

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Télécommunication
Spécialité Réseaux & Télécommunication

Présenté par

Ammour Redhouane

&

Benouh Walid

Réalisation de liaison WSN à base de l'ESP32 pour l'IoT

Encadré par :

Pr. Bensebti Messaoud

Année Universitaire 2020-2021

REMERCIEMENTS

Avant de vous convier à la présentation de ce travail, l'opportunité nous est donnée de témoigner notre gratitude et notre reconnaissance à toutes les personnes qui par leur aide et leurs encouragements nous ont permis de réaliser ce mémoire.

On tient à exprimer notre immense reconnaissance à notre promoteur Monsieur **Bensebti Messaoud** pour avoir accepté de nous encadrer et nous prodiguer ses conseils

Pertinents et ses critiques constructives et nous tenons à saluer en lui sa patience

Un grand merci aussi à tous les membres de ce prestigieux et distingué jury qui nous ont fait l'honneur, D'accepter d'évaluer et de juger notre travail.

Nos remerciements s'adressent à tous les enseignants qui ont contribué à notre formation durant nos années d'études à cette université.

Nous tenons à remercier tous nos amis et collègues pour leur soutien moral tout au long de la préparation de ce mémoire.

Sans oublier en fin de rendre un grand hommage à nos parents et frère(s) et

Sœur(s) qui nous ont encouragé et aidé par tous leurs moyens pour que l'on puisse

Terminer notre projet.

ملخص: هذا العمل هو نتيجة تعليم جامعي ثري يشمل التقنيات الجديدة في عالم الاتصالات والشبكات. مع الخبرات الشخصية لتسليط الضوء على كل المعرفة المكتسبة بالفعل. لهذا تم تنفيذ هذا العمل كجزء من مشروع التخرج الخاص بي، لغرض جعل الحياة أسهل للبشر واتباع الاتجاهات التكنولوجية. في التعامل مع هذا الموضوع، اعتمدنا على تصميم من خلال تحقيق العديد من التطبيقات، كل هذا لتحقيق الأهداف "MQTT "Cayenne" الشبكات اللاسلكية وإتقان منصة المرجوة.

من حيث النتائج، كشف تقييم الوضع الحالي عن بعض أوجه القصور، لكن العمل الذي تم تنفيذه يثبت أن التطبيقات التالية يمكن أن تساعد في تقدم وتحسين مجال إنترنت الأشياء في الجزائر.

من خلال هذا العمل نرغب في المساعدة في تقدم البحث الجامعي بهدف جعل الجزائر قافلة من التقنيات الحالية الكلمات المفتاحية: إنترنت الأشياء، الشبكات اللاسلكية،

Résumé : Ce travail est le fruit d'une formation universitaire riche qui englobe les nouvelles technologies dans le monde des télécommunications et des réseaux. Avec des expériences personnelles pour mettre en exergue toutes les connaissances déjà acquises. Pour cela ce travail a été réalisé dans le cadre de mon projet de fin d'étude, pour un but de facilité la vie de l'être humain et suivre les tendances de la technologie. En traitant ce sujet, nous nous sommes basées sur la conception des réseaux sans fils et la maîtrise d'une plateforme MQTT << Cayenne >> en réalisant plusieurs applications, tout cela pour atteindre les objectifs souhaités.

En terme de résultats, l'évaluation de la situation actuelle a révélé quelques insuffisances, mais le travail réalisé prouve que les prochaines applications peuvent aider a progresser et améliorer le domaine des Internet des Objets en Algérie.

A travers ce travail nous souhaitons aider dans la progression de la recherche universitaire pour le but de rendre l'Algérie convoi les technologies actuelles.

Mots clés : Internet des objets, MQTT, Réseaux sans fils.

Summary: This work is the result of a rich university education that encompasses new technologies in the world of telecommunications and networks. With personal experiences to highlight all the knowledge already acquired.

For this reason, this work was carried out within the framework of my end-of-study project, with the aim of facilitating the life of the human being and following the trends in technology.

In dealing with this topic, we based ourselves on the design of wireless networks and the mastery of an MQTT platform << Cayenne >> by realizing several applications, all this to achieve the desired objectives.

In terms of results, the evaluation of the current situation has revealed some shortcomings, but the work carried out proves that the next applications can help to progress and improve the field of Internet of Things in Algeria.

Through this work we wish to help in the progression of university research for the purpose of making Algeria suitable for current technologies.

Keywords: internet of things, MQTT, Wireless networks.

Table des matières :

Résumé :	II
Table des matières :	IV
Liste des figures :	VIII
LISTE DES ABREVIATIONS	X
LISTES DES TABLEAUX :	XIV
Introduction générale :.....	Erreur ! Signet non défini.
Chapitre 1 : Généralité sur l'IdO	2
1 Introduction	3
2 Historique de l'internet des objets	4
3 Définition de l'internet des objets	4
4 Objets connectés	4
a. Avantage des objets connectés :	5
b. Objets physiques et virtuels.....	5
5 Les « couches » de l'internet des objets	5
6 Les niveaux de l'internet des objets	5
7 Les Composants de l'internet des objets	7
8 Caractéristiques fondamentales de l'internet des objets	8
8.1 Inter connectivité.....	8
8.2 Services reliés aux objets	8
8.3 Métérogénéité.....	8
8.4 Changements dynamiques	8
8.5 Très grande échelle.....	8
9 Architecture du l'IdO	8

10	Domaine d'application de l'internet des objets	10
10.1	Domaine de santé	10
10.2	Domaine d'agriculture	10
10.3	Domaine de l'industrie	11
10.4	La domotique ou les maisons connectées	12
10.5	Domaine du commerce	13
10.6	Des villes intelligentes et connectées	14
10.7	Aux services des sécurités routières	15
11	Les avantages de l'IdO	16
12	Les inconvénients de l'internet des objets	17
13	Sécurité de l'internet des objets	17
14	Conclusion	18
Chapitre 2 : Composants de l'IdO et les protocoles de communications		19
1	Matériel de l'internet des objets	20
1.1	ESP8266	20
1.2	ESP 32	21
1.3	Comparaison entre ESP32 et ESP8266	22
1.4	Les Capteurs	23
1.4.1	Définition d'un capteur	23
1.4.2	Les caractéristiques d'un capteur	24
1.4.3	Critères de choix du capteur	24
1.5	Les actionneurs	24
2	La communication dans un système de l'IdO	25
2.1	Technologies de communication	25
2.1.1	Connectivité filaire	26
2.1.2	Connectivité sans fil	26
2.1.2.1	Bluetooth :	27
2.1.2.2	Zigbee :	27
2.1.2.3	Wi-Fi :	28
2.1.2.4	Etude comparatif entre les trois protocoles utilisés	28
2.2	Les protocoles de communication	29
2.2.1	Comparaison entre les protocoles	30
Chapitre 3 : Le protocole MQTT et les plateformes IdO		31
1	Le protocole MQTT	32

1.1	Définition	32
1.2	Mode de fonctionnement	32
1.2.1	Topologies MQTT	32
1.2.2	Fonctionnement	33
1.2.2.1	Connexion et Déconnexion	33
1.2.2.2	Abonnements et Publications	33
1.2.3	Qualité de service	34
1.2.4	Sécurité	35
1.3	Description informelle du protocole	36
1.3.1	Représentations de données	36
1.3.2	Structure d'un paquet de contrôle MQTT	36
1.3.3	La synchronisation des requêtes envoyées au broker	37
1.3.4	L'identifiant de client MQTT	37
1.3.5	Callbacks	38
2	Les plateformes de l'IDO	39
2.1	Définition d'une plateforme	39
2.2	Choix d'une plateforme	39
2.3	Plateforme Propriétaire vs. Open-Source	39
2.4	Critères du choix d'une plateforme	40
2.4.1	La gestion des devices	40
2.4.2	La gestion de la connectivité	41
2.4.3	Visualisation de la donnée	41
2.4.4	Sécurité des objets et des données	41
2.4.5	Le degré d'agilité	42
Chapitre 4 : Applications WSN		43
1 INTRODUCTION		44
2 Interface Arduino IDE		45
3 Communication entre l'interface et la carte		46
3.1	Bibliothèques	46
4 Serveur et client BLE		47
4.1	Définition du BLE	47
4.2	Caractéristiques	47
4.3	Fonctionnement	48
4.4	Avantages du BLE	48

