succès et à l'évolution de la solution, nous développerions une caractérisation plus approfondie des systèmes IdO qui pourraient mieux élargir nos connaissances pour l'amélioration de notre système IdO ou la mise en place d'un nouveau système IdO plus évolué et plus robuste.

Chapitre 4

Présentation de la solution proposée

Plusieurs travaux et systèmes IdO ont été réalisés tels qu'un système basé sur un Raspberry Pi 3 qui contrôle plusieurs Arduino en déployant des capteurs, moteurs et autres composants, pour la mise en œuvre d'un système capable de détecter les incendies en fournissant l'emplacement de la région affectée. Le fonctionnement du système passe par les capteurs qui collectent des données et les envoient aux Arduino pour traitement. Par la suite, le Raspberry Pi reçoit ces données et envoie un message d'avertissement à l'administrateur à l'aide du module GSM. Pour finir, l'administrateur vérifiera la validité du message d'avertissement du système et confirmera ou niera l'existence d'une alerte.

Un autre système semblable utilise divers capteurs tels que les capteurs de flammes, de la fumée et de gaz qui sont gérés par un Raspberry Pi pour détecter le feu et observer le processus d'extinction. Un système GSM / GPS est utilisé pour localiser et fournir l'emplacement de l'incendie.

Pareillement, un système d'alarme d'incendie conçu avec le module sans fil ZigBee. Le système composé d'un Arduino Uno, d'un capteur de température et d'une alarme sonore, permet de transmettre des signaux à 10 mètres de distance.

Dans un autre type de système, la réalisation d'un système de sécurité domestique à faible coût qui permet une surveillance à distance de la maison. Un Raspberry pi, est utilisé pour minimiser le délai pendant le processus d'alerte par courriel.

Dans une station météo basée sur l'IdO, le système se compose de capteurs environnementaux pour surveiller la météo à un emplacement et télécharge les données sur un serveur en temps réel. En effet, les paramètres météorologiques sont téléchargés et enregistrés en temps réel et permettent également à l'utilisateur de définir des seuils d'alertes pour que le système transmette des avertissements à l'utilisateur grâce au Raspberry pi.

Les études nous ont montré différents types de systèmes IdO efficaces et fonctionnels qui ont été réalisés et développés en se basant sur de composantes telles que les cartes Arduino et

Raspberry Pi ce qui démontre l'efficacité de la combinaison de telles composantes pour la mise en place des systèmes IdO.

De nos jours, plusieurs autres cartes existent sur le marché qui peuvent gérer des projets similaires et offrent une même fiabilité et à peu près les mêmes fonctionnalités pour un prix similaire. Cependant, notre choix des cartes utilisées dans l'implémentation de notre solution est basé d'une part sur la popularité, la documentation et la disponibilité en autre possession des cartes et composantes qui sont suffisantes pour la satisfaction de notre besoin au niveau des fonctionnalités à mettre en place et d'autre part le but de notre travail qui est de prouver le concept de la possibilité de l'utilisation d'une connectivité hybride dans un seul système IdO dépendamment des composantes et des cartes utilisées.

Dans ce chapitre nous allons présenter les outils utilisés pour la réalisation de notre travail. En effet, nous commencerons par une description des différentes composantes de notre solution, logiciels et matériels pour finir avec l'élaboration de l'architecture finale en passant par la conception du système, la configuration des serveurs et le développement des micrologiciels.

4.1. Technologies et composants

4.1.1. Systèmes sur une puce (SoC)

Le système sur une puce [41] est un circuit électronique intégré qui combine plusieurs composantes sur une seule carte. Ayant des fonctions analogiques, numériques, à signaux mixtes ou radiofréquences, il est composé généralement d'une unité de traitement graphique (GPU), une unité de traitement centrale (UC) pouvant être multicœur, et une mémoire système (RAM), il permet de fournir une fonctionnalité complète à une application.

Caractérisé par leurs petites tailles permettant la réduction de l'espace physique requis pour le système, sa consommation d'énergie faible et ces bonnes performances, ils sont intégrés dans les appareils mobiles tels que les téléphones intelligents, montre intelligente et les tablettes.

Avec l'émergence de l'Internet des objets, un grand nombre d'inventions IdO ont été et seront mises en œuvre avec un seul système sur puce afin de fournir le plus haut niveau d'intégration et de conservation de la surface en utilisant des cartes telles que Raspberry Pi.

4.1.2. Arduino Uno

La carte Arduino Uno [42] est un microcontrôleur qui provient d'une société appelée Atmel. Caractérisé par son faible coût et ça facilite d'intégration avec d'autres cartes et composants lui permettant d'offrir une disponibilité Wi-Fi ou Bluetooth. En plus sa faible consommation d'énergie la rend idéale pour plusieurs systèmes Internet des objets.

Table 3. Caractéristiques de l'Arduino Uno

	Arduino Uno
Microcontrôleur	ATmega328
Fréquence d'horloge	16 MHz
Alimentation	5V
E/S numériques	14
E/S analogiques	6
Courant max par broches E/S	40mA
Courant max sur sortie 3,3V	50mA
Mémoire programme flash	32 KB
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB
Température	-40°C à 85°C

4.1.3. Raspberry Pi 3 Model B

Raspberry Pi [43] est un ordinateur complet de la taille d'une carte de crédit, fonctionnant sous le système d'exploitation Linux qui peut être utilisé pour les mêmes tâches que celles effectuées avec un ordinateur de bureau normal.

Ayant les ports nécessaires pour le branchement de périphérique tels que souris et clavier. Dans le modèle Raspberry Pi 3 Model B, on trouve aussi l'intégration du Bluetooth qui est intéressant pour l'utilisation dans des systèmes Internet des objets ayant besoin de tel type de connectivité.

Table 4. Caractéristiques du Raspberry Pi 3 Model B

	Raspberry Pi 3 Model B
Puce (SoC)	Broadcom BCM2837B0
Processeur	Cortex-A53 64-bit
Nombre de cœurs	4
GPU	VideoCore IV
Fréquence processeur	1,4 GHz
Mémoire vive	1 GB DDR2
Wi-Fi	2.4 GHz and 5 GHz 802.11 b/g/n/ac
Bluetooth	4.2, BLE
Consommation	2.5 A
Alimentation	5V
GPIO	40
Température	0 à 50°C

4.1.4. ESP8266

L'ESP8266 [44] est une carte à microcontrôleur programmable sans fil idéal dans les systèmes Internet des objets. Avec une pile de protocoles TCP / IP, elle peut être intégrée à d'autre microcontrôleur tel que Arduino pour lui permettre l'accès au réseau Wi-Fi.

Table 5. Caractéristiques de l'ESP8266

	ESP8266
Mémoire vive	36 KB
CPU	Tensilica LX106 32 bits @ 80 MHz (jusqu'à 160 MHz)
Flash	1, 2, 4, 8 ou 16 MB
Alimentation	3.0 V - 3.6 V
Consommation	80 mA en Moyenne
Température	-40°C à 125°C
Wi-Fi	802.11b/g/n
Bluetooth	-
UART	2
I2C	1
GPIO	17

4.1.5. ESP32

L'ESP32 [44] est le successeur de l'ESP8266. Il offre une connexion Wi-Fi plus rapide, plus de ports d'entrée sortie appelés « GPIO » et prend en charge les technologies Bluetooth 4.2 et Bluetooth basse consommation. Ce qui le rend extrêmement puissant dans les systèmes Internet des objets.

Table 6. Caractéristiques de l'ESP32

	ESP32
Mémoire vive	520 KB

CPU	Tensilica Xtensa LX6 32 bits Dual-Core @ 160 / 240Mhz
Flash	64 MBytes
Alimentation	2.2 V - 3.6 V
Consommation	80 mA en Moyenne
Température	-40°C à 125°C
Wi-Fi	802.11b/g/n
Bluetooth	4.2 BR/EDR + BLE
UART	3
I2C	2
GPIO	32

4.1.6. SIM5320A

Le Tinsine SIM5320 3G [45] permet d'utiliser le réseau de téléphonie mobile 3G / GSM pour recevoir des données d'un lieu distant. Grâce au 3G, cette carte offre une meilleure couverture, une compatibilité ascendante avec le GSM et même un module GPS intégré pour la géolocalisation et le suivi. Ce module permet d'ajouter les fonctionnalités de la téléphonie cellulaire à un projet Internet des objets tels que :

- ✓ Permettre la connexion à n'importe quel réseau GSM.
- ✓ GPS intégré.
- ✓ Émettre et recevoir des appels.
- ✓ Envoyer et recevoir des SMS.
- ✓ Envoyer et recevoir des données GPRS.

Table 7. Caractéristiques du SIM5320A

	SIM5320A
Température	-40°C à +85 °C
Consommation en repos	1.5mA
GPIO	13
Dual-Band	UMTS/HSDPA 850/1900MHz
Quad-Band	GSM/GPRS/EDGE 850/900/1800/1900MHz
Alimentation	3.4 à 4.2V

4.1.7. REYAX RYLR896

Le module émetteur-récepteur REYAX RYLR896 [46] est doté du modem longue portée LoRa qui fournit une portée ultra longue avec diffusion du spectre et immunité aux interférences élevée tout en minimisant la consommation d'énergie.

Caractérisé par :

- ✓ Excellente immunité aux interférences.
- ✓ Faible consommation d'énergie.
- ✓ Haute sensibilité.
- ✓ Conçu avec une antenne intégrée au circuit imprimé.
- ✓ Transmission pouvant atteindre 15km.

Table 8. Caractéristiques du REYAX RYLR896

	REYAX RYLR896
Fréquence	868/915 MHz
Température	-40 à + 85
Alimentation	2 à 3.6 V

Puissance de sortie RF	-4 à 15 dBm
Communication	4,5 à 15 km
Consommation en transmission	43mA
Consommation en réception	16,5mA
Consommation en repos	0.5uA

4.1.8. Micrologiciel (Firmware)

Un logiciel spécial intégré dans les périphériques matériels tels que les lecteurs optiques, une carte réseau, un routeur, une caméra [47]. Pouvant être écrits dans une mémoire morte (ROM), une mémoire EPROM (Mémoire morte programmable) effaçable ou une mémoire flash, ils permettent aux périphériques de donner des instructions permanentes permettant de communiquer avec d'autres périphériques et d'effectuer des tâches telles que des tâches d'entrées / sorties de base.

Ils sont nécessaires pour le fonctionnement du matériel et nécessitent parfois des mises à jour pour s'adapter aux changements. La mise à jour des micrologiciels peut se faire à distance sans fil, mais parfois nécessite l'intervention humaine et la mise à jour manuelle.

De nos jours, les micrologiciels intégrés existent dans plusieurs équipements tels que les téléphones intelligents et les aux appareils destinés pour l'Internet des objets.

Plusieurs langages de programmation existent pour le développement de micrologiciel dont les plus populaires sont C/C++, Arduino et Python.

- **Python**

Python est un langage de script interprété qui permet une approche de programmation modulaire et orientée objet. Python est devenu l'un des langages de programmation les plus populaires pour l'analyse de donnée et l'apprentissage machine. Cependant, il permet aussi la programmation et le contrôle de matériel tel que les langages de bas niveau Arduino, C ou C ++.

- **C/C++ et Arduino**

Ces trois langages de programmation sont différents, néanmoins se ressemblent du fait de leur syntaxe et leurs utilisations pour la création des micrologiciels pour les systèmes embarqués. En effet, ils permettent de manipuler directement la mémoire, les registres et les ports d'entrée et sortie.

4.1.9. Capteurs

Le but d'un capteur est de transformer propriété physique d'entrée et de la convertir en un signal électrique compatible avec les circuits électroniques [48]. Les capteurs sont des appareils électroniques qui mesurent une qualité physique telle que la lumière ou la température et convertissez-la en tension.

4.1.10. Serveur MQTT (Éclipse Mosquito)

Éclipse Mosquito [49] est un serveur MQTT libre qui implémente différentes versions du protocole MQTT. Mosquitto est léger et convient à une utilisation sur tous les appareils, allant des mini-ordinateurs à basse consommation au serveur puissant.

Sa facilité d'intégration le rend approprié pour les projets Internet des objets, impliquant des capteurs de faible puissance ou des appareils mobiles tels que des téléphones, des ordinateurs ou des microcontrôleurs. Sa capacité de transmission de données à travers les WebSockets et son autre avantage intéressant pour la communication à travers le Web.

En effet, le WebSocket est un protocole de communication informatique fournissant des canaux de communication en duplex via une seule connexion TCP / IP. MQTT sur WebSocket permet de recevoir des données MQTT directement dans un navigateur Web. Cela est important, car le navigateur Web peut devenir l'interface pour afficher les données MQTT.

4.1.11. LAMP

Une pile « LAMP » est un groupe de logiciels libres généralement installé ensemble pour permettre à un serveur d'héberger des sites Web dynamiques et des applications Web. Ce terme est en fait un acronyme qui représente le système d'exploitation Linux, avec le serveur Web

Apache. Les données du site sont stockées dans une base de données MySQL et le contenu dynamique est traité par PHP.

4.1.12. XAMPP

XAMPP est une compilation d'un groupe de logiciel libre fonctionnant sur: Linux, Windows et Mac OS X et contenant un serveur Web Apache, le système de gestion de base de données relationnelle MySQL (ou MariaDB) et les langages de script Perl et PHP.

- **Apache**

Le serveur Web libre Apache est le serveur le plus utilisé au monde pour la diffusion de contenu Web. L'application serveur est mise à disposition en tant que logiciel gratuit par Apache Software Foundation.

- **MySQL / MariaDB**

Dans MySQL, XAMPP contient l'un des systèmes de gestion de bases de données relationnelles les plus populaires au monde. En combinaison avec le serveur Web Apache et le langage de script PHP, MySQL offre un stockage de données pour les services Web.

- **PHP**

Le langage de programmation côté serveur PHP permet aux utilisateurs de créer des applications ou des sites Web dynamiques. PHP peut être installé sur toutes les plateformes et prend en charge un certain nombre de systèmes de base de données variés.

4.1.13. Technologies Web : HTML, CSS, JavaScript, PHP, MySQL

La mise en place de note application Web pour le contrôle et suivi du système grâce à une interface Web réactive (responsive) est base sur des technologies Web telle que :

- **HTML**

HTML est le langage utilisé pour créer des pages Web . «Hypertexte» fait référence aux hyperliens qu'une page HTML peut contenir. Le «langage de balisage» fait référence à la manière dont les balises sont utilisées pour définir la mise en page et les éléments de la page.

- **CSS**

Les feuilles de style en cascade sont utilisées pour formater la mise en page des pages Web. Ils peuvent être utilisés pour définir des styles de texte, des tailles de tableau et d'autres aspects des pages Web qui ne pouvaient auparavant être définis que dans le code HTML d'une page.

- **JavaScript**

JavaScript est un langage de programmation couramment utilisé dans le développement Web. Il a été initialement développé par Netscape pour ajouter des éléments dynamiques et interactifs aux sites Web. Il est un langage de script côté client, ce qui signifie que le code source est traité par le navigateur Web du client plutôt que sur le serveur Web. Cela signifie que les fonctions JavaScript peuvent s'exécuter après le chargement d'une page Web sans communiquer avec le serveur.