4.5 Exemple.....	49
5 Communication bidirectionnelle ESP-NOW entre les cartes ESP32.....	51
5.1 Présentation d'ESP-NOW.....	51
5.2 Obtenir l'adresse MAC de la carte.....	52
5.3 Code ESP-NOW de communication bidirectionnelle ESP32.....	53
6 Établissement d'une connexion WI-FI avec la carte esp32.....	53
6.1 contrôle de LED via WEB.....	53
6.2 mesure de qualité du signal (RSSI).....	54
7 Communication WI-FI client-serveur entre deux carte ESP.....	55
7.1 Introduction.....	55
7.2 Les étapes du travail.....	56
7.3 Serveur ESP32.....	56
7.3.1 Diagramme schématique.....	57
7.4 Client ESP32 (Station).....	59
7.5 Conclusion.....	60
Chapitre 5 : Applications Cayenne.....	61
1 Introduction.....	62
2 Historique.....	62
3 À propos de MyDevices.....	63
3.1 Le rôle de la plateforme Cayenne.....	63
3.2 Les Principales caractéristiques.....	63
3.3 Fonctionnement de Cayenne.....	64
4 Création d'un compte.....	65
5 Création d'un appareil Widget.....	68
5.1 Touche.....	68
5.2 LED :.....	70
5.3 L'utilisation d'une boucle dans la plateforme Cayenne MQTT.....	71
5.3.1 Création d'une boucle.....	71
5.3.2 L'utilisation d'un déclencheur dans la plateforme Cayenne MQTT :.....	73
5.4 L'application RSSI :.....	74
6 Conclusion :.....	76
Conclusion Générale :.....	77
BIBLIOGRAPHIE.....	78

Liste des figures :

Figure 1.1 : l'Internet des Objets.....	3
Figure 1.2 : l'Architecture de l'IdO.....	9
Figure 1.3 : connexion et sécurisation des objets connectés de santé	10
Figure 1.4 : l'évolution de l'IdO dans le domaine de l'agriculture	11
Figure 1.5: l'IdO et le monde de l'industrie	12
Figure 1.6: l' IdO et les maisons connectées	13
Figure 1.7: l'IdO et son influence sur le commerce mondial	14
Figure 1.8: l'IdO et les villes connectées	15
Figure 1.9 : Rendre les routes intelligentes	16
Figure 1.10: Les inconvénients de l'IdO	17
Figure 2.1: ESP8266.....	19
Figure 2.2: ESP 32.....	20
Figure 2.3: Principe de fonctionnement d'un capteur	22
Figure 2.4: Les différents réseaux	26
Figure 3.1 : Topologie MQTT.....	31
Figure 4.1 : Un réseau	43
Figure 4.2 : Fonction Setup et Loop.....	44
Figure 4.3 : Paramètre Arduino BLE	45
Figure 4.4: Gestionnaire de la carte.....	46
Figure 4.5 : La disposition du BLE	47
Figure 4.6 : Générateur de UUID	48
Figure 4.7: Connexion entre le téléphone et la carte	49
Figure 4.8 : Connexion entre le téléphone et la carte	49
Figure 4.9: La réception du message.....	50
Figure 4.10 : ESP NOW	51
Figure 4.11: Adresse MAC	51
Figure 4.12 : La réception des paquets avec succès	52
Figure 4.13 : Affichage de l'adresse IP	52
Figure 4.14 : Page de contrôle.....	53
Figure 4.15: Mesure Du RSSI	54
Figure 4.16 : Client serveur ESP32	55
Figure 4.17 : La compilation du code.....	56

Figure 4.18 : Connexion via Smartphone.....	57
Figure 4.19 : Visualisation de la température.....	57
Figure 4.20 : Visualisation de la l'humidité	58
Figure 4.21 : Visualisation de la pression	58
Figure 4.22 : Visualisation de la pression, la température et l'humidité via LCD	59
Figure 5.1 : Page d'inscription à Cayenne.....	64
Figure 5.2 : Page d'accueils Cayenne.....	64
Figure 5.3 : Page des données de la carte	65
Figure 5.4 : Installation des bibliothèques.....	65
Figure 5.5 : Intégration des données dans le code IDE	66
Figure 5.6 : Tableau de bord Cayenne par défauts	66
Figure 5.7 : Création d'un nouveau Widget	67
Figure 5.8 : Paramètres du widget touche	67
Figure 5.9: L'icône du widget touche sans toucher le port dédié.....	68
Figure 5.10 : L'icône du widget quand on touche le port.....	68
Figure 5.11: Paramètres du widget LED	69
Figure 5.12 : Tableau de bord Cayenne après avoir ajouté les Widgets.....	69
Figure 5.13 : Ajouter un gâchette boucle	70
Figure 5.14 : Paramètres de la gâchette boucle	70
Figure 5.15 : Les Widgets sans touché le port.....	71
Figure 5.16 : Les Widgets après avoir touché Le port.....	71
Figure 5.17 : Paramètres du déclencheur	72
Figure 5.18 : l'email reçu de la part de Cayenne.....	72
Figure 5.19 : Paramètres du widget RSSI Valeur.....	73
Figure 5.20 : Paramètres du widget RSSI jauge.....	74
Figure 5.21: Tableau de bord final	75

LISTE DES ABREVIATIONS :

3G/4G : 3e Génération / 4e

ADC: Analogic Digital Signal

ADSL: asymmetric digital subscriber line

AMQP: Advanced Message Queuing Protocol

AO: Analog Out

AP: Access Point

API: Application Programming Interface

AVR GCC: Automatic Voltage Regulator Gnu Compiler Collection

AWS: Amazon Web Services

BLE: Bluetooth Low Energy

BME280: Grove – Temp&Humidity &Barometer Sensor

CCV: Closed Circuit Voltage

CEO: Chief Executive Officer

CPL: Computer Programming Languages

DAC: Digital Analogic Signal

Dbm: Decibel mill watts

DO: Digital out

DRAM: Dynamic Random Access Memory Device.

DSL: Digital subscriber Line

DUP: Dial-up Networking

E/S: Input/output

EDGE: Enhanced Data Rates for GSM Evolution

EEPROM: electrically erasable programmable read-only memory

ESP: Espressif Systems Pune

FTTB: Fibre to The Block

FTTD: Fibre to The Desk

FTTH: Fibre to The House

GND: Ground

GPIO: general-purpose input/output

GPRS: General Packet Radio Service

GSM: Global system Mobile communication

http: Hypertext Transfer Protocol

IDE: integrated development environment

IDO: Internet des Objects

ILS: Instrument Landing System

IoT: Internet of Things

IP: Internet Protocol

ISM: Integration Scaling Matrix

ISO: International Organization for Standardization

LAD: Lampe a décharge

LAN: Local Area Network

LED: Light Emitting Diode

LoRa: Long Range

LPWA: Low-Power Wide-Area

LTE: Long-Term Evolution

LTE-M: Long Term Evolution for Machines

M2M: Machine to Machine

MAC: Media Access Control

M-BUS: Meter – bus

MCU: Microcontroller Unit

MQTT: Message Queuing Telemetry Transport

NBIOT: Narrowband IOT

NB-IOT: Narrowband-Internet of Things

NFC: Near Field Communication

OLED: Organic Light-Emitting Diode

OPEX: Operating Expense

P&G: Procter & Gamble

PaaS: Platform as a Service

PC: Personal Computer

PING: Packet Internet Groper

PLC: Power Line Communication

PWM: Pulse-width modulation

QoS: Quality of Service

RAM : Random access memory

RCSFs : Réseau de Capteur Sans Fils

RFID: Radio Frequency identification

RJ11: Registered Jack-11

RSSI: Received Signal Strength Indication

RTC: Real Time Connection

SCL: System Control Layer

SDA: Small Domestic Appliances

SDKS: Software Development Kit System

SDSL: symmetric digital subscriber line

SP: Extra Sensory Perception

SPI: Service Provider Interface

SRAM: Static random-access memory

SSL: Secure Sockets Layer

TCP/IP: Transmission Control Protocol/Internet Protocol

UART : Universal Asynchrone Recepteur Transmetteur

UIT : Union internationale des télécommunications

UMTS: Universal Mobile Telecommunications System

URL: Uniform Resource Locator

USB: Universal Serial Bus

UTF-8: Universal Character Set Transformation Format - 8 bits

UUID: Universally Unique Identifiers

VCC: Voltage Common Collector

VIN: Vehicle Identification Number

VLAN: Virtual Local Area Network

Wi-Fi: Wireless Fidelity

WiMAX: Worldwide Interoperability for Microwave Access

WLAN: Wireless Local Area Network

WMAN: Wireless Metropolitan Area Network

WPAN: Wireline Packet Data Network

LISTE DES TABLEAUX :

Tableau 1 : Les composants de base d'un système IdO	7
Tableau 2: Comparaison entre ESP32 et ESP8266	21
Tableau 3 : Les caractéristiques des réseaux sans fil.....	28
Tableau 4 :Comportement de synchronisation des méthodes qui donnent lieu à des requêtes au broker	36
Tableau 5: Les avantages et les inconvénients des plateforme Propriétaires et Open-source	39
Tableau 6 : Table de branchement (BME280 /ESP32 server).....	56
Tableau 7 : Table de branchement (ESP client / LCD)	58

Introduction générale :

L'internet des objets est un nouvel outil de connectivité et de mobilité, qui transforme les affaires et la vie quotidienne à des objets connectés. Les objets courants deviennent actifs et intelligents, s'intégrant de façon transparente à un réseau mondial et sont en mesure de produire et d'échanger des données utiles sans intervention humaine. Il s'agit d'un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et sans fil, d'identifier et de communiquer numériquement avec des objets physiques afin de pouvoir mesurer et échanger des données entre les mondes physiques et virtuels. A travers un tel paradigme, aujourd'hui l'IdO couvrira un large éventail d'applications et touchera quasiment à tous les domaines que nous affrontons au quotidien, ceci permettra l'émergence d'espaces intelligents.