- **PHP**

PHP est un langage de script Web intégré au HTML. Cela signifie que le code PHP peut être inséré dans le HTML d'une page Web. Lors de l'accès à une page PHP, le code PHP est lu ou analysé par le serveur sur lequel la page réside. La sortie des fonctions PHP sur la page est généralement renvoyée sous forme de code HTML, qui peut être lu par le navigateur. Comme le code PHP est transformé en HTML avant le chargement de la page, les utilisateurs ne peuvent pas afficher le code PHP sur une page. Cela rend les pages PHP suffisamment sécurisées pour accéder aux bases de données et à d'autres informations sécurisées.

- **MySQL**

MySQL est un système de gestion de base de données relationnelle libre. Il est basé sur le langage de requête structurée (SQL), qui est utilisé pour ajouter, supprimer et modifier des informations dans la base de données. Les commandes SQL standard, telles que ADD, DROP, INSERT et UPDATE peuvent être utilisés avec MySQL. MySQL peut être utilisé pour une variété d'applications, mais se trouve le plus souvent sur les serveurs Web. Un site Web qui utilise MySQL peut inclure des pages Web qui accèdent aux informations d'une base de données. Ces pages sont souvent appelées dynamiques, ce qui signifie que le contenu de chaque page est généré à partir

d'une base de données lors du chargement de la page. Les sites Web qui utilisent des pages Web dynamiques sont souvent appelés sites Web basés sur des bases de données.

4.2. Conception de la solution

La solution proposée comprend plusieurs étapes telles que l'interfaçage du circuit, les capteurs, les actionneurs et les microcontrôleurs, le développement et l'optimisation du micrologiciel, la configuration et la mise en œuvre des serveurs et des bases de données, la création et le développement des interfaces utilisateur.

4.2.1. Vue d'ensemble du circuit

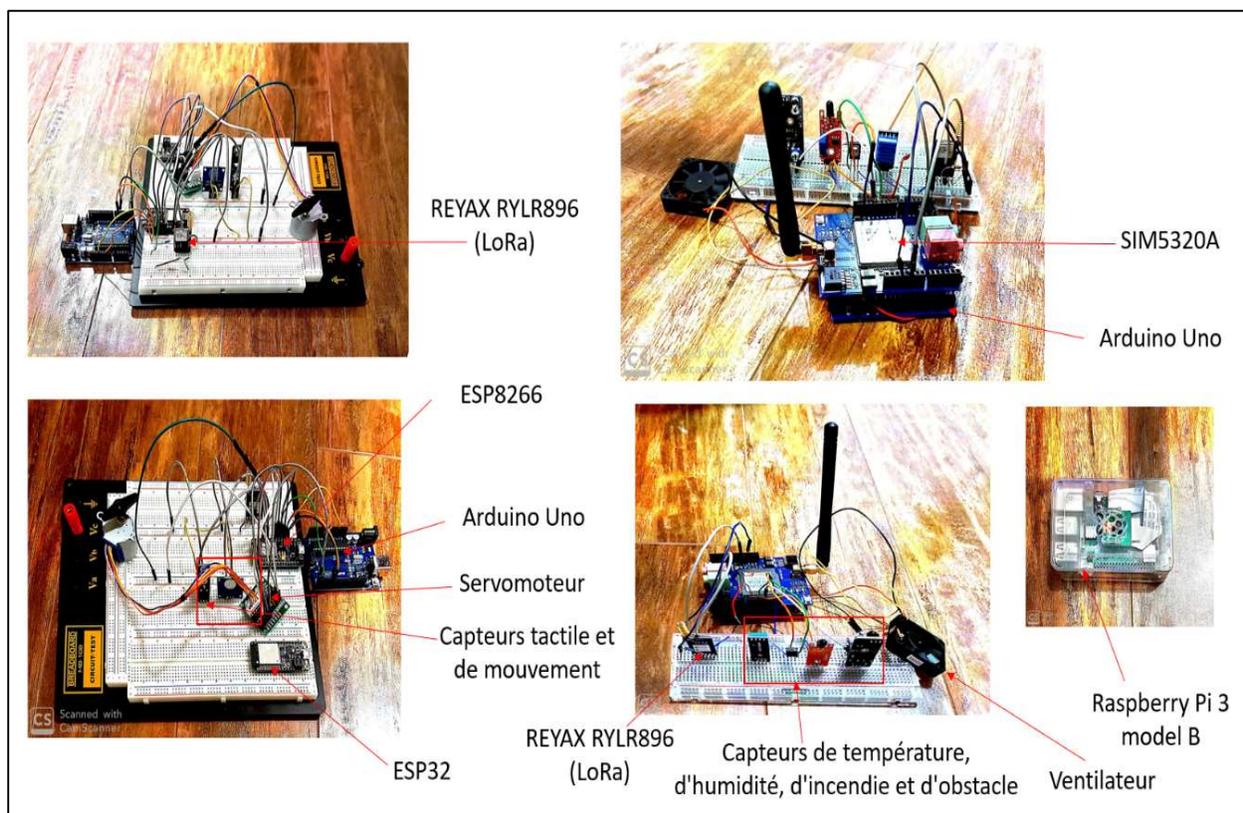


Figure 7. Les composants du circuit

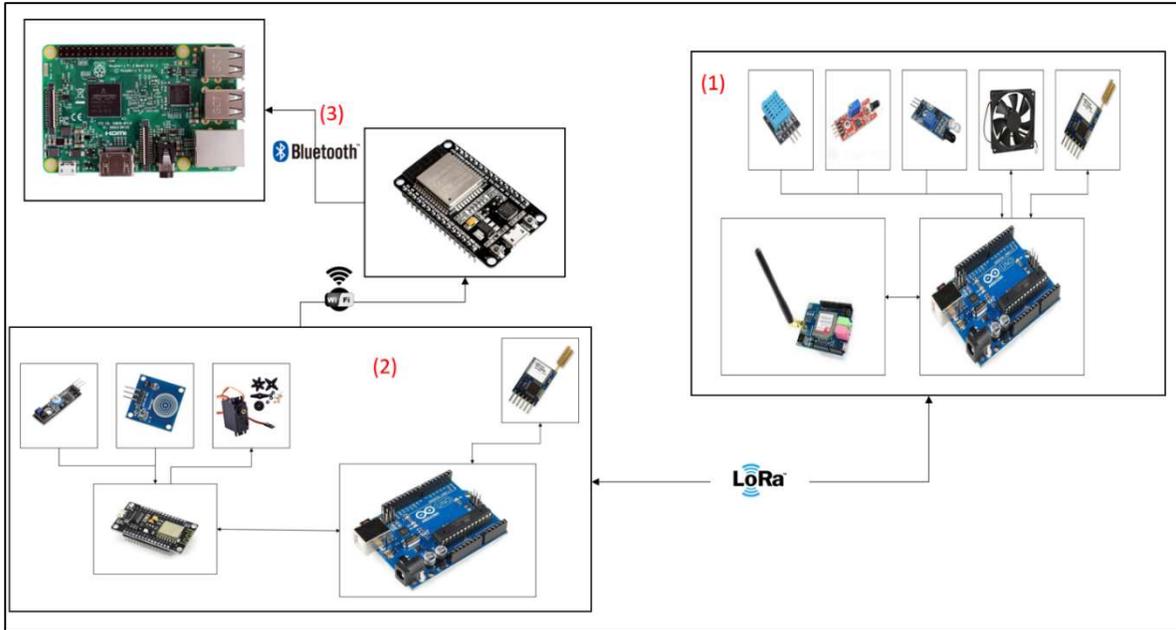


Figure 8. Vue d'ensemble du circuit

Notre solution est composée de trois modules physiquement indépendants avec chaque module qui comprend différents composants, offrant des fonctionnalités diversifiées et déployées dans différents endroits. Néanmoins, ces modules communiqueront efficacement et échangeront des données via différents protocoles de communication.

Chaque module joue un rôle dans notre système et contribue à l'efficacité et au bon fonctionnement du système. L'identification des composants inclus dans chaque module du système a été choisie après avoir réalisé une étude et une analyse approfondie des besoins de la solution. Ce processus nous a permis d'identifier les éléments à inclure dans chaque partie en fonction de l'objectif du système. Les paramètres du système tel que la sécurité, les éléments les plus critiques et les plus risqués, les spécificités de chaque composant, le taux et le retard de transmission, le résultat attendu et l'équipement disponible pour finalement proposer trois modules comme illustrés dans les figures 7 et 8 comme suit :

- Premier module comme indiquer sur la figure 9, contenant un Arduino Uno, REYAX RYLR896, SIM5320A, des capteurs et des actionneurs.

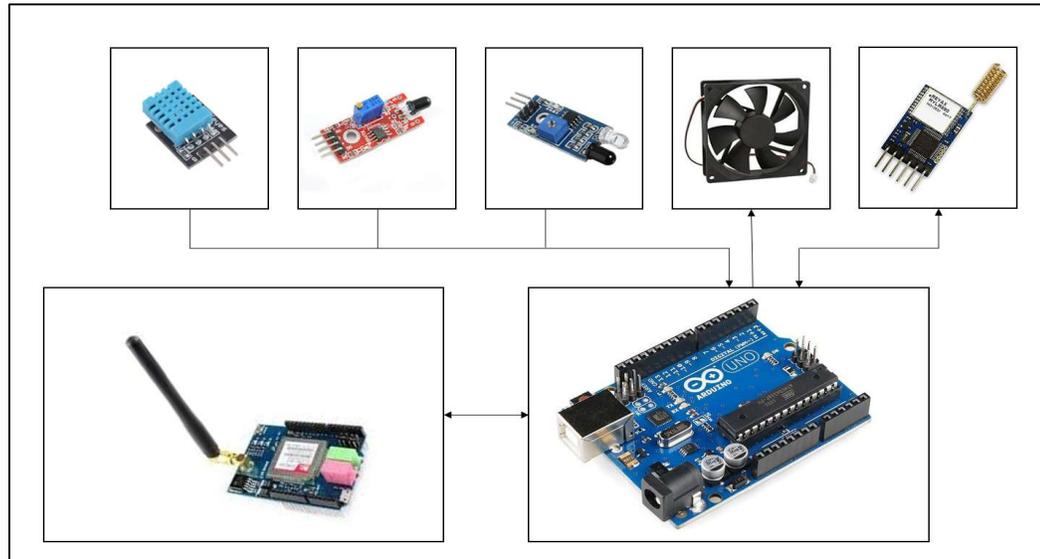


Figure 9. Premier module du circuit

- Deuxième module comme le montre la figure 10, contenant un Arduino Uno, REYAX RYLR896, ESP8266, des capteurs et des actionneurs.

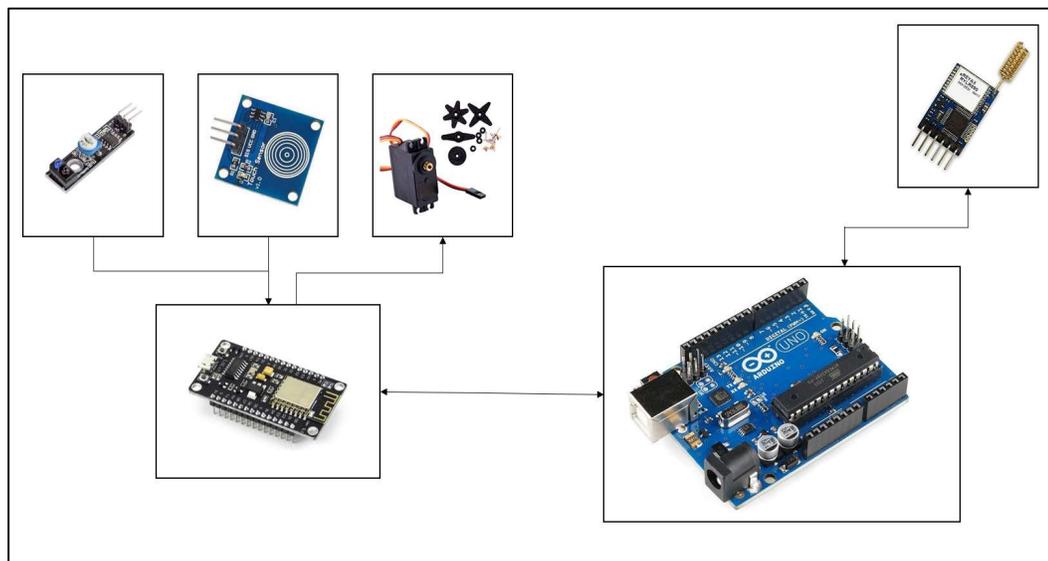


Figure 10. Deuxième module du circuit

- Troisième module comme illustré à la figure 11, contenant un ESP32 et un Raspberry Pi 3 modèle B.

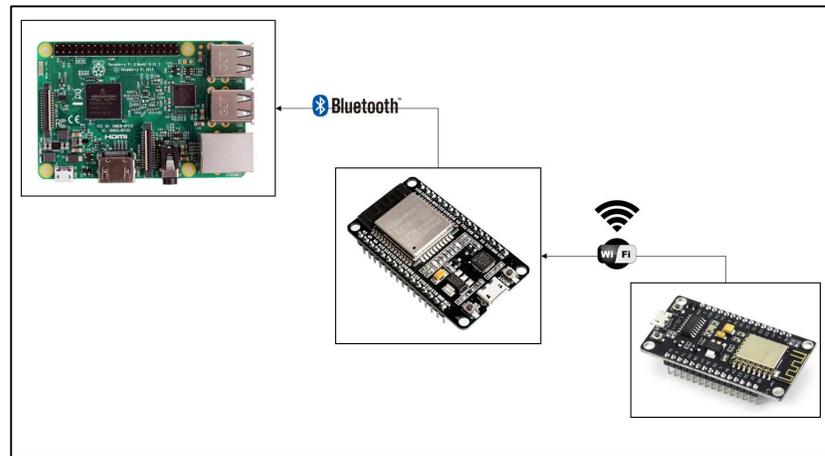


Figure 11. Troisième module du circuit

Pour chaque module, des micrologiciels (firmware) ont été programmés et intégrés en fonction du rôle, des composants et des besoins de ces derniers. Différents langages de programmation ont été utilisés tels que C / C ++, Arduino et Python.