Notre monde connaît une prolifération étonnante des objets intelligents ce qui a poussé les sociétés des services à mettre à la disposition des intéressés des cartes des plateformes.

Leur but est de Contrôler des objets a distances sur l'infrastructure réseau existante, en créant des opportunités d'intégrations plus direct entre le monde physique et les systèmes informatiques.

L'internet des objets aspire à répondre aux besoins de l'utilisateur qui recherchent le bon service au bon moment et au bon endroit, et c'est en développant de nouvelles applications qu'il sera possible d'exploiter cette technologie pour le bien-être de l'homme aussi bien le plan individuel et sur le plan de la société.

Dans ces travaux nous allons nous intéresser à cette nouvelle tendance des objets connecté dans le but d'améliorer et faciliter leur utilisation. Pour cela nous allons réaliser plusieurs applications en relient notre travail avec une plateforme générique pour la gestion des réseaux IOT.

Ce mémoire va être organisé en cinq chapitres.

Le premier chapitre sera consacré à la présentation de l'internet des objets, en expliquant les objets connecté, les couches, les niveaux, les caractéristiques, l'architecture, fonctionnement et les domaines d'application de l'ido

Dans le deuxième chapitre nous présenterons les matériels utiliser ainsi les différents systèmes de communication.

Dans le chapitre trois on va s'intéressé au protocole MQTT et les différentes plateformes IdO

Dans les deux derniers chapitres nous allons exposer les différentes applications et présenter les Résultat

Chapitre 1 :

Généralité sur l'IdO

1 Introduction

Internet en général et le web en particulier n'ont cessé d'évoluer : du web des informations au web des objets² individualisés, via divers objets connectés grâce à la miniaturisation et le développement technologique, qui permet un double aspect : être connecté et communiquant en permanence sans contrainte pour l'espace et le temps afin de répondre aux exigences et aux besoins de l'utilisateur au niveau des services, de la communication et de l'information

Internet se transforme progressivement en un Hyper Réseau, comme un réseau formé par des multitudes de connexions entre des Artefacts (physiques, documentaires), des acteurs (biologiques, algorithmiques), des écritures et des concepts (linked data, metadata, ontologies), appelé « Internet of Things (IoT) Internet des objets (IdO) », connectant des milliards d'êtres humains mais aussi des milliards d'objets. Il devient l'outil le plus puissant jamais inventé par l'homme pour créer, modifier et partager les informations. Cette transformation montre l'évolution du réseau d'internet : d'un réseau de calculateurs vers un réseau d'ordinateurs personnels, puis vers un réseau nomade intégrant les technologies des communications.

Les développements des technologies Machine-to-Machine (M2M) pour le contrôle de machine à distance et aussi l'apparition dans l'année



Figure 1.1 : l'Internet des Objets

2 Historique de l'internet des objets

L'IdO est un système utilisant diverses technologies, allant de l'Internet à la communication sans fil des systèmes micro-électromécaniques, jusqu'aux systèmes embarqués.

Le principe de contrôler des objets tel que les équipements électriques et électroniques à distance via internet existent depuis le début des années 1990, lorsque John Romkey a créé le premier appareil avec Internet : un grille-pain, qu'il pouvait allumer et éteindre via Internet. Le terme (internet of things) a été proclamé pour la première fois par le Britannique Kevin Ashton employé de Procter & Gamble (P&G), dans une présentation qu'il a faite en 1999, reliant la nouvelle idée de la RFID (Radio Frequency identification), grâce à cette présentation, se diffuse l'idée de machines capables de communiquer entre elles.

Le terme « objets » fait référence à tout ce qui nous entoure, d'un petit atome à un gros navire

Le premier objet connecté est commercialisé en 2003 par la firme Violet. C'est la lampe DAL, équipée de 9 LEDs qui s'allument en fonction des événements. Deux ans plus tard, en 2007 c'est l'apparition des Smartphones. Enfin en 2008 c'était la création des adresses IPSO, adresses IP des objets connectés qui leur permettent d'interagir entre eux. [1]

3 Définition de l'internet des objets

L'Internet des objets est un système interdépendant de dispositifs mécaniques et numériques, d'objets, d'animaux ou de personnes, dotés d'identifiants uniques permettant de transférer des données sur un réseau sans avoir besoin d'interactions humain-humain ou humain ordinateur. L'environnement IdO permet aux utilisateurs de gérer et d'optimiser les équipements électroniques et électriques via Internet. [2]

4 Objets connectés

Les objets sont des participants actifs dans un système IdO, ils peuvent interagir et communiquer entre eux et avec leurs environnement, échanger des données et des informations détectées ou collectées à partir de son environnement, tout en réagissant de manière autonome aux événements réels /physiques [3]

Un objet connecté est dans la plupart du temps composé de :

- Un ou des capteurs
- Une petite mémoire
- Un processeur
- Une antenne de communication (Bluetooth, wifi, ...)

a. Avantage des objets connectés :

- Possibilité de stocker et de traiter de nombreuses informations.
- Taille réduite de l'équipement nécessitant moins de composants.
- Cout réduit à la production de l'équipement.
- Ils permettent de contrôler l'état de l'environnement.

b. Objets physiques et virtuels

Les objets physiques : ils font partie du monde physique. Ces objets peuvent être à la fois commandés, détectés et connectés. Les différents robots ainsi que tous les biens et équipements électriques sont des exemples d'objets physiques

Les objets virtuels : ils font partie du monde de l'information. Ces objets peuvent être stockés, traités et accédés. Les contenus multimédias ou différents logiciels sont des exemples d'objets virtuels. [4]

5 Les « couches » de l'internet des objets

Dans un système IdO, les données sont générées par plusieurs types de dispositifs, traitées de manières différentes et transmises dans différents emplacements par des applications.

L'architecture orientée aux services a été appliquée comme technologie habilitante par qui divisait l'IdO en quatre couches [5] :

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

6 Les niveaux de l'internet des objets

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

d'équipement ou pourrait être distribué par ailleurs au monde. Les réseaux traditionnels de communication de données ont de multiples fonctions, comme en témoigne le modèle de référence à 7 couches de l'Organisation internationale de normalisation (ISO). Cependant, un système IdO complet contient de nombreux niveaux en plus du réseau de communication. Le modèle de référence IdO permet également de varier les traitements effectués à chaque niveau de trivial à complexe, en fonction de la situation. Le modèle décrit comment les tâches à chaque niveau doivent être traitées pour maintenir la simplicité, permettre une grande évolutivité et assurer un soutien. Enfin, le modèle définit les fonctions nécessaires pour qu'un système IdO soit complet.

Niveau 1 : Dispositifs physiques et contrôleurs

- Niveau 2 : Connectivité
- Niveau 3 : Traduction (Computation de Borde)
- Niveau 4 : Accumulation des données
- Niveau 5 : Abstraction des données
- Niveau 6 et 7 : Application, collaboration et processus

7 Les Composants de l'internet des objets

Le tableau suivant présente les composants principaux de l'IdO:

Composant IdO	La description
Objets physiques	Un objet connecté est un objet physique équipé de capteurs ou d'une puce qui lui permettent de transcender son usage initial pour proposer de nouveaux services. Il s'agit d'un matériel électronique capable de communiquer avec un ordinateur, un Smartphone ou une tablette via un réseau sans fil (Wi-Fi, Bluetooth, réseaux de téléphonie mobile, réseau radio à longue portée de type Sigfox ou LoRa, etc.), qui le relie à Internet ou à un réseau local.
Capteurs	Ils sont installés sur les es objets connectés, ils sont plus ou moins intelligents, selon qu'ils intègrent ou non eux-mêmes des algorithmes d'analyse de données, et qu'ils soient pour certains auto-adaptatifs. Les capteurs connus sont : Capteurs de température et thermostats, Capteurs de pression, Humidité / niveau d'humidité, Détecteurs d'intensité lumineuse, Capteurs d'humidité, Détection de proximité, Étiquettes RFID....
Gens	Exemple : Les humains peuvent contrôler l'environnement via des applications mobiles
Prestations de service	Exemple : Services Cloud - peuvent être utilisés pour: <ul style="list-style-type: none"> • Traiter les Big Data et les transformer en informations précieuses • Construire et exécuter des applications innovantes • Optimiser les processus métier en intégrant les données de l'appareil.
Plateformes	Elle est considérée comme un type d'intergiciel utilisé pour connecter les composants IdO (objets, personnes, services, etc.) à l'environnement l'IdO. Elle fournit de nombreuses fonctions : <ul style="list-style-type: none"> • Accès aux appareils • Assurer une installation / un comportement correct de l'appareil • Analyse des données • Connexion interopérable avec le réseau local, le cloud ou d'autres périphériques.
Réseaux	Les composants IdO sont liés entre eux par des réseaux, utilisant diverses technologies, normes et protocoles sans fil et filaire .

Tableau 1 : Les composants de base d'un système IdO

8 Caractéristiques fondamentales de l'internet des objets

Le paragraphe ci-dessous, indique les caractéristiques fondamentales de l'internet des objets ainsi que leurs exigences.

8.1 Inter connectivité

Tous les objets présents dans l'internet des objets peuvent être connectés à l'infrastructure mondiale de l'information et de la communication. Parmi les formes de connectivités les plus connus entre l'IdO et l'internet.

Cependant, tous les appareils peuvent se connecter via une large variété de mode de connexion et de technologies, qu'elles soient avec (Ethernet) ou sans fil.

8.2 Services reliés aux objets

L'IdO est capable de fournir des services qui sont liés aux objets tout en tenant compte des contraintes telles que la production de la vie privée ainsi que la cohérence sémantique entre les objets physiques

8.3 Métérogénéité

Etant donné qu'ils sont basés sur des plates-formes matérielles ou des réseaux différents, les appareils reliés à l'IdO sont hétérogènes. Ils peuvent ainsi interagir avec d'autres appareils et dispositifs ou des plates-formes de service via d'autres réseaux.

8.4 Changements dynamiques

L'état des dispositifs (par exemple connecté/déconnecté) change de manière dynamique tout comme le contexte dans lequel ces dispositifs fonctionnent qu'ils sont reliés au cadre spatio-temporel, ou également dans le cadre de la vitesse ou encore de la localisation.

8.5 Très grande échelle

Dans le futur, l'ensemble des dispositifs sera au moins dix fois plus nombreux qu'à l'heure actuelle comme le précise l'UIT.ils devront donc être à la fois gérés et être capables de communiquer entre eux.

9 Architecture du l'IdO

L'architecture d'un système IdO est composée de plusieurs niveaux qui communiquent entre eux pour relier le monde tangible des objets au monde virtuel des réseaux et du cloud. Tous les projets n'adoptent pas une architecture formellement identique, néanmoins il est possible de schématiser le parcours de la donnée.

Un capteur commence par relever une mesure, La mesure relevée par le capteur est ensuite envoyée à la passerelle grâce à un premier protocole de communication. Le rôle de la passerelle est de traduire les protocoles pour établir une communication entre les objets et le réseau (public ou privé, souvent le cloud). Parfois, une intelligence embarquée dans la passerelle permet le traitement et le stockage de la donnée, ainsi que des fonctions de pilotage

de l'objet. Pour les protocoles courte portée (Zigbee, Z-Wave, WiFi, BLE,...), la passerelle est locale et se connecte souvent à la box du fournisseur d'accès à internet. Pour les protocoles longue portée (Sigfox, LoRa, LTEM, NBIoT, 3G/4G, ...), la passerelle se trouve sur le réseau de l'opérateur télécom. [6]

La plateforme IdO est une plateforme technique qui permet de collecter les données et superviser la flotte d'objets. La plateforme logicielle IdO embarque une intelligence qui offre de multiples possibilités. Les mesures peuvent être stockées dans l'objectif de créer une historisation permettant la prédiction. La plateforme IdO communique avec différentes plateformes de services qui délivrent un service digital à l'utilisateur final restituant les données collectées des objets, permettant des actions de pilotage et apportant une couche d'intelligence (alertes, conseils, ...).

Il faut noter que la stratégie de sécurité doit être intégrée entièrement dans l'architecture pour éviter les failles matérielles et logicielles à tous les niveaux du système.

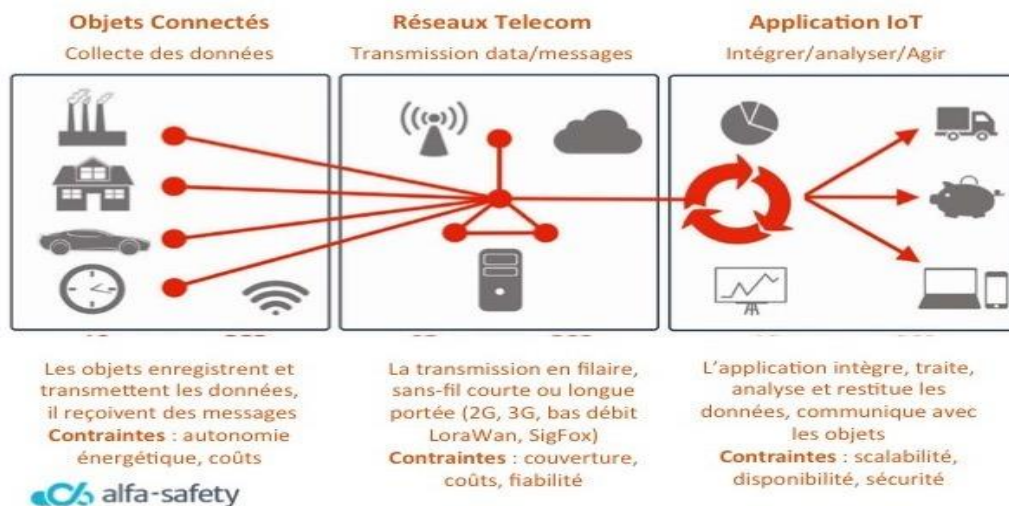


Figure 1.2 : l'Architecture de l'IdO

10 Domaine d'application de l'internet des objets

10.1 Domaine de santé

Machines à rayons X et imagerie, moniteurs connectés, compteurs d'énergie... 60 % des hôpitaux mondiaux utilisent déjà l'Internet des Objets pour augmenter leur productivité et améliorer les soins apportés aux patients. L'étude d'Aruba networks montre que d'ici 2019, c'est presque 90 % des services de santé qui auront intégré les objets connectés dans leur matériel médical.

Les objets connectés sont utilisés au quotidien pour :

- La surveillance au sein des établissements médicaux et la maintenance
- Les opérations chirurgicales et le contrôle à distance
- Les services de géo localisation



Figure 1.3 : connexion et sécurisation des objets connectés de santé

10.2 Domaine d'agriculture

La croissance rapide de la population mondiale, les changements d'habitudes alimentaires, les perturbations climatiques sont trois grands facteurs, parmi d'autres, qui font de l'agriculture moderne un défi au quotidien.

D'ici 2050, la productivité agricole devra avoir augmenté de 70 % pour répondre à la demande mondiale. Plus qu'un défi technologique, il s'agit d'un enjeu humanitaire.

Les céréaliers et maraîchers ont d'ores et déjà mis à profit les drones afin de récolter en temps réel des informations essentielles à la gestion de l'exploitation :

- Humidité de la terre
- État des plantations
- Climat, etc.

Les données récoltées sont transférées aux tracteurs connectés (parfois autonomes). Cela permet de doser finement le niveau d'engrais et d'arrosage sur telle ou telle parcelle et de réduire les coûts, tant financiers qu'énergétiques.



Figure 1.4 : L'évolution de l'IdO dans le domaine de l'agriculture

10.3 Domaine de l'industrie

L'industrie n'est pas en reste sur l'usage de l'Internet des Objets et des bénéfices que celui-ci lui apporte. Dans le cadre des problématiques rencontrées dans le domaine industriel, l'usage des objets connectés est très spécifique et répond à des besoins :

- D'optimisation (chaîne logistique)
- De transformation des processus d'entreprise
- D'amélioration de l'efficacité et de la productivité
- De traçabilité et de sécurité



Figure 1.5: l'IdO et le monde de l'industrie

10.4 La domotique ou les maisons connectées

Appelée également domotique, la maison intelligente est en train de se normaliser. Une étude du cabinet Juniper Research prévoit d'ailleurs un accroissement de 200 % du nombre d'objets connectés à l'intérieur des habitations d'ici fin 2021.

Outre les objets de divertissement comme les télévisions intelligentes ou les enceintes connectées, la domotique a pensé également la sécurité et l'économie d'énergie au sein de l'habitat :

- Centrale domotique : contrôle et programmation de différentes interventions à l'intérieur du foyer
- Capteurs d'informations (système d'alarme, variations de température, etc.)
- Actionneurs, qui permettent la programmation et le contrôle des différents appareils électroniques du foyer, même à distance



Figure 1.6: l' IdO et les maisons connectées

10.5 Domaine du commerce

Le commerce physique subit aussi les transformations de l'ère digitale. Fortement concurrencées par le e-commerce et m-commerce, les boutiques de vente au détail veulent tirer profit de la popularité de l'IdO en alliant l'e-commerce à la vente traditionnelle.