4.2.2. Configuration des serveurs

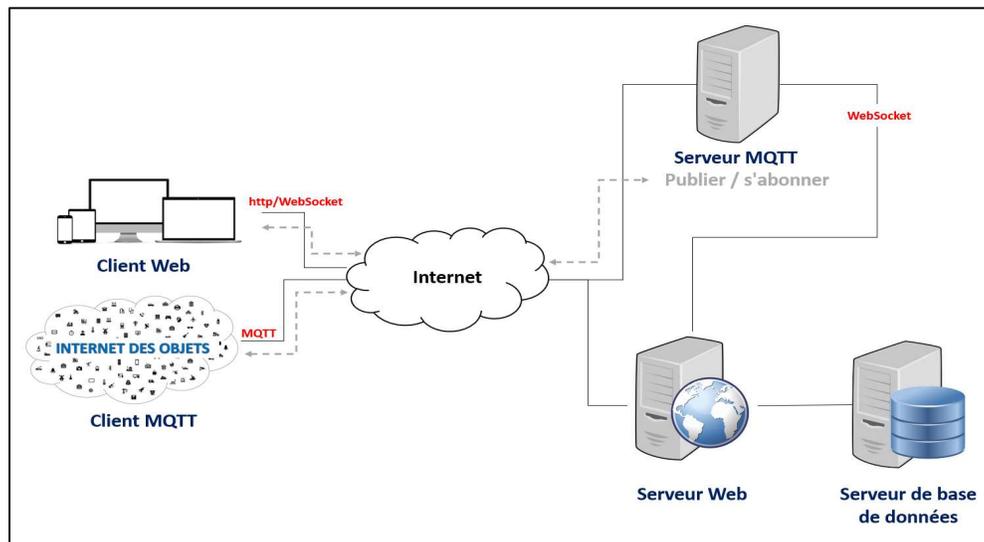


Figure 12. Architecture client / serveur de la solution

Comme le montre la figure 12, outre les circuits, la configuration et le développement, nous proposons l'installation des serveurs. En effet, nous configurons un serveur MQTT pour la

transmission des données entre notre système et les clients Web via les WebSockets. Cette méthode de communication permet à l'utilisateur de suivre l'état en temps réel grâce à une interface Web réactive (responsive) qui offrira la possibilité d'effectuer le contrôle et la surveillance des données.

De plus, nous mettrons en place notre serveur Web et notre serveur de base de données, de sorte que les données transférées soient collectées, traitées et stockées dans des bases de données pour une utilisation ultérieure pour les rapports. En effet, des rapports et statistiques seront mis à disposition de l'utilisateur pour permettre la prédiction et l'analyse d'état.

4.3. Architecture de la solution

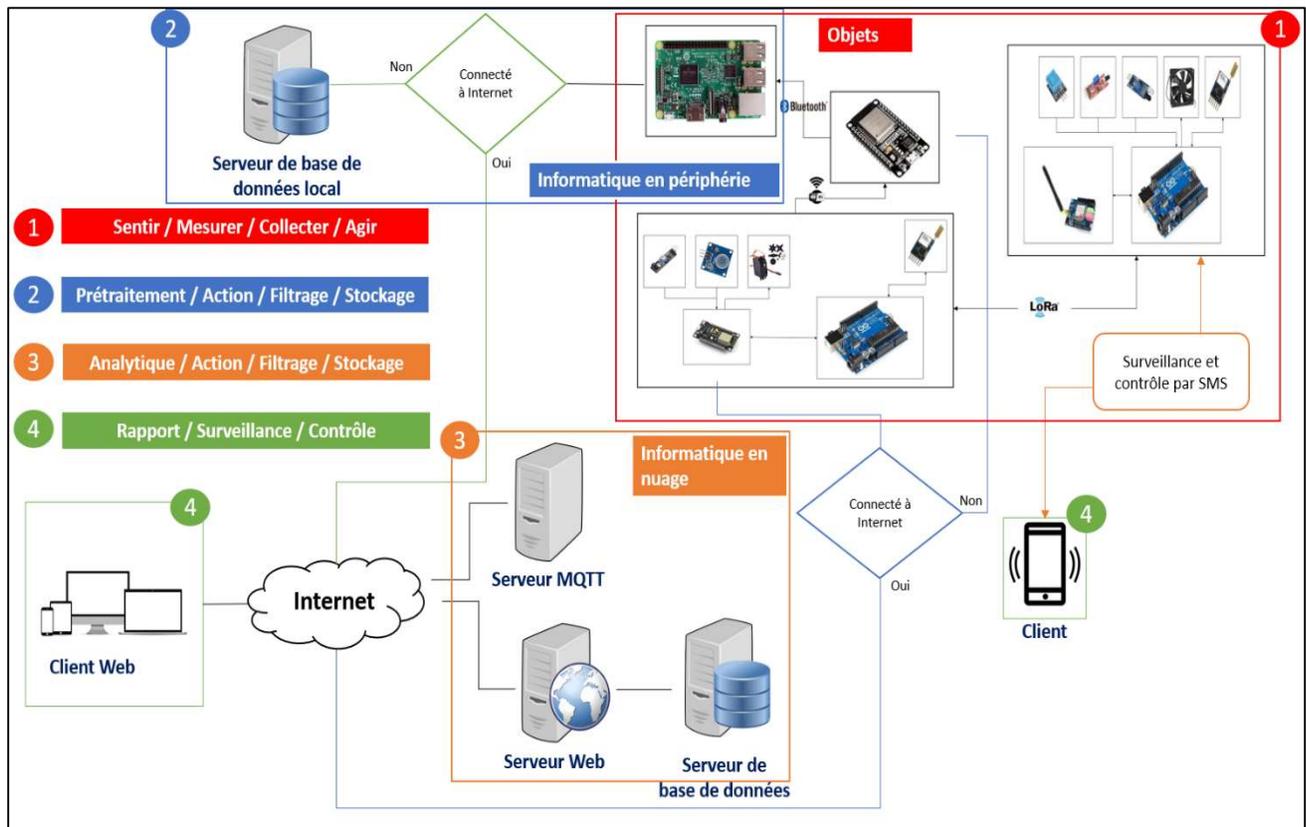


Figure 13. Architecture générale de solution proposée

Comme le montre la figure 13, nous déploierons des capteurs et des actionneurs sur deux groupes. Le premier groupe sera contrôlé par un Arduino Uno relié à un émetteur / récepteur LoRa et dispose d'une carte SIM5320A pour la communication cellulaire. L'Arduino Uno, d'un côté, contrôlera son premier groupe d'objets soit en effectuant des actions automatiques soit en ayant

reçu la commande de l'utilisateur qui pourra contrôler le système via l'interface Web ou par SMS, et de l'autre côté, transmettre les données au deuxième groupe d'objets.

Pour sa part, le deuxième groupe comprendra des capteurs / actionneurs qui seront contrôlés par l'ESP8266. Connecté à Internet via Wi-Fi, il recevra l'action demandée du client via l'interface Web grâce au serveur MQTT. En plus de la réception des actions, il effectuera la transmission des données reçues soit du premier groupe d'objets, en passant par un Arduino Uno intégrant un émetteur / récepteur LoRa qui communiquera pour l'échange de données avec l'Arduino Uno du premier groupe, via le protocole LoRa, ou la transmission de données de capteurs directement attachées à celui-ci.

Dans un premier temps, connecté au Web grâce à son module Wi-Fi qui lui permettra d'interagir avec le serveur MQTT vers lequel il recevra et transmettra les données, en cas de problèmes et de perte de cette communication il basculera pour la transmission des données à l'ESP32.

En effet, jouant le rôle d'un point d'accès, l'ESP32 offrira un réseau local sur lequel l'ESP8266 pourra se connecter et transmettre des données. Par la suite, ces données transmises à l'ESP32 seront à leur tour transférées au Raspberry Pi via Bluetooth.

Au niveau du Raspberry Pi, dès réception des données, celles-ci seront traitées et stockées dans une base de données configurée localement. Une fois la connexion à Internet rétablie, une synchronisation avec la base de données distante peut être effectuée, ce qui maintiendra la conformité des données et le bon fonctionnement continu du système et conduira à des résultats de haut niveau pour les statistiques et les prévisions. De plus, la configuration des serveurs Web, MQTT et de base de données permettra l'interaction avec l'utilisateur et le stockage de la base de données.

Outre l'interfaçage des composants de notre système et la configuration et l'implémentation des serveurs. Nous avons procédé à la mise en œuvre à plusieurs niveaux. En effet, des micrologiciels (firmwares) ont été développés et intégrés pour automatiser les processus des différents composants de notre système. Ensuite, nous avons conçu et créé une base de données pour le stockage à distance et reproduit une copie de cette base de données localement près du matériel. Cette dernière permettra le stockage temporaire des données transmises en cas de

problème. Après cela, la synchronisation avec la base de données distante sera effectuée en cas de besoin pour permettre la mise à jour des données collectées et traitées afin de maintenir la conformité des données stockées pour donner de meilleurs résultats analytiques.

Enfin, nous avons procédé au développement et à la mise en œuvre d'une interface Web réactive permettant la surveillance de l'état du système, ainsi que la fourniture de rapports et de différents formulaires graphiques pour les statistiques aux utilisateurs.

Chapitre 5

Validation expérimentale

Dans cette section nous présenterons les résultats obtenus de la solution mise en place dans le contexte d'un système IdO de suivi et contrôle à distance.

5.1. Mise en place du circuit

Dans cette partie nous présenterons l'interfaçage et le rôle de chaque module du circuit incluant les liaisons et la mise en place de différentes composantes qui permettront de garantir la communication et l'échange de données entre les différentes parties. En effet, à ce niveau des micrologiciels ont été programmés pour chaque partie exécutant des tâches et jouant un rôle spécifique. Cette phase représente la base de notre solution étant donné que l'échec du fonctionnement à ce niveau bloquera l'avancement de la réalisation de notre solution.

À cette étape nous avons expérimenté l'intégration de plusieurs composantes hétérogènes telles que capteurs, actionneur et microcontrôleur qui communiquent par le biais de différents protocoles de communications tels que filaire, LoRa, Wi-Fi, Bluetooth et Cellulaire.

Premier module du circuit

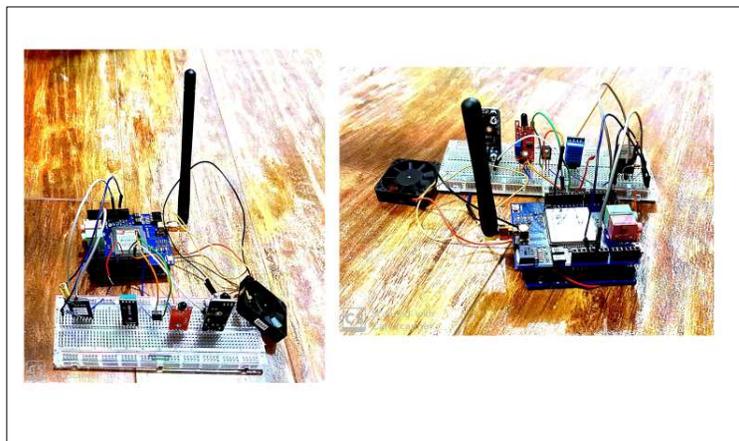


Figure 14. Vue réelle du premier module du circuit

Cette partie du système, comme le montre la figure 14 incluant les composantes telles que des capteurs de flamme, d'obstacle, de température et d'humidité, le ventilateur, le SIM5320A, l'Arduino Uno et REYAX RYLR896, permet le contrôle et le suivi à distance.

Tout d'abord, l'intégration du transmetteur récepteur REYAX RYLR896 a été fait pour permettre la transmission et la réception des données et des actions en se basant sur le protocole LoRa.

Ensuite, l'intégration de la carte SIM5320A, qui a été mise en place pour permettre la transmission et la réception des données et des actions à travers le GSM, spécifiquement par SMS.

Puis, l'Arduino Uno jouera le rôle du cœur de cette partie en orchestrant le bon fonctionnement du système grâce au micrologiciel développé et intégré, permettant la gestion des différents périphériques de cette partie.

En effet, grâce à la liaison par fil avec ces composantes, l'Arduino Uno va s'occuper de collecter les données reçues des capteurs. Par la suite, en fonction de l'information et en plus du transfert des données, il procédera, soit au lancement automatique des actions tel que le démarrage du ventilateur et l'envoi des SMS d'alerte, soit à l'exécution d'action demandée par l'utilisateur.

Enfin, la transmission et la réception des données entre ce module et le deuxième module se feront sans fil et par l'intermédiaire du REYAX RYLR896.

Les caractéristiques de cette partie sont qu'elle procède à la prise de décision automatique et le lancement d'action en cas de besoin soit en envoyant un ordre aux actionneurs soit en lançant des alertes aux utilisateurs. En plus, elle permet le contrôle et le suivi à travers les SMS en tout temps.

Deuxième module du circuit

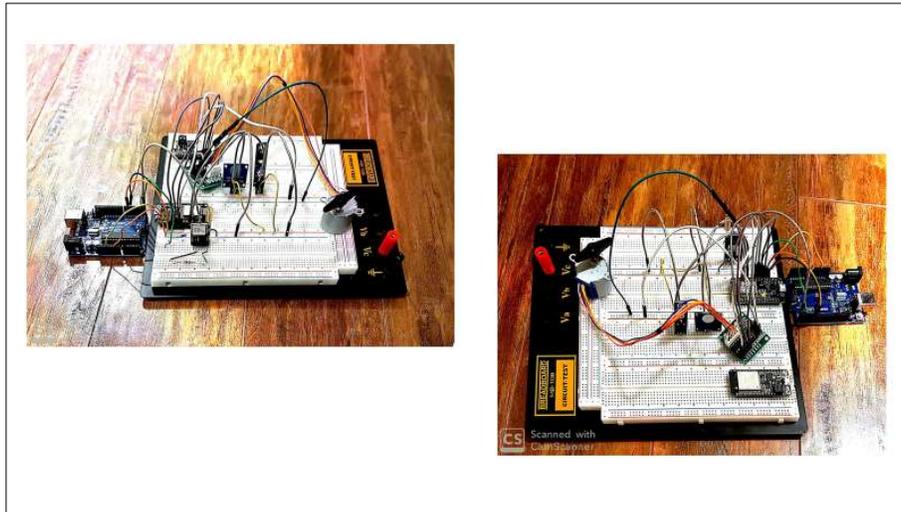


Figure 15. Vue réelle du deuxième module du circuit

Dans la figure 15, cette partie incluant les capteurs de mouvement et tactile, un servomoteur, ESP8266, Arduino Uno, REYAX RYLR896.

D'abord, le REYAX RYLR896 qui est attaché à l'Arduino Uno joue le rôle du pont qui permettra la communication et l'échange entre le premier module du circuit et l'ESP8266 grâce à une liaison par fil entre l'Arduino Uno et l'ESP8266 qui permettra la transmission des données entre ces deux composantes d'un côté, et au protocole LoRa pour l'échange des données avec le premier module de l'autre côté.

Ensuite, les données reçues du premier module vont être transmises à l'ESP8266 et vice versa des données reçues par l'ESP8266, généralement de la part de l'utilisateur sous forme de mise à jour ou des actions à lancer, vont être transmises vers l'autre partie qui est le premier module. En plus, l'ESP8266, qui est lui-même attaché à des capteurs et un servomoteur va être programmé de son côté pour collecter les données ou exécuter des actions.

Puis, grâce au module Wi-Fi de l'ESP8266 pour la connexion à Internet et la communication avec un serveur MQTT qui permettront la transmission et la réception des données à travers le Web en utilisant le protocole MQTT.

Enfin, cette partie jouera le rôle de pont entre le système et le Web soit pour le contrôle et le suivi ou bien le traitement et stockage des données.

Les caractéristiques de cette partie sont que mis à part le contrôle des composantes qui lui sont directement attachées, un échange et une communication avec le premier module du circuit, soit pour le suivi ou le contrôle, seront faits par l'intermédiaire de l'Arduino Uno intégrant le REYAX RYLR896. En plus, la transmission des données vers l'externe, dans le nuage, débute à ce niveau.

Troisième module du circuit

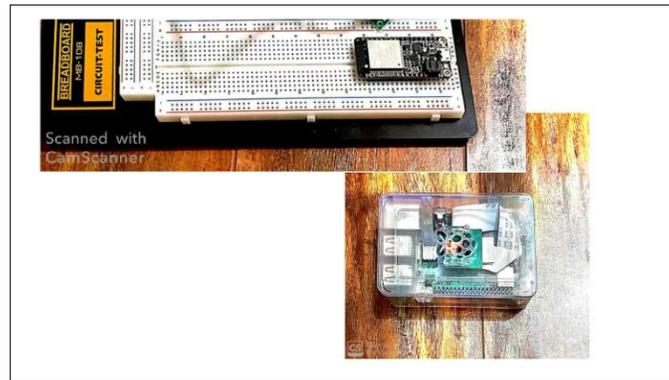


Figure 16. Vue réelle du troisième module du circuit

Cette partie inclura l'ESP32 et un Raspberry Pi 3 Model B comme montré dans la figure 16.

Initialement, notre ESP32 configuré pour jouer le rôle de point d'accès, il sera en état passif. Pas d'interaction tant que le système fonctionne correctement. Cependant, en cas de perte de la connexion à Internet pour la transmission à l'externe, au niveau du deuxième module, c'est là que l'ESP32 passera de l'état passif à l'état actif.

En effet, en cas de perte de la connexion à Internet, le système ne pourra plus effectuer la transmission des données vers le nuage pour le traitement et le stockage. Néanmoins, grâce aux technologies cellulaires le premier module, restera complètement fonctionnel et permettra le contrôle et le suivi soit grâce à l'aspect automatisé soit grâce à la communication GSM, cependant les données seront perdues et ne seront plus stockées dans la base de données.

C'est pour cela, qu'au deuxième module, l'ESP8266 sera programmé à ce qu'en cas de perte de connexion Internet, un basculement automatique de transmission des données vers l'ESP32 sera fait en connectant l'ESP8266 sur le réseau local disponible grâce à l'ESP32. Et dès

le début de réception des données, ce dernier va commencer à transmettre à son tour ces données à travers le Bluetooth vers le Raspberry Pi qui contient une base de données pour leur stockage temporaire en local après traitement.

En effet, à la réception des données, le Raspberry Pi pourra traiter et effectuer le stockage des données reçu. Une fois la connexion avec l'externe est rétablie, une synchronisation entre les données dans le nuage et les données stockées en local sera faite. Ce qui permettra de garder la conformité des données pour permettre un meilleur suivi et le bon fonctionnement tout en évitant, la perte des données.

5.2. Évaluation du système

5.2.1. Transmission de données LoRa

Table 9. Résultat expérimental pour la transmission de données LoRa

Portée (m)	Température		Humidité	
	Échantillon	Valeur	Échantillon	Valeur
< 1	1	23,42	1	32
	2	23,42	2	32
	3	23,42	3	32
	4	23,35	4	32
	5	23,32	5	32
	<= 5	1	23,56	1
2		23,42	2	31
3		23,56	3	32
4		23,56	4	32

	5	23,56	5	32
<= 10	Échantillon	Valeur	Échantillon	Valeur
	1	23,42	1	32
	2	Indéfinie	2	Indéfinie
	3	22,85	3	33
	4	23,56	4	33
	5	23,56	5	32
<= 15	Échantillon	Valeur	Échantillon	Valeur
	1	Indéfinie	1	Indéfinie
	2	21,81	2	34
	3	22,55	3	33
	4	23,45	4	32
	5	23,45	5	32
Moyenne	23		32	

Les résultats du tableau 9 montrent différents échantillons de valeurs de température et d'humidité prises à différentes distances pour la transmission LoRa. Ces valeurs démontrent que malgré la présence d'obstacles dans notre environnement de test, LoRa reste capable de transmettre des données entre les deux extrémités avec un taux de précision acceptable. Le seul inconvénient est que plus la distance est longue, plus il faut attendre la retransmission d'une valeur fiable.

5.2.2. Lancement des alertes

Table 10. Résultat expérimental pour le lancement des alertes

Portée (m)	Feu		Obstacle	
	État	Alertes	État	Alertes
<1				

	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
<= 5	État	Alertes	État	Alertes
	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
<= 10	État	Alertes	État	Alertes
	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Début d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes	Fin d'alerte	Web : déclenchement en moins de 10 secondes SMS : déclenchement en moins de 5 secondes
<= 15	État	Alertes	État	Alertes
	Début d'alerte	Web : déclenchement en moins de 15 secondes SMS : déclenchement en moins de 5 secondes	Début d'alerte	Web : déclenchement en moins de 15 secondes SMS : déclenchement en moins de 5 secondes
	Fin d'alerte	Web : déclenchement en moins de 15 secondes SMS : déclenchement en moins de 5 secondes	Fin d'alerte	Web : déclenchement en moins de 15 secondes SMS : déclenchement en moins de 5 secondes

Les résultats du tableau 10 montrent différents temps de réaction du système aux alertes, prises à différentes distances pour la transmission LoRa. Ces valeurs démontrent que plus la distance est longue, plus il faut de temps pour lancer des alertes sur le Web. En effet, étant donné que les informations sont transmises d'un point à un autre au départ via LoRa puis sur le Web, il est normal de constater ce retard. Cependant, étant donné que la carte SIM5320A est directement connectée à l'Arduino gérant les capteurs de flamme et d'obstacle, le lancement d'alertes par SMS est plus rapide.

5.2.3. Réception des données

Table 11. Résultat expérimental pour la réception des données

Intervalle de temps (s)	Données reçues
0-60	✓
60-120	✓
120-180	✓
Perte de la connexion Internet (déplacer les données vers la base de données locale)	
180-250	Aucune
250-310	✓
310-370	✓
Rétablissement de la connexion Internet	
370-425	Aucune
425-485	✓
485-545	✓

Comme le montre le tableau 11, les données transitent par notre système dans les deux situations avec et sans connexion Internet. Lorsque la connexion Internet est disponible, les données continuent d'être traitées et stockées dans la machine distante (Cloud). Ensuite, nous avons forcé la perte de la connexion Internet, ce qui a conduit notre système à un total de plus de 1 minutes (70 secondes) de perte de données qui représente l'intervalle de temps qu'a pris le système pour passer au stockage local (Edge). Une fois la connexion Internet rétablie, la transmission des données est revenue à la normale avec une latence de commutation de 55 secondes de perte de données. Ce délai pourrait être amélioré en appliquant davantage de techniques d'optimisation telle qu'opter pour une fabrication d'une carte personnalisée et qui sera optimisée selon le besoin au lieu d'utiliser les cartes Arduino commercialisées.

Autre façon d'optimisations qui peuvent être au niveau du code des micrologiciels en appliquant des règles simples d'optimisation du code lors de la programmation, mais qui peuvent améliorer les performances du système telles que la suppression du code inutile, utilisation de types de données plus petits, utiliser des fonctions au lieu de répéter le code, utilisation de variables locales au lieu de variables globales et manipulation directe des ports.

5.2.4. Interaction entre l'utilisateur et le système via les SMS

Table 12. Résultat expérimental pour l'interaction avec l'utilisateur via les SMS

	Interaction	Envoi	Réception	Temps de réponse maximum(s)
Demander le menu	Utilisateur au système	✓		1 - 5
Options du menu	Système à l'utilisateur		✓	1 - 5
Choix d'une option	Utilisateur au système	✓		1 - 5
Réaction à l'option choisie	Système à l'utilisateur		✓	5 - 10
Alertes	Système à l'utilisateur	✓		5 - 10

Le tableau 12 montre le temps de réponse pour l'interaction entre le système et l'utilisateur via des messages texte. Le temps de réponse est d'environ 1 à 10 secondes, ce qui est acceptable et peut être amélioré et reste dépendant du fournisseur de services cellulaires.

5.2.5. Vue d'ensemble des fonctionnalités

Table 13. Vue d'ensemble des fonctionnalités

	Connexion disponible à Internet (Données stockées dans une base de données distante)	Connexion non disponible à Internet (Données stockées dans la base de données locale)
Données stockées dans la base de données	✓	✓
Synchronisation	✓ Déplacer les données de la base de données locale vers la base de données distante	Aucune

Contrôle Web / moniteur / alertes / analytique	✓	Aucune
Contrôle / moniteur / alertes SMS	✓	✓
Détecter un obstacle	✓	✓
Détection tactile	✓	✓
Détecter le feu	✓	✓
Détecter le mouvement	✓	✓
Obtenez la température	✓	✓
Obtenez l'humidité	✓	✓
Contrôle du ventilateur	✓	✓
Contrôle du servomoteur	✓	Aucune
Alertes d'obstacles	✓	✓
Alertes tactiles	✓	Aucune
Alertes de mouvement	✓	Aucune
Alertes d'incendie	✓	✓
Alertes de seuil de température dépassée	✓	✓

Le tableau 13 présente les différentes fonctionnalités disponibles sur le système et leur disponibilité selon l'existence ou non d'une connexion Internet. En effet, le contrôle via le Web sera perdu au cas où la connexion Internet serait perdue tandis que le stockage des données continuerait de s'exécuter dans la base de données locale, car le système continuerait de fonctionner et sera disponible pour consultation une fois la connexion Internet rétablie grâce à notre stratégie de stockage local. Ainsi qu'une partie de la surveillance et du contrôle liés au premier module présenté précédemment ne restera disponible que par l'échange de SMS, car le premier module contient la carte SIM5320A qui permet des interactions utilisateurs / système via SMS

contrairement au deuxième module qui contient la carte ESP8266 qui permet des interactions utilisateurs / système via le Web.

5.3. Résultats

5.2.1. Surveillance et contrôle par SMS



Figure 17. Contrôle du système par SMS

La figure 17 montre les étapes d'interaction avec le système via les SMS comme suit :

1. L'utilisateur envoie un SMS avec le mot clé "MENU".
2. Le système renverra un message contenant 4 options avec le code pour chaque option comme suit:
 - Démarrer code 1 : pour démarrer le ventilateur.
 - Humidité code 2 : Pour connaître le niveau d'humidité.
 - Température code 3 : pour obtenir la température.
 - Arrêt code 4 : pour arrêter le ventilateur.
3. L'utilisateur répond par un SMS contenant le numéro de l'option choisie

4. Le système renverra une réponse soit la valeur demandée soit la confirmation du début d'une action

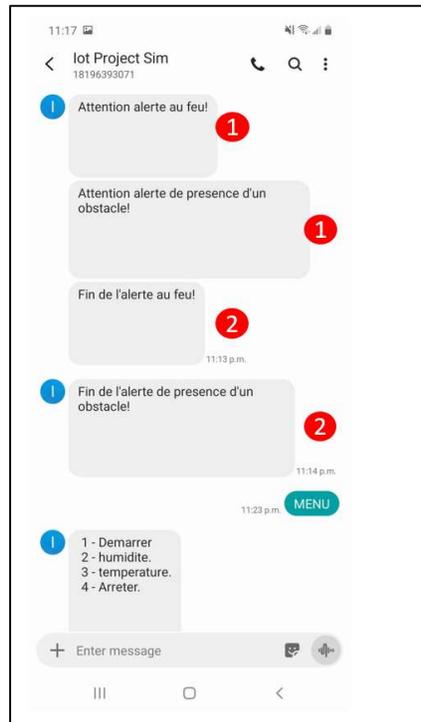


Figure 18. Alertes reçues par SMS

Contrairement à ce qui est présenté sur la figure 17 où l'utilisateur s'interagit avec le système. En cas d'alerte, le système envoie des SMS pour avertir l'utilisateur en cas de début et de fin d'alertes, figure 18.

Comme présenté aux figures 17 et 18, notre système permet le contrôle et le suivi par SMS. En effet, grâce au Sim5320A, nous avons ajouté l'interaction directe avec le système soit pour le suivi de la température ou de l'humidité soit pour le contrôle du ventilateur. De plus, nous avons programmé notre microcontrôleur pour suivre la température et permettre le lancement automatique du ventilateur dans le cas où la température dépasse un seuil défini par les utilisateurs, ainsi que la transmission d'alertes par SMS à l'utilisateur pour l'informer de tout problème tel que la détection de flamme ou la présence d'obstacles.

La définition du seuil de température à ne pas dépasser peut être modifiée par l'utilisateur via l'interface Web.