Les boutiques physiques ont donc elles aussi pris le pas de la révolution numérique et sont de plus en plus nombreuses à proposer des fonctionnalités ludiques et interactives afin de renforcer l'expérience de vente et accroître le taux de conversion.

On retrouve, parmi les concepts de « smart retail », la technologie d'identification par radiofréquence (RFID) qui permet de renforcer l'expérience client en offrant un parcours client ultra personnalisé. Outre les applications mobiles, des concepts de caddies connectés ont déjà été pensés pour faciliter les courses en supermarché :

- Liste de courses intégrée
- Parcours guidé pour optimiser le temps de course
- Calcul automatique du montant du panier, ...

Les commerçants investissent également dans les applications mobiles pour fidéliser et attirer les clients vers les boutiques physiques, par le biais, par exemple, de notifications sur les promotions / soldes en cours lors du passage d'un client près d'une boutique.



Figure 1.7: L'IdO et son influence sur le commerce mondial

10.6 Des villes intelligentes et connectées

Appelées « smart cities » ou encore « villes connectées », les villes ont entamé leur transition digitale pour répondre aux enjeux de la société moderne. Urbanisme, économie et développement durable forment des enjeux importants pour les villes de demain, qui doivent répondre aux besoins d'une population toujours plus dense avec des ressources de plus en plus limitées.

En effet, la pollution de même que la surpopulation sont deux problématiques majeures pour les agglomérations, qui doivent répondre aux enjeux économiques et sociaux d'une population croissante sans dégrader la qualité environnementale dans laquelle elle évolue.

Là encore, l'Internet des Objets est utile :

- Foyers domotiques
- Supports numériques
- Capteurs et compteurs intelligents
- Bornes de rechargement pour véhicules électriques
- Éclairage citoyen intelligent, etc.

Les applications des systèmes connectés sont nombreuses et c'est un véritable viviers d'innovations pour les métiers de l'informatique et du digital. Au sein des villes intelligentes, la gestion de la consommation du particulier est mieux gérée, tout comme le traitement à plus large échelle des consommations de la ville (coût et pollution des

transports en communs et déplacement urbains, traitement des déchets, gestion de l'électricité...).

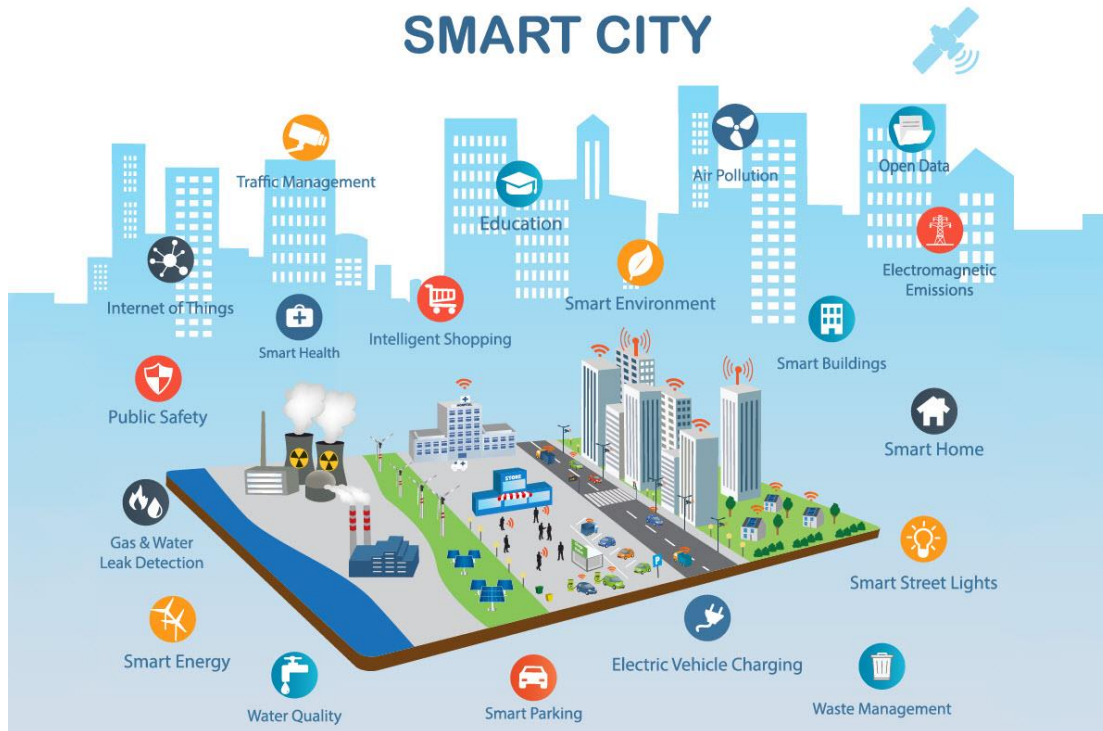


Figure 1.8: L'IdO et les villes connectées

10.7 Aux services des sécurités routières

Particulièrement populaire depuis ces dernières années la voiture connectée participe grandement au renforcement de la sécurité routière. La révolution numérique a offert à l'industrie automobile des perspectives jusqu'alors jamais vues.

- Boîtier d'appel d'urgence autonome
- Tableau de bord synchronisé avec le smartphone
- Développement d'applications sur les plateformes dédiées...

La voiture d'aujourd'hui se transforme en véritable ordinateur qui conduit peu à peu à la voiture autonome, comme celle actuellement en essai chez Google. Si nos véhicules ne sont pas encore capables de se conduire tout seuls, ils n'en deviennent pas moins de plus en plus autonomes grâce à un système d'automatisation de certaines tâches de conduite (allumage des phares, park assist, freinage automatique).

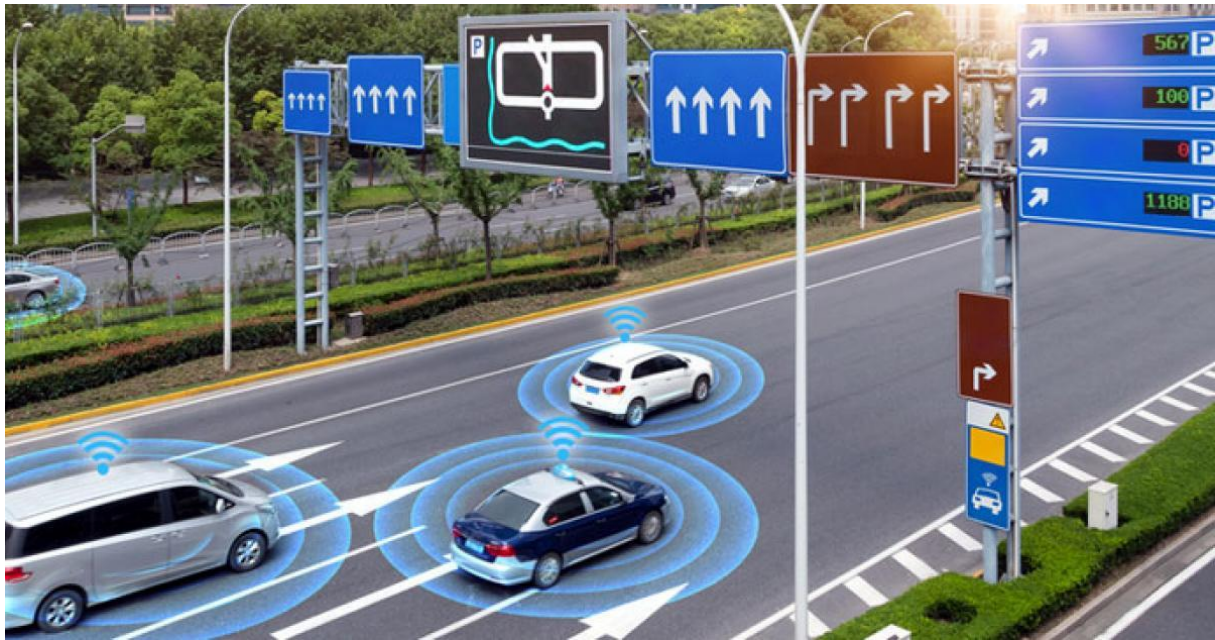


Figure 1.9 : Rendre les routes intelligentes

11 Les avantages de l'IdO

L'Internet des objets promet donc une forte valeur ajoutée à chaque organisation. En connectant les objets, les personnes et les environnements, il devient possible de développer des améliorations qui ne peuvent être que bénéfiques.