5.2.2. Surveillance et contrôle via l'interface Web

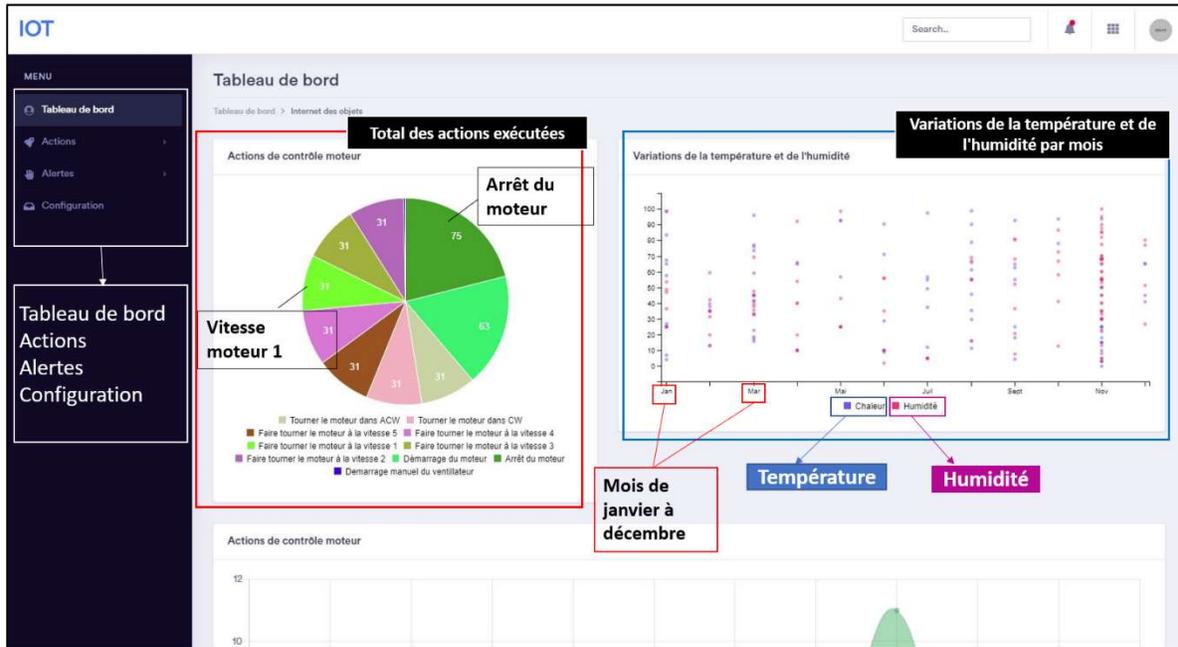


Figure 19. Rapports des états

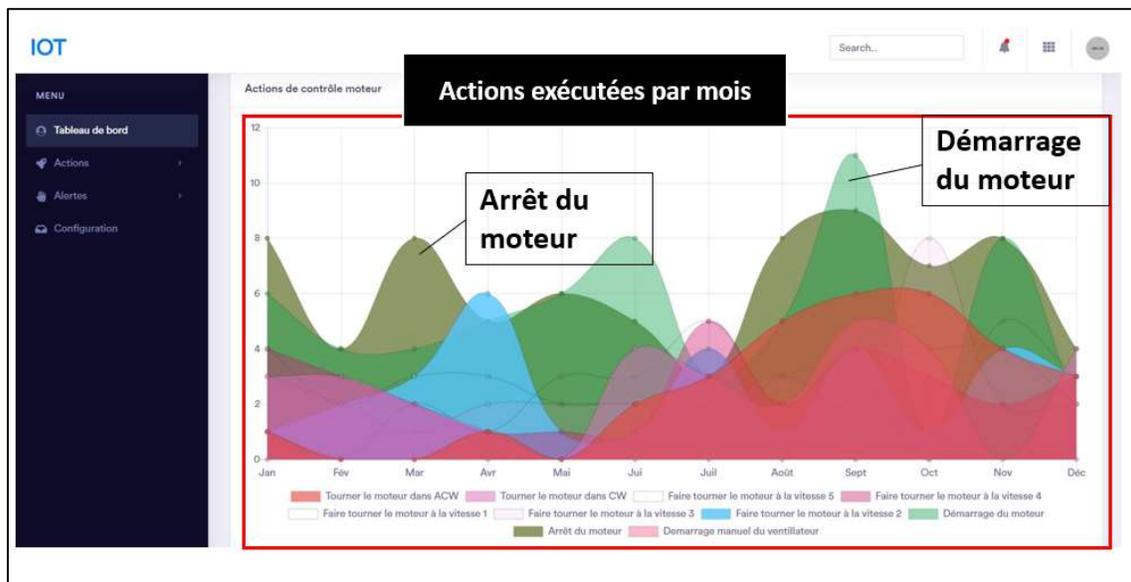


Figure 20. Rapports des actions

Grâce à l'interface Web, figures 19 et 20, nous proposons des statistiques et des rapports pour l'utilisateur tels que les variations de température ou d'humidité dans le temps, le nombre d'actions exécutées soit au lancement automatique soit au lancement manuel ainsi que des alertes grâce aux données stockées.

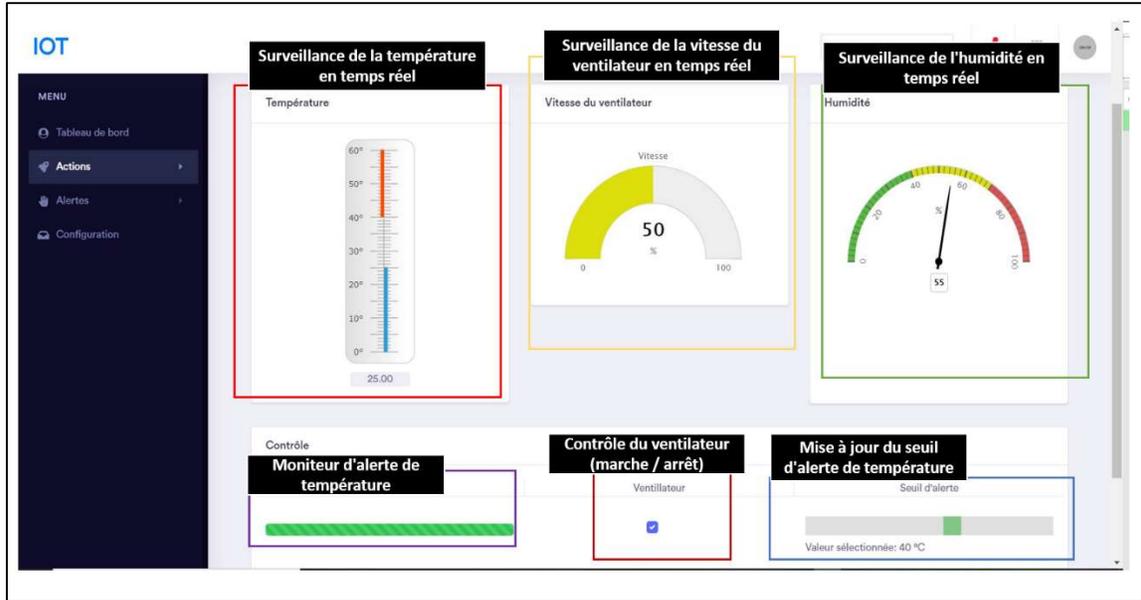


Figure 21. Surveillance et contrôle en temps réel

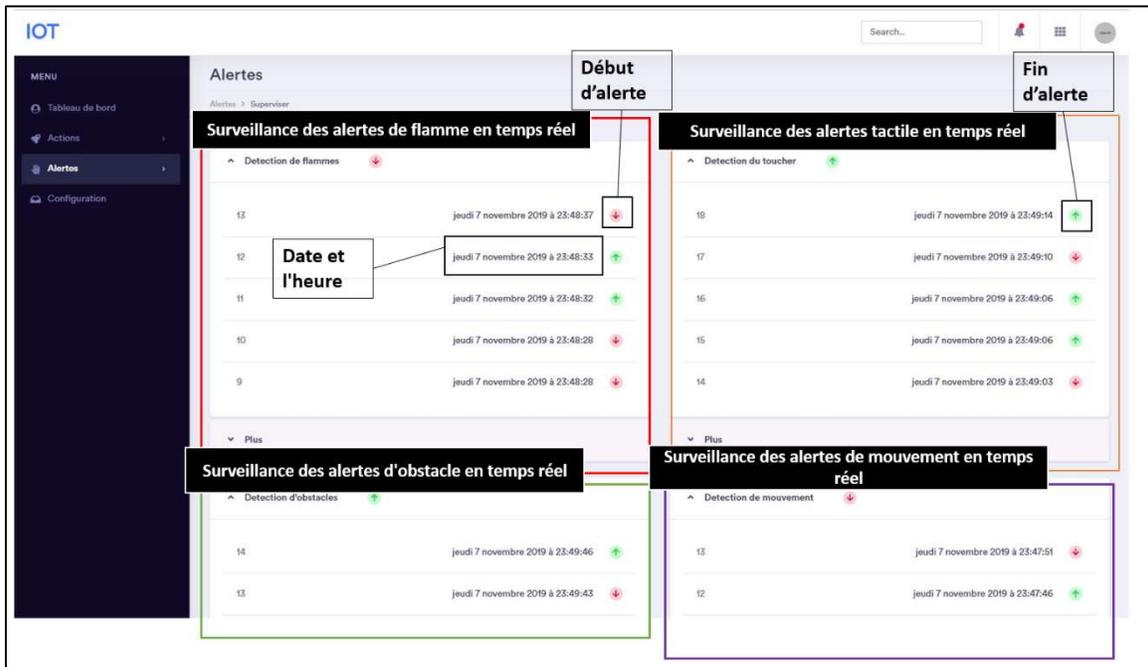


Figure 22. Surveillance des alertes en temps réel

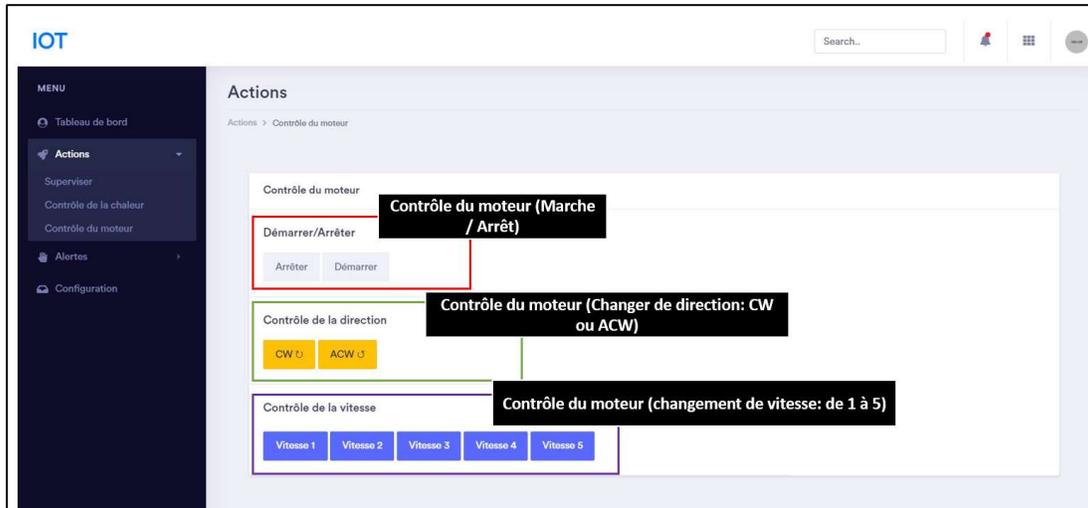


Figure 23. Contrôle des actions

Comme le montrent les figures 21 et 22, nous permettons également à l'utilisateur d'obtenir un suivi en temps réel des changements de température et d'humidité ou un suivi des alertes via la communication avec le serveur MQTT.

De plus, figure 21 et 23, nous proposons le contrôle pour initier des actions telles que le démarrage d'un servomoteur ou d'un ventilateur et la possibilité de mettre à jour les paramètres d'alertes par exemple, le seuil des alertes de température.

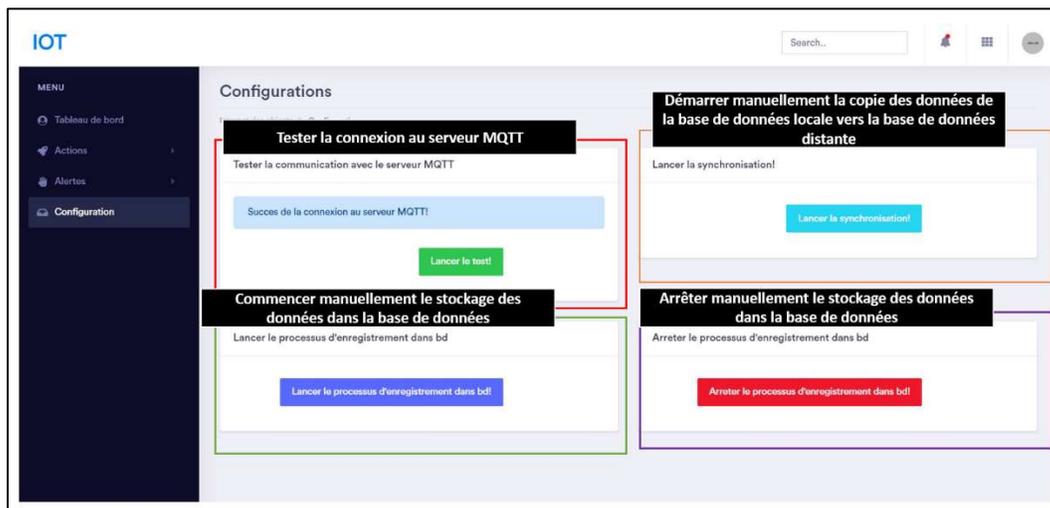


Figure 24. Configuration du système

Enfin, nous offrons la possibilité via une page Web de configuration du système illustrée à la figure 24, la possibilité de choisir de stocker les données dans des bases de données ou non, de

tester la connexion avec le serveur MQTT et de démarrer manuellement la synchronisation pour récupérer les données stockées dans la base de données locale et la transmission à la base de données distante.

Chapitre 6

Conclusion

L'Internet des objets n'est pas tant une chose qu'un concept. C'est un concept qui nous permet d'instrumenter notre monde avec des capteurs et de réagir de manière significative aux données provenant de ces capteurs.

La connectivité est le cœur et l'élément indispensable dans l'IdO et sa disponibilité est primordiale pour le bon fonctionnement des systèmes. Cependant, malgré l'émergence de nouveau protocole de connectivité, l'échec de la perte de cette dernière restera un défi à surmonter.

En plus, au fur et à mesure que des millions d'appareils se connectent, l'IdO va déclencher un afflux massif de données volumineuses, l'analyse et le traitement de ces données le plus efficacement possible avec le plus de rapidité possible représentent un autre défi à surmonter.

Dans ce travail, la mise en œuvre de la combinaison de plusieurs protocoles de connectivité dans un seul système ainsi que le stockage des données en périphérie et la synchronisation vers le nuage ont été proposé comme solution pour assurer la continuité et la fiabilité de la collecte des données, du traitement et du stockage dans le but d'assurer sa conformité pour donner lieu à de meilleures analyses et interprétations.

En effet, après avoir effectué l'étude des différentes composantes d'une plateforme IdO ainsi que les protocoles de communication et transmission de données IdO les plus populaires, nous avons proposé la mise en place d'un système hybride composé de plusieurs technologies de connectivités.

Par la suite, nous avons inclus l'intégration de l'informatique en périphérie (Edge Computing) dans notre solution comme moyen de sauvegarde pour le stockage des données en local et évité la perte des données qui par la suite seront transmises vers le nuage (Cloud).

Puis, nous sommes passés à la conception et la mise place de notre solution en commençant par la réalisation du circuit, l'intégration des différentes composantes, tester le fonctionnement et

l'interaction globale entre les différents périphériques ainsi que la configuration des serveurs nécessaires et le développement de l'application Web réactive permettant ainsi d'offrir une interface graphique à l'utilisateur pour le suivi et contrôle en temps réel ainsi que la consultation des rapports et statistiques pour le suivi des états.

Pour finir, nous avons démontré l'intégration efficace de plusieurs protocoles de connectivité dans un seul système IdO pour assurer la continuité et la fiabilité de la collecte, du traitement et du stockage des données dans le but d'assurer la conformité des données pour une meilleure analyse et interprétation. De plus, ce travail prouve l'efficacité de l'intégration de différentes technologies telles que l'informatique en périphérie (Edge Computing) et l'informatique en nuage (Cloud Computing) pour gérer et faire face à l'énorme quantité de données transmises qui seront ensuite traitées pour donner aux utilisateurs des statistiques et des rapports solides ainsi qu'un suivi de l'état et des prévisions.

Dans le cadre des travaux futurs, ce travail peut être évolué en intégrant les techniques des mégadonnées plus avancées et l'adoption de l'intelligence artificielle en appliquant des algorithmes d'apprentissage machine pour bâtir des modèles qui peuvent être utilisés pour donner de meilleurs résultats soit pour la maintenance du système soit pour fournir des prédictions utiles pour l'utilisateur.