Les principaux avantages que peut avoir une entreprise dotée d'un système IdO sont les suivants :

- Amélioration de la productivité : L'utilisation des objets connectés permet de surveiller, de suivre et de contrôler les différents processus, ce qui optimise les différentes opérations qui augmentent la productivité et l'efficacité
- Analyse prédictive : Grâce à la collecte d'un grand nombre de données, les nouvelles technologies de L'utilisation des objets connectés permettent d'examiner des modèles récurrents et contribuent à l'analyse prédictive, qui peut être utilisée principalement dans le cadre de la maintenance. Ces informations précises seront utilisées pour améliorer les processus et les services existants
- Réponse rapide : les données permettent de surveiller les systèmes en place en temps réel et même à distance. Elles facilitent l'optimisation des interventions de maintenance, mais donnent également à l'entreprise un avantage stratégique pour suivre l'évolution du marché.
- Réduction des erreurs humaines : grâce à la complémentarité de technologies telles que l'intelligence artificielle, L'utilisation des objets connectés permet de réduire les erreurs humaines dues à des tâches banales ou répétitives. [10]

12 Les inconvénients de l'internet des objets

L'IdO facilite grandement le quotidien de nombreuses personnes, que ce soit dans un cadre particulier ou professionnel. Toutefois, à trop se reposer sur cette technologie révolutionnaire, si un problème de performance venait à l'impacter cela aurait pour conséquence une baisse sur l'activité et le chiffre d'affaires. De plus, il ne faut pas oublier qu'il s'agit d'objet connecté, et le risque de protection des données reste donc présent, même si de nombreuses sécurités sont mises en place.

D'autre part, certains sont réticents à mettre en œuvre cette technologie par un manque de connaissance. Dans ce cas de figure, les risques liés à l'utilisation de l'IdO surpassent les bénéfices.

En effet, l'IdO, lorsqu'elle est bien utilisée, permet de réduire les coûts, et d'améliorer la productivité d'une entreprise. Il est donc important pour les professionnels de rester informés et surtout de se former aux nouvelles technologies [10]

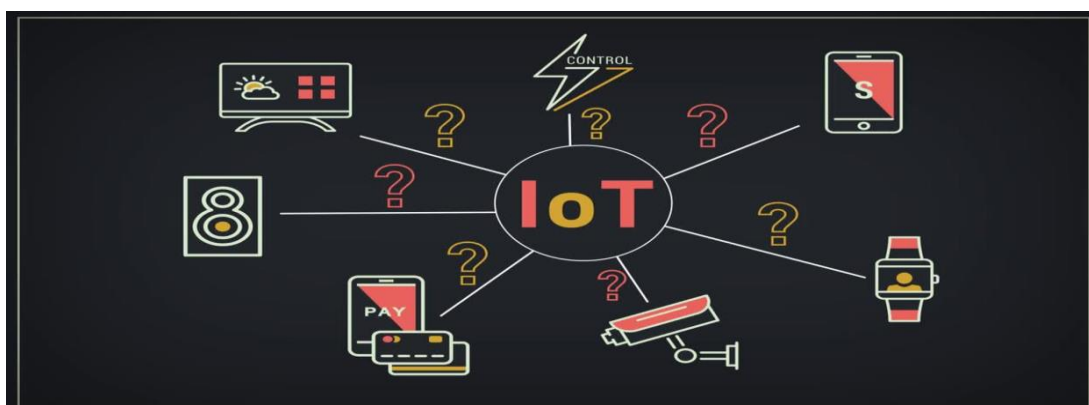


Figure 1.10: Les inconvénients de l'IdO

13 Sécurité de l'internet des objets

Les méthodes de sécurité de l'IdO varient en fonction de votre application spécifique de l'IdO et de votre place dans l'écosystème de l'IdO. Par exemple, les fabricants de l'IdO – des fabricants de produits aux entreprises de semi-conducteurs – doivent se concentrer sur l'intégration de la sécurité dès le départ, en rendant le matériel inviolable, en construisant du matériel sécurisé, en assurant des mises à jour sécurisées, en fournissant des mises à jour/rapports de micro-logiciels et en effectuant des tests dynamiques. Les développeurs de solutions doivent se concentrer sur le développement de logiciels sécurisés et l'intégration sécurisée. Pour ceux qui

déploient des systèmes IdO, la sécurité du matériel et l'authentification sont des mesures essentielles. De même, pour les opérateurs, il est essentiel de maintenir les systèmes à jour, d'atténuer les logiciels malveillants, de procéder à des audits, de protéger l'infrastructure et de sauvegarder la référence.

14 Conclusion

Dans un monde « hyperconnecté » via des objets connectés où les usagers sont à la fois émetteurs et récepteurs des données, l'IdO ouvre des champs nouveaux à explorer pour les sciences de l'information et de la communication pour étudier d'une part les enjeux sociétaux de ces nouveaux bouleversements technologiques et numériques et d'autre part analyser si les objets connectés répondent à des besoins d'utilisateurs de plus en plus exigeants en matière de service, de communication et d'information. L'Internet des objets doit être traité sous deux aspects d'une part de la réalité industrielle et technologique des objets connectés comme la gestion de l'entreprise, l'e-administration, l'e-gouvernement mais aussi gestuelle : podomètre, direction du regard, GPS, etc. et d'autre part les impacts des objets connectés dans la vie quotidienne dans les domaines suivants : la santé, l'habitat, l'automobile, l'assurance, etc.

Chapitre 2 :

Matériaux de l'IdO et les protocoles de communications

1 Matériel de l'internet des objets

1.1 ESP8266

L'ESP8266 est un circuit intégré avec un microcontrôleur permettant la connexion en Wi-Fi. Les modules intégrant ce circuit sont très utilisés pour contrôler des périphériques par Internet. L'ESP8266 est livré avec un firmware préinstallé vous permettant d'en prendre le contrôle à l'aide de « commandes AT » standards pouvant provenir d'une carte Arduino avec qui il peut communiquer par liaison série. Mais toute la souplesse et la puissance de ce module résident dans le fait que vous pouvez aussi y développer et flasher votre propre code, rendant ainsi le module entièrement autonome.



Figure 2.1 : ESP8266

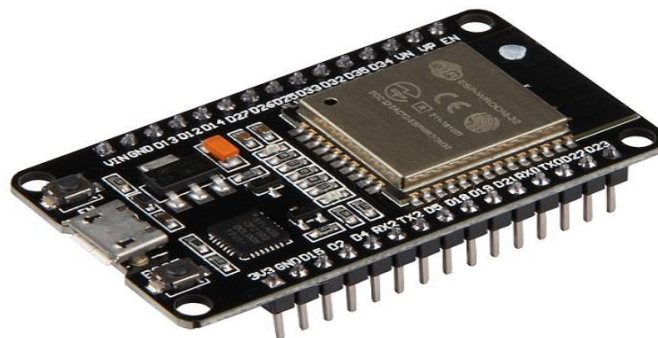
- Les caractéristiques du l'ESP8266 :
 - Alimentation:
 - 5 Vcc via micro-USB
 - 5 à 9 Vcc via broche Vin (régulateur intégré)
 - Microcontrôleur : ESP8266
 - Microprocesseur : Tensilica LX106
 - Fréquence : 80 MHz
 - Mémoire RAM : 64 kB
 - Mémoire Flash : 96 kB
 - 10 E/S digitales compatibles PWM
 - Interfaces : I2C, SPI, UART
 - Interface Wifi 802.11 b/g/n 2,4 GHz
 - Antenne intégrée

- Température de service : -40 à 125 °C
- Dimensions : 58 x 31 x 12 mm

1.2 ESP 32

ESP32 est un microcontrôleur avec des modules Wifi et Bluetooth intégrés. Très simple d'utilisation il est léger et possède une capacité de mémoire et de calculs supérieurs aux Arduino. Ce qui en fait une carte idéale pour l'apprentissage de la programmation, le développement d'objets connectés ou de serveur.

La base, pour créer des objets connectés, est de les connecter à un réseau comme le réseau WiFi. L'émetteur et l'antenne WiFi intégrés au microcontrôleur permette la connexion à Internet. Grâce à cela, il est possible de créer un serveur qui héberge une interface Web permettant de piloter le microcontrôleur ESP32 à distance.



- 2 x sorties analogiques (DAC)
- 15 x entrées analogiques (ADC)
- Interfaces : I2C, SPI, 2 x UART
- Interface Wi-Fi 802.11 b/g/n 2,4 GHz
- Bluetooth : Classique / BLE
- Antenne intégrée
- Température de service : -40 à 125 °C
- Dimensions : 48 x 26 x 11,5 mm

1.3 Comparaison entre ESP32 et ESP8266

	ESP8266	ESP32
MCU	Xtensa Single-core 32-bit	Xtensa Dual-Core 32-bit
Wi-Fi	HT20 (802.11 b/g/n)	HT40 (802.11 b/g/n)
Bluetooth	Non	Oui (4.2 et BLE)
Fréquence	80Mhz	160 a 240 Mhz
SRAM	64kB	520kB + 10 pour RTC
DRAM	96kB	328kB
Flash	4mB	4mB (suivant versions)
GPIO	17	36
Analog GPIO	1	18

Tableau 2: Comparaison entre ESP32 et ESP8266

- Pour résumer, au niveau matériel, l'esp32 est plus rapide que l'esp8266, il dispose de plus de GPIO, supporte plus d'interruptions et dispose de plus de chaînes analogiques.

1.4 Les Capteurs

1.4.1 Définition d'un capteur

Un capteur est un dispositif qui permet de convertir une grandeur physique observée (température, luminosité, humidité, débit, présence d'objet...) en une grandeur normée et utilisable, généralement un signal électrique (courant, tension, niveaux logiques, valeur moyenne, fréquence, amplitude, nombre binaire...), qui peut être interprétée par un dispositif de contrôle commande.

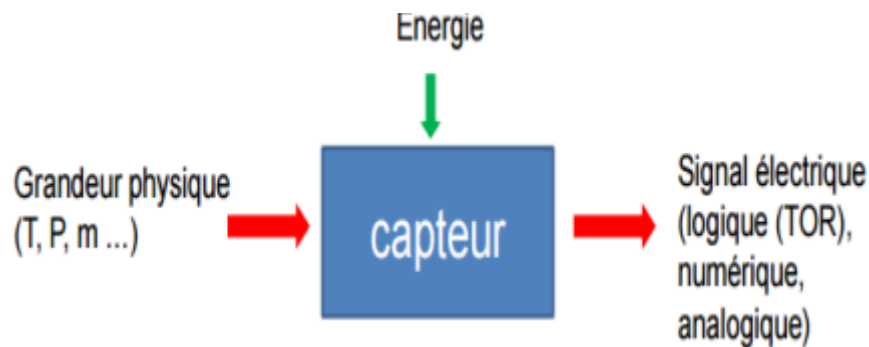


Figure 2.3: Principe de fonctionnement d'un capteur

Les capteurs jouent un rôle indispensable dans le domaine d'IoT, ils peuvent être classés selon deux critères à savoir :

- Grandeur mesurée : nous distinguons par exemple un capteur de température, de position, d'humidité, de vitesse, de force, de pression, etc.
- Le signal délivré : nous trouvons par exemple des capteurs logiques (appelés aussi capteurs TOR11), capteurs analogiques et numériques.
- Capteurs logiques : La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs exemple : capteurs de fin de course.
- Capteurs analogiques : La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur.
- Capteur numérique : La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type : code numérique binaire, train d'impulsions, etc.

1.4.2 Les caractéristiques d'un capteur

Les capteurs sont distingués par les caractéristiques suivantes :

- Étendue de mesure (Pleine Echelle) : L'étendue de mesure est la différence entre la limite supérieure et la limite inférieure de la grandeur mesurable par un capteur. Lorsque le capteur fournit une valeur de la grandeur entre 0 et le maximum, ce maximum est appelé « Pleine Echelle ».
- Résolution : plus petite variation de grandeur mesurable par le capteur.
- Rapidité : temps de réaction du capteur. La rapidité est liée à la bande passante.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Sensibilité : représente la variation du signal de sortie par rapport à la variation du signal d'entrée.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure.
- Finesse : Elle permet d'estimer l'influence que peut avoir le capteur et de son support ou de ses liaisons sur la grandeur à mesurer. Par exemple, dans le cas d'un capteur de température, une capacité calorifique importante réduit sa finesse
- Fréquence de résonance : Un capteur possède une réponse qui peut dépendre de la fréquence de la grandeur mesurée. Lorsqu'il existe une fréquence à laquelle la réponse est particulièrement élevée, celle-ci est appelée fréquence de résonance.

1.4.3 Critères de choix du capteur

Le choix d'un capteur peut être adopté des points suivants :

- Le type de l'évènement à détecter
- LA nature de cet évènement
- La grandeur de l'évènement
- L'environnement de l'évènement
- L'efficacité, la qualité et la précision du capteur
- La disponibilité dans le marché.

1.5 Les actionneurs

Un actionneur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Nous pouvons classer les actionneurs selon les trois critères suivants :

- **Energie utilisée** : pneumatique, hydraulique, électrique, mécanique, combustible fossile.
- **Phénomène physique utilisable** : déplacement ou freinage, chaleur ou Froid, lumière, son.
- **Principe mis en œuvre** : pneumatique (les propriétés de compression et de dilatation sont utilisées), hydraulique (la propriété d'incompressibilité du fluide est utilisée).

Cette électrovanne, qui est normalement fermée au repos, fonctionne en 12V grâce à un solénoïde qui actionne l'ouverture du flux. Elle est idéale pour vos projets comme de l'arrosage automatique, un mur d'eau, ou tout autre projet nécessitant un contrôle de flux comme l'eau.

❖ **Caractéristiques :**

- Tension de travail : 12VDC
- Courant max : 450mA
- Mode de fonctionnement t: normalement fermé
- Taille d'entrée / sortie : G1 / 2 "
- Matériaux : nylon / acier inoxydable / Polyoxyméthylène
- Type Valve : membrane de la vanne (exploité par servo)
- Filtre : filtre d'entrée en acier inoxydable
- Température de fonctionnement maximum : 120 ° C
- plage de pression de fonctionnement: de 0,02 à 0,8 MPa
- Durée de vie prévue: 200,000+

2 La communication dans un système de l'IdO

2.1 Technologies de communication

Certains objets fixes peuvent être connectés par des réseaux filaires, la croissance de l'internet des objets devrait majoritairement être portée par l'utilisation de technologies sans fil et mobiles. Les technologies de connectivité sans fil sont nombreuses et variées, et l'usage de l'une ou l'autre est souvent avant tout décidé par la portée du réseau envisagé. Certains cas d'usage nécessitent également l'association de technologies sans fil et filaires pour relier les équipements à Des réseaux privés étendus ou à Internet. En bref, nous distinguons dans les méthodes de transmission de données entre les objets connectés, deux solutions : filaire et sans fil [7]

1.1.1 Connectivité filaire

Le réseau filaire est un réseau qui comme son nom l'indique est un réseau que l'on utilise grâce à une connexion avec fil. Ce réseau utilise des câbles Ethernet pour relier des ordinateurs et des périphériques grâce à un routeur ou à un commutateur. C'est grâce aux réseaux filaires que l'on obtient les débits les plus élevés et les plus stables. Nous donnons comme exemples : Ethernet, DSL, ADSL, SDSL, le câble, Fibre Optique (FTTH, FTTB, FTTD...), CPL, etc.

1.1.2 Connectivité sans fil

Les réseaux sans fil permettent de relier des ordinateurs et d'autres appareils informatiques sans avoir à installer de câblage, ce qui représente plus de confort et fait économiser de l'argent au niveau des infrastructures. En revanche, ce genre de réseaux a généralement moins de sécurité car, sans protection efficace, l'introduction d'intrus est probable

Selon le type de couverture, le réseau sans fil peut être classé comme WPAN, WLAN, WMAN ou WAN.

Le réseau WPAN est commun concernant la technologie Bluetooth tandis que les systèmes Wi-Fi établissent généralement des réseaux WLAN.

Les technologies à base de WiMax, à leur tour, permettent d'établir des réseaux WMAN tandis que les réseaux.

WWAN sont utilisés avec des communications GSM, GPRS (2G) ou UMTS (3G)

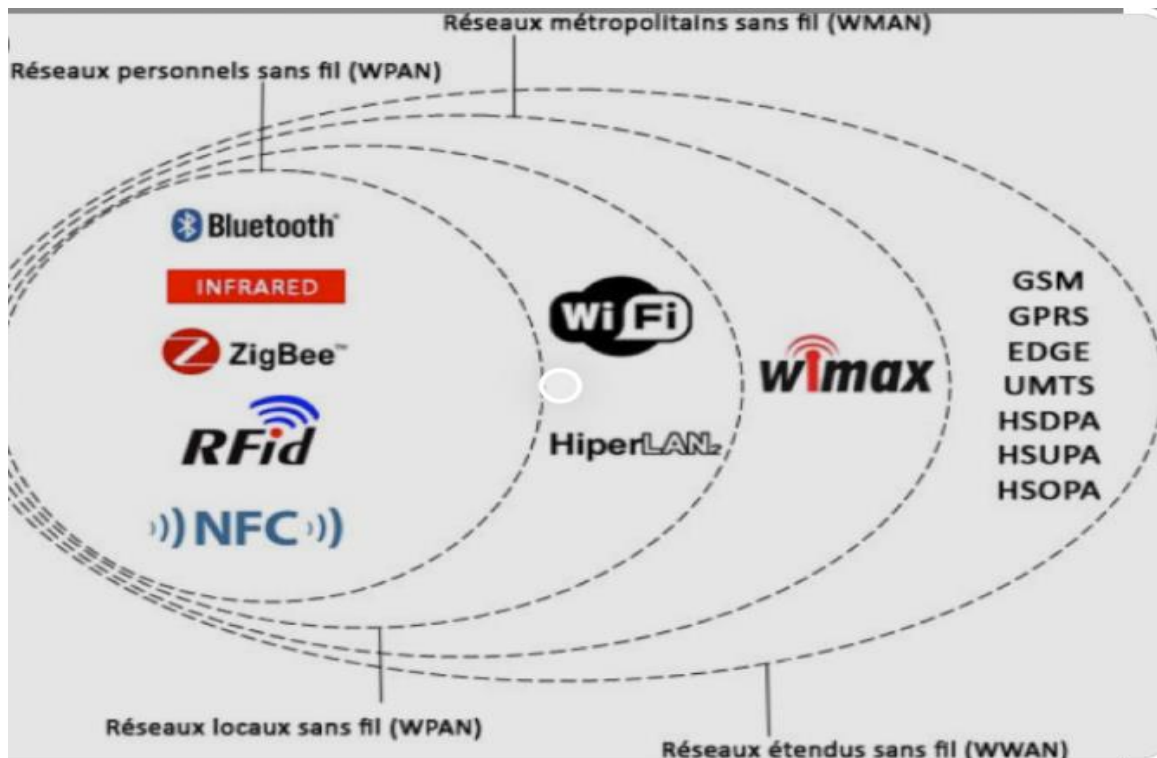


Figure 11.5: Les différents réseaux

Il existe de nombreuses options de technologies sans fil utilisées dans l'IdO notamment : Bluetooth, Zigbee, Z-wave, 6LowPAN, Thread, Wi-Fi, outre la technologie cellulaire tel que : NFC, Sigfox, Neul, Lora WAN.