Bibliographie

- [1] Sharon Varghese (2016), *Application of IoT to improve the lifestyle of differently abled people*, IOSR Journal of Computer Engineering (IOSR-JCE), pages 29-34.
- [2] Deloitte (2018), *The Future of Connectivity in IoT Deployments*.
- [3] Cloud Standards Customer Council (2016), *Cloud Customer Architecture for IoT*.
- [4] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic et Marimuthu Palaniswami (2013), *Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions*, Future Generation Computer Systems (Vol.29), Pages 1645-1870, [doi : 10.1016/j.future.2013.01.010] .
- [5] Keyur K Patel et Sunil M Patel (2016), *Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges*, International Journal of Engineering Science and Computing (IJESC)(Vol.6), [doi : 10.4010/2016.1482].
- [6] Kevin Ashton (2009), *That 'Internet of Things' Thing*, RFID Journal, [Disponible en ligne : <https://www.rfidjournal.com/articles/view?4986>], [Date de la dernière consultation : 15 Novembre 2019].
- [7] Cisco Internet Business Solutions Group (IBSG) (2011), *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*.
- [8] IBM (2016), *What is the Internet of Things?*
- [9] Dr. John Barrett, *The Internet of Things*, Nimbus Centre for Embedded Systems Research, Cork Institute of Technology.
- [10] Embitel (2018), *Unraveling the Story of Evolution of IoT and Its Rapid Adoption*, [Disponible en ligne : <https://www.embitel.com/blog/embedded-blog/unraveling-the-story-of-evolution-of-iot-and-its-rapid-adoption>], [Date de la dernière consultation : 15 Novembre 2019].

- [11] Matt Rosoff (2015), *Why is tech getting cheaper?*, World economic forum, [Disponible en ligne : <https://www.weforum.org/agenda/2015/10/why-is-tech-getting-cheaper/>], [Date de la dernière consultation : 15 Novembre 2019].
- [12] Keith D. Foote (26 Mars 2019), *A Brief History of Machine Learning*, DATAVERSISTY, [Disponible en ligne : <https://www.dataversity.net/a-brief-history-of-machine-learning/>], [Date de la dernière consultation : 15 Novembre 2019].
- [13] Prof. Sudip Misra, *Introduction to Internet of Things*, Department of Computer Science & Engineering Indian Institute of Technology.
- [14] Cisco Internet Business Solutions Group (IBSG) (2011), *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*.
- [15] Tulasi Bomatpalli et Girish J Vemulkar (2016), *Blending iot and big data analytics*, International Journal Of Engineering Sciences & Research Technolog, Pages 192-196, [doi : 10.5281/zenodo.48868].
- [16] EDUCBA (2018), *Artificial Intelligence vs Human Intelligence*, [Disponible en ligne : <https://www.educba.com/artificial-intelligence-vs-human-intelligence/>], [Date de la dernière consultation : 15 Novembre 2019].
- [17] Gonçalo J. F. Carnaz et Vitor Nogueira (2016), *An Overview of IoT and Healthcare*, Université de Évora École des sciences et technologies.
- [18] Timothy Malche et Priti Maheshwary (2017), *Internet of Things (IoT) for building Smart Home System*, IEEE International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), SCAD Institute of Technology, Palladam, [doi : 10.1109/I-SMAC.2017.8058258].
- [19] Eunil Park, Angel P. del Pobil et Sang Jib Kwon (2018), *The Role of Internet of Things (IoT) in Smart Cities: Technology Roadmap-oriented Approaches*, MDPI Open Access Journals (Sustainability), [doi : 10.3390/su10051388].
- [20] Duncan McFarlane, *Industrial Internet of Things Applying IoT in the Industrial Context*, Engineering and Physical Sciences Research Council (EPSRC).

- [21] Anni Junnila (2018), *How IoT Works*, Dans Trackino, [Disponible en ligne : <https://trackinno.com/2018/08/09/how-iot-works-part-4-user-interface/how-iot-works-summary-001/>], [Date de la dernière consultation : 15 Novembre 2019].
- [22] Karandeep Kaur (2018), *A Survey on Internet of Things – Architecture, Applications, and Future Trends*, IEEE, First International Conference on Secure Cyber Computing and Communication (ICSCCC), pages 581-583.
- [23] Deepti Sehrawat and Nasib Singh Gill (2019), *Smart Sensors: Analysis of Different Types of IoT Sensors*, IEEE, 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pages 523-528.
- [24] Ray P.P. (2016), *A survey on Internet of Things architectures*, Journal of King Saud University - Computer and Information Sciences (Vol.30).
- [25] Jeffrey Lee (2019), *How to Choose the Right IoT Connectivity Protocol for Your Connected Device*, [Disponible en ligne : <https://blog.particle.io/2019/03/06/iot-connectivity/>], [Date de la dernière consultation : 15 Novembre 2019].
- [26] Tara Salman et Raj Jain (2017), *Networking Protocols and Standards for Internet of Things*, Internet of Things and Data Analytics Handbook, pages 215-238, [doi : 10.1002/9781119173601.ch13].
- [27] Tommaso Polonelli, Davide Brunella, Alberto Girolami, Gerardo Nahuel Demmi, Luca Benini (2019), *A multi-protocol system for configurable data streaming on IoT healthcare devices*, IEEE, IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI), pages 112-117.
- [28] Tara Salman et Raj Jain (2017), *Networking Protocols and Standards for Internet of Things*, Internet of Things and Data Analytics Handbook, pages 215-238.
- [29] Ravi K Kodali, Aditya Valdas, *MQTT Implementation of IoT based Fire Alarm Network*, 2018, International Conference on Communication, Computing and Internet of Things (IC3IoT), pages 143-146.

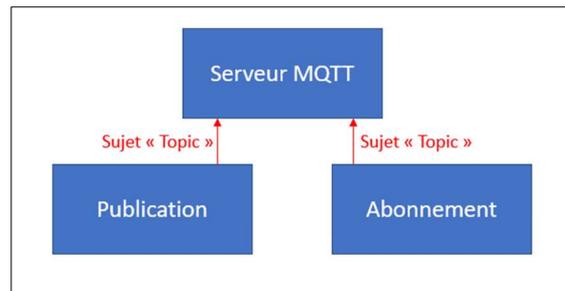
- [30] Soumyalatha Naveen, Manjunath R Kounte, *Key Technologies and challenges in IoT Edge Computing*, 2019, Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pages 61-65.
- [31] Industrial Internet Consortium (2018), *Introduction to Edge Computing in IIoT*.
- [32] Jiang Rui, Sun Danpeng, *Architecture Design of the Internet of Things based on Cloud Computing*, 2015, Seventh International Conference on Measuring Technology and Mechatronics Automation, pages 206-209.
- [33] Hourieh KHODKARI et Saied Ghazi MAGHREBI (2016), *Necessity of the integration Internet of Things and cloud services with quality of service assurance approach*, Bulletin de la Société Royale des Sciences de Liège (Vol.85), pages 434 – 445.
- [34] K. Jayaram, K. Janani, R. Jeyaguru, R. Kumaresh, N. Muralidharan, *Forest Fire Alerting System With GPS Co-ordinates Using IoT*, 2019, 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pages 488-491, 2019.
- [35] Ahmed Imteaj, Tanveer Rahman, Muhammad Kamrul Hossain, Mohammed Shamsul Alam and Saad Ahmad Rahat, *An IoT based Fire Alarming and Authentication System for Workhouse using Raspberry Pi 3*, 2017, International Conference on Electrical, Computer and Communication Engineering (ECCE), pages 899- 904.
- [36] K.M.Gaikwad, Ahmed Quadri, Shelar Akshada, Zagade Reshma, *Fire Monitoring and Control System*, 2016, International Research Journal of Engineering and Technology (IRJET).
- [37] Mohd Faris Mohd Fuzi, Alif Faisal Ibrahim, Mohammad Hafiz Ismail, Nur Syakira Ab Halim, HOME FADS: *A Dedicated Fire Alert Detection System Using ZigBee Wireless Network*, 2014, IEEE, 5th Control and System Graduate Research Colloquium, pages 53-58.
- [38] S. Tanwar_, P. Pately, K. Patelz, S. Tyagix, N. Kumar, M. S. Obaidat, *An Advanced Internet of Thing based Security Alert System for Smart Home*, 2017, International Conference on Computer, Information and Telecommunication Systems (CITS).

- [39] Adil Hamid Malik, Aaqib jalal, Bilal Ahmed Parray, Meena kohli (2017), *Smart City IoT Based Weather Monitoring System*, 2017 International journal of Engineering Science and Computing (IJESC) Volume 7 Issue no.5
- [40] Vijay Vaishnavi, Bill Kuechler, et Stacie Petter (2019), *Design Science Research in Information Systems*, [Disponible en ligne : <http://www.desrist.org/design-research-in-information-systems/>], [Date de la dernière consultation : 15 Novembre 2019].
- [41] FUTURA TECH (2018), *System on a chip*, [Disponible en ligne : <https://www.futura-sciences.com/tech/definitions/soc-system-on-chip-14843/>], [Date de la dernière consultation: 15 Novembre 2019].
- [42] Arduino (2018), *What is Arduino ?* [Disponible en ligne : <https://www.arduino.cc/>], [Date de la dernière consultation : 15 Novembre 2019].
- [43] The Raspberry Pi Foundation (2018), *Raspberry Pi 3 Model B*, [Disponible en ligne : <https://www.raspberrypi.org/>], [Date de la dernière consultation : 15 Novembre 2019]
- [44] circuits4you.com (2018), *ESP32 vs ESP8266*, [Disponible en ligne : <https://circuits4you.com/2019/03/02/esp32-vs-esp8266/>], [Date de la dernière consultation : 15 Novembre 2019]
- [45] TinySine, *3G/GPRS/GSM Shield for Arduino with GPS - American version SIM5320A*.
- [46] Reyax Technology corporation LTD (2018), *RYLR896, UART Interface 868/915MHz Lora Antenna Transceiver Module, Datasheet*.
- [47] Fisher T. (2019), *What Is Firmware?*, [Disponible en ligne : <https://www.lifewire.com/what-is-firmware-2625881>], [Date de la dernière consultation : 15 Novembre 2019].
- [48] Fraden Jacob (2010), *Handbook of Modern Sensors: Physics, Designs, and Applications, Fourth Edition*.
- [49] Mosquitto (2018), *Eclipse Mosquitto™ An open source MQTT broker*, [Disponible en ligne : <https://mosquitto.org/>], [Date de la dernière consultation : 15 Novembre 2019]

Annexe A

Principe de fonctionnement du MQTT

Le MQTT se base sur la publication/abonnement où plusieurs clients se connectent à un serveur unique appelé serveur MQTT, pour publier des informations ou s'abonner pour la réception des informations. Contrairement au principe du client/serveur utilisé sur le Web.



Les sujets « Topic »

L'une des notions fondamentales dans la communication avec le serveur pour la publication et l'abonnement ce sont les sujets « Topic ».

En effet, le client s'abonne pour s'enregistrer auprès du serveur MQTT sur des sujets, à fin d'être notifié des publications reçues dans le serveur sur ce même sujet.

Généralement, un standard sur la forme des sujets est d'être sous la forme de mots séparés par des barres obliques « / ».

Exemple de forme de sujet

- /capteur/salon/temperature
- /capteur/chambre/temperature

On peut s'abonner à un ensemble de sujets en utilisant des « Wildcards » (# ou +).

Exemple de forme de sujet avec des « Wildcards »

- /capteur/salon/#

- /capteur+/temperature

Exemple de cas 1

- Un **client 1** s'abonne auprès du serveur avec le sujet /capteur/salon/temperature
 - Un **client 2** publie auprès du serveur avec le sujet /capteur/salon/temperature **X**
 - Le **client 1** sera notifié de la réception de l'information du **client 2** et obtiendra la valeur **X**.
- ⇒ Exemple de situation : faire le suivi de la température d'une chambre.
- **X est la valeur associée au sujet envoyé.**

Exemple de cas 2

- Un **client 1** s'abonne auprès du serveur avec le sujet /capteur+/temperature
 - Un **client 2** publie auprès du serveur avec le sujet /capteur/salon/temperature **X**
 - Un **client 2** publie auprès du serveur avec le sujet /capteur/chambre/temperature **X**
 - Le **client 1** sera notifié de la réception de l'information du **client 2** et obtiendra la valeur pour le salon et la chambre.
- ⇒ Exemple de situation : faire le suivi de la température d'une chambre et du salon.
- **X est la valeur associée au sujet envoyé.**

Exemple de cas 3

- Un **client 1** s'abonne auprès du serveur avec le sujet /capteur/salon/#
 - Un **client 2** publie auprès du serveur avec le sujet /capteur/salon/temperature **X**
 - Un **client 2** publie auprès du serveur avec le sujet /capteur/salon/humidite **X**
 - Le **client 1** sera notifié de la réception de l'information du **client 2** et obtiendra la valeur pour la température et l'humidité dans le salon.
- ⇒ Exemple de situation : faire le suivi de la température et de l'humidité d'un salon.
- **X est la valeur associée au sujet envoyé.**

Le client MQTT

Les clients ou bien clients MQTT représentent tous ceux qui peuvent publier ou s'abonner au serveur MQTT.

En effet, le client MQTT est tout périphérique qui exécute une bibliothèque MQTT et se connecte à un serveur MQTT à travers le réseau. Il peut être sous la forme d'un micro périphérique (microordinateur, etc.) comme il peut être un ordinateur puissant.

⇒ Tout périphérique qui communique avec un serveur peut être appelé un client MQTT.

Les bibliothèques de clients MQTT sont disponibles pour une grande variété de langages de programmation. Par exemple, Android, Arduino, C, C ++, C #, Go, iOS, Java, JavaScript et .NET.

Le serveur MQTT

Le serveur est le cœur de tout le processus de publication/abonnement. Selon l'implémentation, il peut gérer jusqu'à des milliers de clients MQTT connectés simultanément.

Il est chargé de recevoir tous les messages, de filtrer les messages, de déterminer qui est abonné à chaque message et d'envoyer le message à ces clients abonnés.

Le serveur peut détenir également les sessions de tous les clients, y compris les publications/abonnement, et les messages manqués, ainsi que gérer l'authentification et l'autorisation des clients.

La publication

Un client MQTT peut publier des messages dès qu'il se connecte à un serveur. MQTT utilise un filtrage par sujet des messages sur le serveur. D'où l'importance des sujets.

En effet, chaque message doit contenir un sujet que le serveur peut utiliser pour transférer le message aux clients intéressés. Le client expéditeur (éditeur) décide s'il souhaite envoyer des données binaires, des données texte ou même des informations complètes. XML ou JSON.

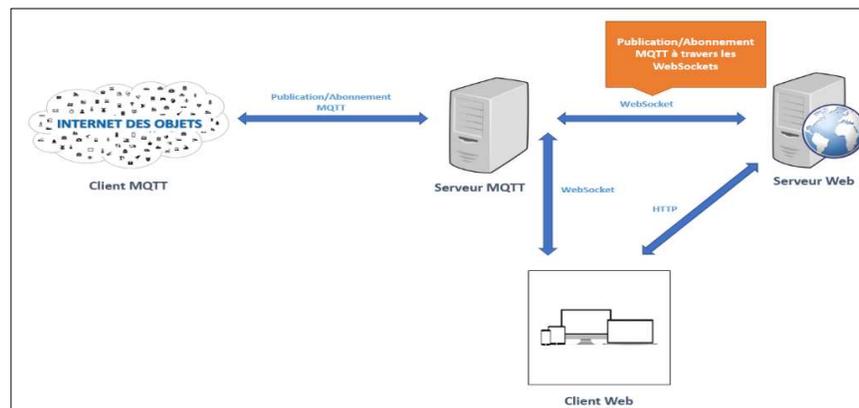
L'abonnement

Publier un message n'a aucun sens si personne ne le reçoit jamais. En d'autres termes, s'il n'y a aucun client qui s'abonne au sujet des messages. Pour recevoir des messages sur des sujets d'intérêt, le client envoie un message d'abonnement au serveur MQTT. Ce message d'abonnement est très simple, il contient un identifiant de paquet unique et une liste d'abonnement.

MQTT et les WebSockets

Le besoin de l'intégration avec d'autres serveurs tel que les serveurs Web est inévitable. En effet, dans le cas de scénario où l'on veut offrir une interface Web pour le suivi et le contrôle des objets à distance, un serveur Web est requis pour pouvoir exécuter du PHP, etc.

Dans ce cas-là, une possibilité est de pouvoir utiliser un serveur Web pour l'interface HTTP, et de configurer un tunnel de connexions WebSocket avec le serveur MQTT.



Avec l'intégration des WebSockets, tous les navigateurs peuvent jouer le rôle de client MQTT.

En effet, grâce à l'abonnement /publication auprès du serveur MQTT, les messages peuvent être reçus en temps réel sur le navigateur lorsqu'un événement se produit, tant que le client s'est abonné au bon sujet.

Qu'est-ce que le WebSocket ?

WebSocket est un protocole permettant de créer un canal bidirectionnel rapide entre un navigateur Web et un serveur. WebSocket surmonte les limitations avec HTTP pour permettre des communications à faible latence entre un utilisateur et un service Web.

Alors que la vitesse d'Internet continue d'augmenter, les utilisateurs commencent à voir Internet comme une source d'informations dynamiques en temps réel. À l'instar des applications de bureau, le Web s'appuyait généralement sur l'utilisateur pour effectuer une action avant que le service puisse traiter et fournir des informations. Bien que cela fonctionne, cela crée une expérience statique et empêche les utilisateurs de s'engager réellement dans le contenu.

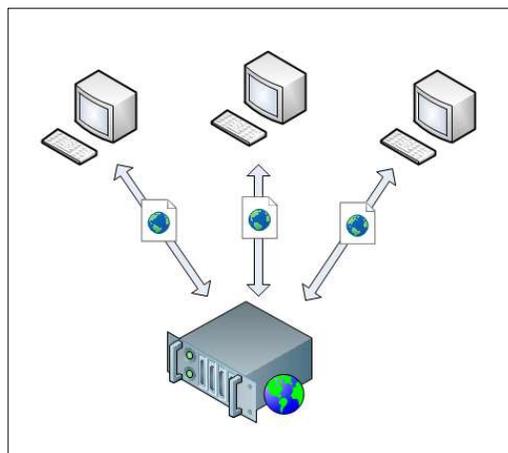
WebSockets ajoute une véritable force au Web en permettant aux sites Web de mettre à jour du contenu sans attendre l'utilisateur. Contrairement aux autres techniques qui s'appuient sur le protocole HTTP, le protocole WebSocket crée une véritable connexion permanente entre l'utilisateur et le service Web, permettant ainsi aux informations de circuler facilement entre les deux ordinateurs d'extrémité.

WebSocket fournit une connexion persistante entre un client et un serveur que les deux parties peuvent utiliser pour commencer à envoyer des données à tout moment.

Serveur Web

Un serveur Web est un ordinateur qui exécute des sites Web. C'est un programme informatique qui distribue les pages Web au fur et à mesure de leur demande. L'objectif de base du serveur Web est de stocker, traiter et fournir des pages Web aux utilisateurs. Cette intercommunication est effectuée à l'aide du protocole HTTP.

Ces pages Web sont principalement des contenus statiques comprenant des documents HTML, des images, des feuilles de style, des tests, etc. Outre un serveur HTTP, un serveur Web prend également en charge les protocoles SMTP et FTP pour l'envoi par courrier électronique et le transfert et stockage de fichiers.



Raspberry Pi comme serveur Web

Le Raspberry Pi est un ordinateur monocarte polyvalente. En raison de sa petite taille et de sa faible consommation énergétique, le Raspberry Pi constitue un serveur Web idéal. En effet,

idéal pour les sites Web de petite taille ne nécessitant ni la capacité ni la puissance de traitement côté serveur d'un ordinateur plus puissant. Le Raspberry Pi offre un moyen simple d'héberger un site Web personnel ou professionnel.

Cependant, pour bénéficier des avantages du Raspberry Pi comme serveur Web, on a recours à l'installation de l'un des logiciels les plus populaires et le plus utilisés qui est Apache.

Apache

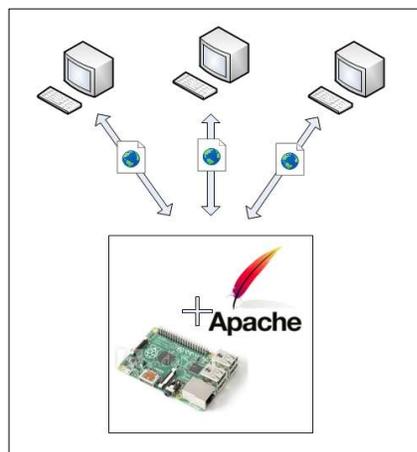
Apache est un serveur Web extrêmement important. Développé par Apache Software Foundation, il s'agit d'un serveur open source qui prend en charge environ 37% des 100 millions de sites Web et 43% des sites Internet.

C'est un serveur HTTP multi-plate-forme qui peut envoyer des fichiers HTML via HTTP. Avec des ressources supplémentaires telles que PHP, Apache peut être utilisé pour créer des pages dynamiques avec des langages tels que PHP.

Apache sur Raspberry Pi

Apache est le serveur Web le plus populaire et peuvent s'installer sur le Raspberry Pi pour lui permettre de servir des pages Web. À lui seul, Apache peut servir des fichiers HTML via HTTP. Avec des modules supplémentaires, il peut servir des pages Web dynamiques utilisant des langages de script tels que PHP.

En effet, Apache est un serveur Web open source. Utilisé avec des langages tels que PHP, il devient encore plus robuste et performant.



Raspberry Pi comme serveur MQTT

MQTT est le protocole de choix pour les applications machine à machine et Internet des objets. De nos jours, il existe plusieurs serveurs basés sur le nuage (Cloud). Cependant, grâce aux ordinateurs monocarte tel que Raspberry Pi, il est possible de configurer un serveur MQTT personnel et profiter des multiples avantages que peut avoir la configuration d'un serveur MQTT local, telle que la sécurité, la flexibilité, la fiabilité, une faible latence, la rentabilité, une meilleure implémentation de la qualité de service, etc.

En effet, le Raspberry Pi dispose de suffisamment de puissance de calcul pour exécuter et fonctionner comme un serveur MQTT personnel, capable de répondre à la plupart des besoins personnels en MQTT. Pour ce faire, il faut installer un serveur MQTT tel que Mosquitto et le configurer.

Annexe B

Implémentation

Exemple de code pour la communication entre l'ESP8266, serveur MQTT et l'interface Web

- ✓ Importation des bibliothèques

```
/* Importation des bibliothèques nécessaires */
#include <ESP8266WiFi.h> // Pour connecter ESP8266 au wifi
#include <PubSubClient.h> // Pour la publication et l'abonnement au serveur MQTT
#include "DHT.h" // Pour le capteur de la température et l'humidité
```

- ✓ Définition des ports

```
/* Définition des ports */
#define DHTTYPE DHT11 //Définition du type de capteur
#define DHTPIN 5 //Définition du port du capteur

const int fanPin = 4; //Définition du port pour le ventilateur
const int ledRedPin = 0; //Définition du port pour la lampe rouge
const int ledGreenPin = 2; //Définition du port pour la lampe verte
```

- ✓ Déclaration des variables

```
/* Déclaration des variables utiles */
long lastMsgCheckedTime = 0; //Temps de lecture du dernier message en milliseconde
float lastTemperatureValue = 0; //Enregistrer la dernière valeur de température reçue
bool forceStart = false; //Vérifiez si l'alerte force le système à démarrer le ventilateur et le voyant rouge (vrai si oui sinon faux)
float maxTempAlert = 40; //Seuil de température maximale définie avant le démarrage de l'alerte
```

- ✓ Déclaration des objets

```
/* Déclaration / Instanciation d'objets */
DHT dht(DHTPIN, DHTTYPE); // Création d'un objet DHT pour manipuler le capteur de température et humidité

const char* ssid = "testmqtt"; //Définir le nom du point d'accès pour se connecter SSID
const char* password = "testmqtttest"; //Définir le mot de passe pour se connecter
IPAddress mqtt_server(192, 168, 50, 10); // Création d'un objet IPAddress (adresse IP du serveur MQTT)

WiFiClient espClient; // Création d'un objet WiFiClient pour se connecter
PubSubClient client(espClient); // Création d'un objet PubSubClient pour interagir avec le serveur MQTT
```

✓ Fonction pour effectuer la connexion au Wi-Fi

```
/*
 * Cette fonction est utilisée pour configurer la connexion au point d'accès
 * en utilisant le ssid et le mot de passe définis
 */
void setup_wifi()
{
    delay(1000);
    Serial.print("Connecting to ");
    Serial.println(ssid);
    // Tentative de connexion
    WiFi.begin(ssid, password);
    /* En cas d'échec de connexion : afficher l'état */
    while (WiFi.status() != WL_CONNECTED) {
        delay(5000);
        Serial.printf("Connection status: %d\n", WiFi.status());
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
```

✓ Fonction pour l'écoute des messages publiés dans le serveur MQTT

En cas de réception et traitement d'un message, on publie un message au serveur MQTT pour notifier la réception et le traitement de l'action dans le but de synchroniser les informations avec notre site Web.

```
/* Cette fonction est utilisée pour gérer le message reçu (écouter) du serveur MQTT
 */
void callback(char* topic, byte* payload, unsigned int length)
{
    /* 3 types de sujets:
     * Les messages dont le sujet (topic) est lié à la valeur de température maximale qui peut être choisie par l'utilisateur
     * Les messages dont le sujet (topic) est lié à l'action pour démarrer le ventilateur et la lampe rouge
     * Les messages dont le sujet (topic) est lié à l'action pour arrêter le ventilateur et la lampe rouge */
    char result;
    Serial.print("Message reçu: [");
    Serial.print(topic);
    Serial.print("] ");
    // Lorsque le sujet (topic) a été reçu, nous vérifions le type de sujet
    // Pour mettre à jour le seuil de température maximale ou pour une action effectuée sur le ventilateur et la lampe
    if (String(topic) == "client/MaxTemp") {
        // Si l'utilisateur sélectionne une nouvelle valeur pour le seuil de température à ne pas dépasser avant le lancement d'alerte, mettre à jour la nouvelle valeur définie
        payload[length] = '\0';
        maxTempAlert = atof((char*)payload);
        Serial.print("Result: [");
        Serial.print(maxTempAlert);
        Serial.print("] ");
    }
    // Une fois le message reçu, nous envoyons une réponse (publier sur MQTT) pour confirmer le message reçu
    client.publish("client/MaxTempReceived/", "1");
}
else {
    // Si le sujet consiste à effectuer une action sur le ventilateur et la lampe, nous vérifions s'il faut démarrer le mode d'alerte (ventilateur allumé) ou arrêter (lampe verte allumée)
    if (payload[0] == 49) {
        Serial.println("Action ----> Allumer.");
        Serial.println();
        digitalWrite(fanPin, HIGH);
        client.publish("client/fanspeed/", "100");
        userStartFan = true;
    }
    else {
        Serial.println("Action ----> Eteindre.");
        Serial.println();
        digitalWrite(fanPin, LOW);
        client.publish("client/fanspeed/", "0");
        userStartFan = false;
    }
}
// Une fois le message reçu, nous envoyons une réponse (publier sur MQTT) pour confirmer le message reçu
client.publish("client/received/", "1");
}
```

✓ Fonction pour la reconnexion au serveur MQTT et l'inscription à l'écoute des messages

```
/* Cette fonction permet de gérer la connexion au serveur MQTT en cas de perte */
void reconnect()
{
  //Boucle : Lorsqu'on se déconnecte, une tentative de reconnexion est effectuée jusqu'à ce que nous soyons reconnectés
  while (!client.connected()) {
    // Serial.print("Attempting MQTT connection...");
    // Créer un ID client aléatoire
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Tentative de connexion
    // client.connect(clientId.c_str());
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      //Une fois connecté au serveur MQTT, nous nous abonnons au sujet de l'action et de la valeur de température maximale
      client.subscribe("client/action/");
      client.subscribe("client/MaxTemp/");
    }
    else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Attendre 5 secondes avant de réessayer
      delay(5000);
    }
  }
}
```

✓ Initialisation de notre système

- Définition des ports en entrée et sortie
- Connexion au réseau Wi-Fi

```
/* Cette fonction est utilisée pour initialiser notre système*/
void setup()
{
  Serial.begin(115200);

  // Initialiser l'objet dht (capteur de température et d'humidité)
  dht.begin();

  // Définir les port
  pinMode(fanPin, OUTPUT);
  pinMode(ledRedPin, OUTPUT);
  pinMode(ledGreenPin, OUTPUT);

  // Initialiser l'état
  digitalWrite(fanPin, LOW);
  digitalWrite(ledRedPin, LOW);
  digitalWrite(ledGreenPin, HIGH);

  // Connexion
  setup_wifi();
  //Définir l'IP et le port du serveur MQTT
  client.setServer(mqtt_server, 1883);
  //Définissez la fonction d'écoute sur le serveur MQTT
  client.setCallback(callback);

  Serial.print(" Starting... ");
}
```

- ✓ Fonction de traitement en boucle du système
 - Vérification de la température
 - Lancement des alertes
 - Publication de la température

```

/* Fonction de bouclage du système pour mettre à jour les valeurs et garder un suivi de l'état */
void loop()
{
  char msgBuffer[20];

  /* Essayer de se reconnecter au serveur MQTT si la connexion est perdue */
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  long now = millis();
  // Lire la valeur du capteur DHT11 toutes les 5 secondes: Utiliser la variable lastMsgCheckedTime qui contient la valeur de la dernière vérification de la température
  // pour vérifier toutes les 5 secondes
  if (now - lastMsgCheckedTime > 5000) {
    lastMsgCheckedTime = now;
    float temperatureValue = dht.readTemperature();
    float humidityValue = dht.readHumidity();

    // Vérifier si la valeur de température du capteur dht n'est pas vide: S'il est vide, nous gardons la dernière valeur stockée, sinon mettre à jour la nouvelle valeur
    if (!isnan(temperatureValue)) {
      lastTemperatureValue = temperatureValue;
    }
    else {
      temperatureValue = lastTemperatureValue;
    }

    // Vérifier si la valeur de l'humidité du capteur dht n'est pas vide: S'il est vide, nous gardons la dernière valeur stockée, sinon mettre à jour la nouvelle valeur
    if (!isnan(humidityValue)) {
      lastHumidityValue = humidityValue;
    }
    else {
      humidityValue = lastHumidityValue;
    }

    String msg = "real time temperature: ";
    msg = msg + dtostrf(temperatureValue, 6, 2, msgBuffer);
    msg = msg + " C ";
    char message[58];
    msg.toCharArray(message, 58);
    Serial.println(message);

    // Publier la valeur de la température au serveur MQTT
    client.publish("client/temperature/", dtostrf(temperatureValue, 6, 2, msgBuffer));

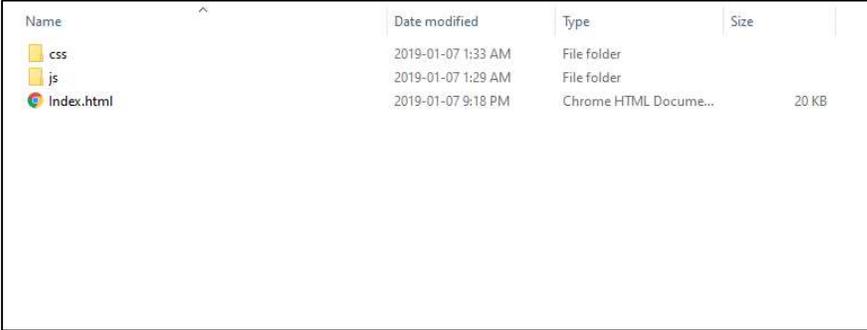
    // Publier la valeur d'humidité au serveur MQTT
    client.publish("client/humidity/", dtostrf(humidityValue, 6, 2, msgBuffer));

    // Vérifier la valeur de la température. Si elle dépasse le seuil maximale définie, nous exécutons l'alerte (ventilateur allumé, lampe rouge allumée) sinon
    // nous désactivons l'alerte (lampe verte et ventilateur éteint)
    if (temperatureValue >= maxTempAlert) {
      digitalWrite(fanPin, HIGH);
      digitalWrite(ledRedPin, HIGH);
      digitalWrite(ledGreenPin, LOW);
      forceStart = true;
    }
    else {
      if (forceStart == true) {
        digitalWrite(fanPin, LOW);
        digitalWrite(ledRedPin, LOW);
        digitalWrite(ledGreenPin, HIGH);
        forceStart = false;
        Serial.println("forceStart == false");
      }
    }
  }
}

```

Notes : Serial.println() : est une fonction d'affichage utilisée pour le débogage de notre système lors du développement.

Côté Web



Name	Date modified	Type	Size
css	2019-01-07 1:33 AM	File folder	
js	2019-01-07 1:29 AM	File folder	
Index.html	2019-01-07 9:18 PM	Chrome HTML Docume...	20 KB

Notre exemple de solution Web se compose de 2 dossiers (CSS et JS) et d'un fichier Index.html. En effet le fichier Index.html contiendra l'interface Web principale, le dossier CSS pour la mise en style et finalement JS pour tout ce qui est animation, configuration, WebSocket, JQuery, etc.

Dossier « CSS »

Ce dossier contient les fichiers de style pour la présentation de documents sur une page Web. Tous les styles, couleurs, les animations, les formes, etc. sont gérées dans ces fichiers. Nous utilisons la librairie Bootstrap pour les animations, l'aspect réactif de la page, les curseurs, etc. d'où l'inclusion des fichiers CSS de Bootstrap.

- Qu'est-ce que Bootstrap ?

Bootstrap est une infrastructure frontale gratuite pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres, ainsi que des plug-ins JavaScript en option.

Bootstrap donne également la possibilité de créer facilement des conceptions réactives

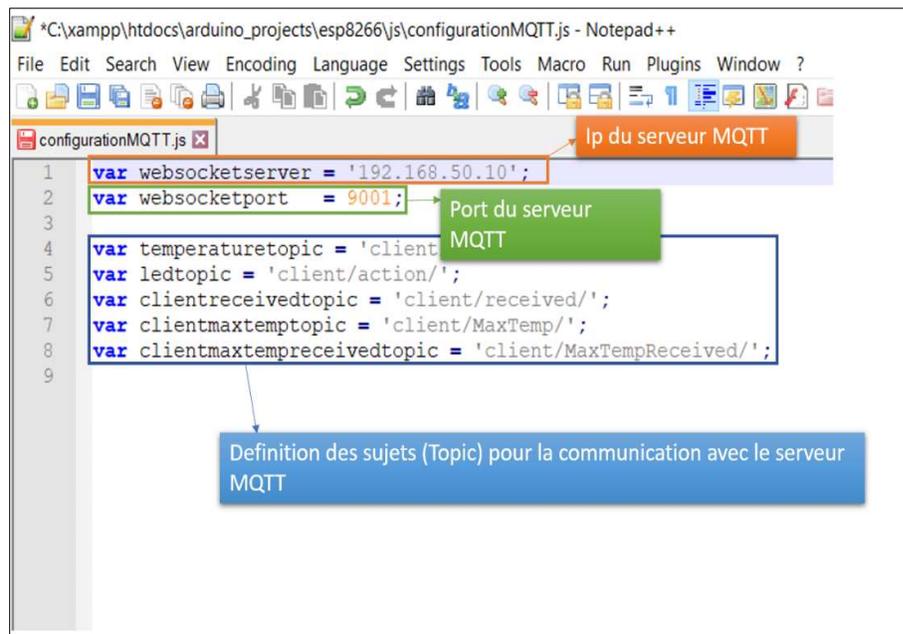
Dossier « JS ».

Le dossier JS contient les fichiers et librairies JavaScript utiles pour le bon fonctionnement de notre site. Il contient les librairies telles que Bootstrap, CanvasJs, Chart, etc. qui sont utiles pour les animations et les graphes présents dans notre interface Web.

En plus des fichiers utiles pour tout ce qui est de l'expérience utilisateur, nous trouvons aussi deux fichiers pour la configuration et la gestion des communications avec notre serveur **MQTT** à travers les **WebSockets**.

- Le fichier de configuration pour MQTT :

Ce fichier contient les informations utiles en rapport avec notre serveur MQTT. En effet il contient des variables qui contiennent les valeurs des sujets à communiquer avec le serveur MQTT, l'adresse IP du serveur MQTT et le port.



```
1 var websocketserver = '192.168.50.10';
2 var websocketport = 9001;
3
4 var temperaturetopic = 'client/temperature';
5 var ledtopic = 'client/action/';
6 var clientreceivedtopic = 'client/received/';
7 var clientmaxtemptopic = 'client/MaxTemp/';
8 var clientmaxtempreceivedtopic = 'client/MaxTempReceived/';
9
```

Annotations in the image:

- An orange box highlights the IP address '192.168.50.10' with the label "Ip du serveur MQTT".
- A green box highlights the port '9001' with the label "Port du serveur MQTT".
- A blue box highlights the topic definitions from line 4 to 8 with the label "Definition des sujets (Topic) pour la communication avec le serveur MQTT".

- La bibliothèque MQTT :

Ce fichier JavaScript est une bibliothèque client basée sur un navigateur MQTT écrite en JavaScript qui utilise WebSockets pour se connecter à un serveur MQTT implémenter dans le cadre de projet Paho de la fondation Éclipse.

Fichier « Index.html »

Ce fichier contient la présentation de notre page Web avec des design des courbes de la température et les boutons d'interaction pour le contrôle de notre système à distance.

En effet, dans ce fichier nous incluons une référence aux fichiers inclus dans les dossiers « CSS » pour tout ce qui est du style des éléments présents dans notre interface Web. Ainsi que,

les fichiers inclus dans le dossier « JS » pour inclure les bibliothèques JavaScript pour les animations d'une part et la communication à travers les Web socket avec notre serveur MQTT.

Pour finir, ce fichier inclut une section JavaScript dans laquelle nous effectuerons l'initiation de la communication à travers les Web socket avec notre serveur MQTT en utilisant des fonctions de la bibliothèque « mqttws31.js » ainsi que la gestion de l'aspect dynamique de notre interface est par la mise à jour de la température en temps réel, grâce à la réception des messages de mise à jour ou pour effectuer des actions telles que la mise en marche du ventilateur ou la modification du seuil de la température maximale.

Voici ci-dessous des portions de code pour l'initiation de la communication à travers les WebSockets présentent dans le fichier « Index.html » :

✓ Importation des bibliothèques

Importation de la bibliothèque et du fichier de configuration pour configurer le client MQTT

```
<title>Mqtt Sabrina</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap-toggle.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap-slider.css">
<!-- MQTT over Websockets -->
<script type="text/javascript" src="js/mqttws31.js"></script>
<script type="text/javascript" src="js/configurationMQTT.js"></script>
<script src="js/gauge.min.js"></script>
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
```

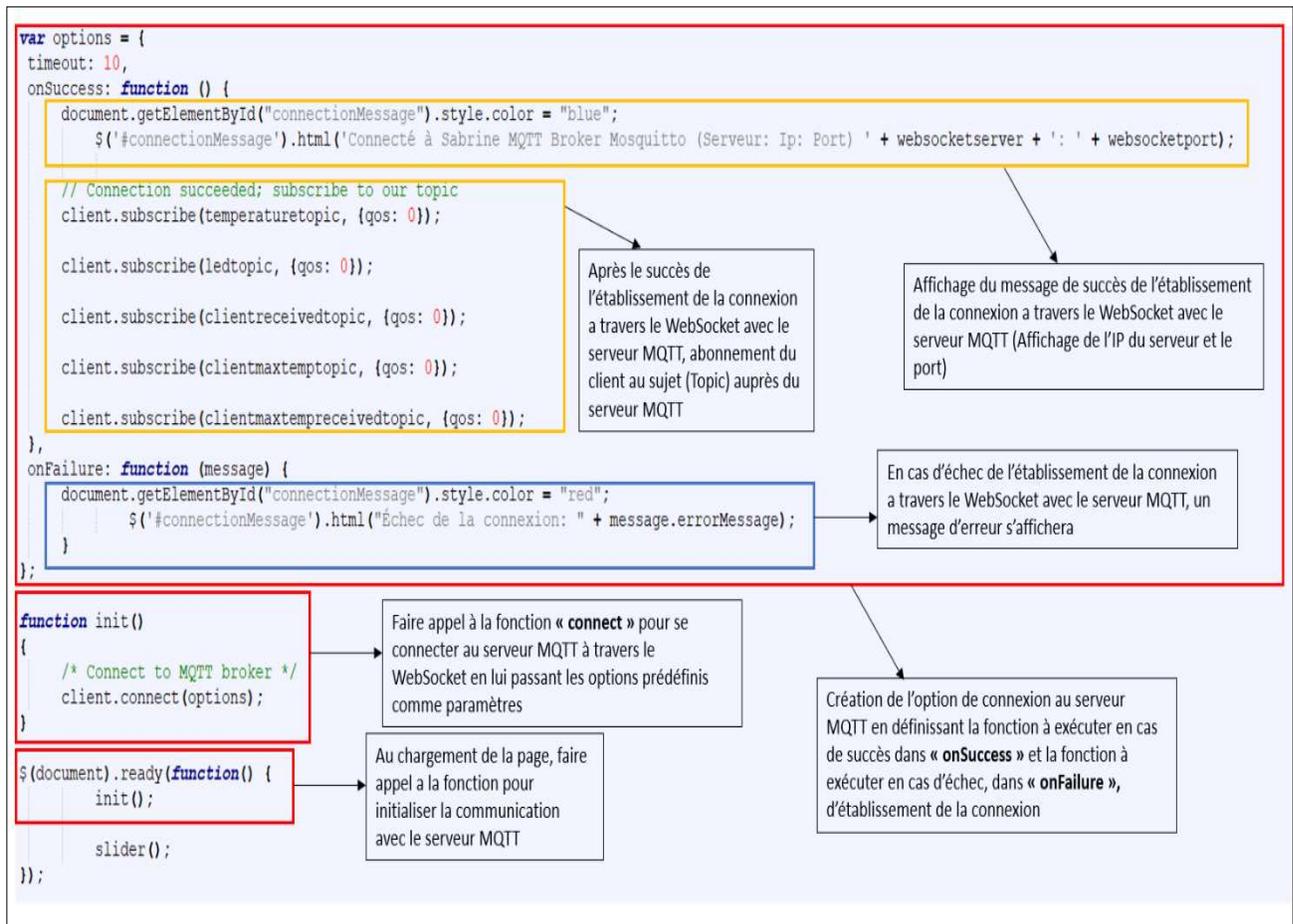
✓ Création d'un client MQTT

Création d'une instance de client pour se connecter au serveur MQTT

```
<script type="text/javascript">
var client = new Messaging.Client(websocketserver, websocketport, "web " + parseInt(Math.random() * 100, 10));
var gauge = null;
var chart = null;
var startDelete = false;
var lengthlabels = 3;
```

Adresse IP du serveur MQTT Port Identifiant du client

✓ Connexion au serveur MQTT à travers les WebSockets



✓ Exécution de la fonction « onConnectionLost » à la perte de la connexion au serveur MQTT

«onConnectionLost» Est un évènement qui se déclenche/s'exécute automatiquement en cas de perte de la connexion avec le serveur MQTT, un message d'erreur s'affichera

```
client.onConnectionLost = function (responseObject) {
  $('#connectionMessage').html("connection lost: " + responseObject.errorMessage);
  document.getElementById("connectionMessage").style.color = "red";
};
```

- ✓ Exécution de la fonction « onMessageArrived » à la réception d'un message du serveur MQTT à travers les Web socket

```

client.onMessageArrived = function (msg) {
    $('#connectionMessage').val(msg.payloadString);

    console.log(JSON.stringify(msg.destinationName));
    console.log(JSON.stringify(msg.payloadString));

    gauge = document.gauges.get('temperatureGauge');

    topic_parts = msg.destinationName.split('/');

    if (topic_parts[topic_parts.length - 2] == 'temperature')
    {
        gauge.value = msg.payloadString;
        addData(msg.payloadString, 1);
    }

    if (topic_parts[topic_parts.length - 2] == 'received')
    {
        clientreceived = true;
    }

    if (topic_parts[topic_parts.length - 2] == 'MaxTempReceived')
    {
        clientMaxTempreceived = true;
    }
};

```

«onMessageArrived» Est un évènement qui se déclenche/s'exécute automatiquement en cas de réception de message du serveur MQTT à travers le WebSocket pour les sujets dont il est souscrit

Récupération du sujet du message reçu

Récupération de la valeur associée au sujet reçu

Instruction à exécuter à la réception d'un message du serveur MQTT

- ✓ Publication des messages/Sujet auprès du serveur MQTT à travers les Web sockets

```

function slider()
{
    // Without JQuery
    var slider = new Slider("#ex6");
    slider.on("slide", function(sliderValue) {
        document.getElementById("ex6SliderVal").textContent = sliderValue + '°C';
    });
    slider.on("slideStop", function(sliderValue) {
        clientMaxTempreceived = false;

        firstRoomMaxValue = sliderValue;
        var message = new Messaging.Message(sliderValue.toString());
        message.destinationName = clientmaxtemptopic;
        envoi(message);
    });
}

function envoi (message)
{
    var refreshId = setInterval(function() {
        if (!clientMaxTempreceived)
        {
            client.send(message);
        }
        else
        {
            clientMaxTempreceived = false;
            clearInterval(refreshId);
        }
    }, 1000);
}

```

Création du message à publier au près du serveur MQTT en définissant la valeur

Définir le topic du message à publier au près du serveur MQTT

Publication du message au près du serveur MQTT à travers les WebSockets