Nous présentons quelques-unes d'eux :

1.1.2.1 Bluetooth :

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

- Norme : la spécification fondamentale de Bluetooth4.2
- Fréquence : 2,4 GHz(ISM).
- Portée : 50-150 m(Smart/BLE)
- Vitesses de transmission : 1 Mbit/s(Smart/BLE)

1.1.2.2 Zigbee :

Comme le Bluetooth, ZigBee dispose d'une importante base d'exploitation installée, surtout en milieu industriel. Parmi les profils ZigBee disponibles, ZigBee PRO et

ZigBeeRemote Control (RF4CE) reposent sur le protocole IEEE802.15.4 ; fonctionnant à 2,4 GHz, cette technologie de réseau sans fil standard de l'industrie cible les applications nécessitant des échanges de données relativement peu fréquents à de faibles vitesses de transmission sur un espace restreint et dans une portée de 100 m (résidence ou bâtiment...).

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

1.1.2.3 Wi-Fi :

La connectivité Wi-Fi s'impose souvent comme le choix évident pour beaucoup de développeurs. Elle ne nécessite pas de longues explications, sauf pour rappeler l'évidence, à savoir la vaste infrastructure existante, le transfert de données rapide et la possibilité de gérer de grandes quantités de données. À l'heure actuelle, la 802.11n s'impose comme la norme Wi-Fi la plus utilisée dans le contexte privé et professionnel. Cette norme offre un débit élevé, de l'ordre de centaines de mégabits par seconde, idéal pour les transferts de fichiers, mais peut-être trop énergivore pour la plupart des applications IdO.

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

1.1.2.4 Etude comparatif entre les trois protocoles utilisés

Le tableau récapitulatif ci-dessous compare les trois principaux protocoles utilisés dans l'IdO. Il démontre que ces typologies présentent différents avantages et inconvénients, et leur utilisation dépend du contexte dans lequel il se trouve

caractéristiques	Zigbee	Bluetooth	wifi
IEEE	802.15.4	802.15.1	802.11 /a/b/g/n/ac
Autonomie avec pile	années	Mois	Jours
Topologie	Maille	Machine a machine	Etoile
Vitesse de transfert	250KbpS	1mbpS	1000mbps
Portée	100 m	10 m	300 m

Tableau 3 : Les caractéristiques des réseaux sans fil

2.2 Les protocoles de communication

Pour répondre à la problématique du nombre grandissant d'objets connectés qui vont faire leur apparition sur la toile dans les années à venir (selon une étude Gartner : c'est près de 26 milliards d'objets connectés qui seront sur Internet d'ici 2020), et pour le but de faire dialoguer et d'automatiser leur comportement du quotidien pour qu'ils fournissent des services ou envoient des informations sur leur état ou leur environnement afin d'en extraire de la valeur. L'IdO s'est doté des standards :

- **HTTP** : signifie Hypertext Transfer Protocol (protocole de transfert hypertexte). Ce protocole définit la communication entre un client et un serveur sur le World Wide Web.
- **MQTT** : signifie Message Queuing Telemetry Transport. Il utilise le principe de « Publisher / Subscriber » pour connecter les systèmes entre eux.
- **AMQP**: signifie Advanced Message Queuing Protocol. Un protocole pour les systèmes de messagerie orientés messages(MOM).
- **STOMP**: signifie (Simple TextOriented Messaging Protocol). Est un protocole textuel audessus de TCP conçu pour permettre l'interaction avec un middleware orienté messages.

2.2.1 Comparaison entre les protocoles

Il y'a beaucoup de protocole pour l'IdO mais le meilleur protocole pour tous les scénarios ne sont pas existant. Le choix du protocole dépend au scénario. Certains protocoles ont plus de fonctionnalités que les autres, mais n'est pas capables pour le projet ne peut pas être bon choix.

Un système complexe peut utiliser plusieurs protocoles de profiter de leurs avantages.

Plusieurs standards peuvent être utilisé pour évaluer protocole que la performance, robuste, scalaire, la sécurité, mais toujours besoin d'équilibre entre des exigences et des performances de l'appareil.

HTTP est robuste, omniprésent mais très lourd pour les appareils à faible rendement.

CoAP est conçu pour le réseau de contrainte avec bouteille, sécurité et facilement applicable à http application, mais le modèle REST lorsque les capteurs sont généralement serveur ne semble pas efficace comme ils ont besoin pour traiter la demande et la réponse.

MQTT est mieux en comparaison avec CoAP, car son en-tête est plus petit mais fiable en raison de la QoS. Le cryptage des données assure que les informations sont en sécurité.

Publier / Souscrire modèle permet à chaque utilisation de l'appareil seulement leur propre tâche (publier ou de souscription).

Client ne dispose pas d'envoyer la demande à des capteurs pour récupérer les données. Ce système permet d'éviter problème potentiel.

Chapitre 3 :

Le protocole MQTT

et les plateformes

IdO

1 Le protocole MQTT

1.1 Définition

MQTT, pour "Message Queuing Telemetry Transport", est un protocole open source de messagerie qui assure des communications non permanentes entre des appareils par le transport de leurs messages. Il a été créé en 1999 par Andy Stanford-Clark, ingénieur chez IBM, et Arlen Nipper, chez EuroTech, principalement dans la communication M2M pour permettre à deux appareils utilisant des technologies différentes de communiquer. "Devenu une norme ISO en 2016, MQTT connectait déjà à cette date des millions d'appareils dans le monde entier, dans toutes sortes d'applications et d'industries. C'est une technologie d'avenir", affirme Fabien Pereira Vaz, technical sales manager chez Paessler AG. Les géants du web parmi lesquels AWS ou Microsoft utilisent MQTT pour remonter les données sur leur plateforme cloud. [8]

1.2 Mode de fonctionnement

MQTT est un protocole standardisé reposant sur TCP/IP. "Il est particulièrement utilisé pour transporter des données des objets connectés sur le cloud. Néanmoins, il reste lourd pour les réseaux LPWA (Low-Power Wide-Area) contraints type NB-IoT et ne permet d'adresser que la remontée de données sans prendre en compte les services de device management. MQTT permet la gestion des déconnexions et des reconnexions de devices de manière simplifiée. La taille maximale d'un message envoyé avec MQTT est de 256 Mo.

1.2.1 Topologies MQTT

Le standard ne définit pas de topologie particulière d'utilisation, la plus simple étant celle avec un seul broker et un certain nombre de clients. D'autres organisations restent évidemment possible, avec en particulier les systèmes à base de bridge dans lesquels plusieurs brokers sont utilisés.

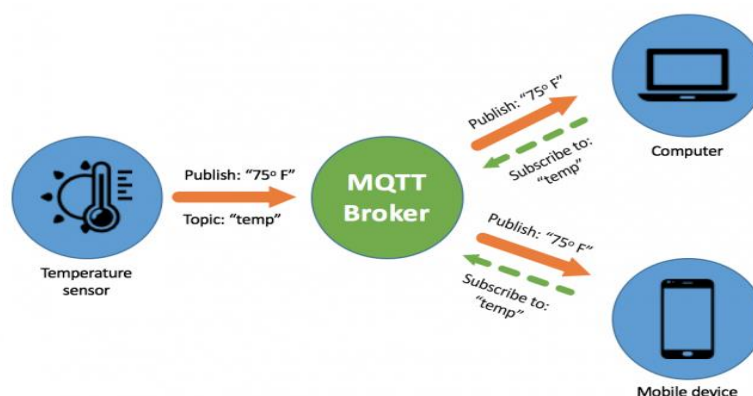


Figure 12.1 : Topologie MQTT

1.2.2 Fonctionnement

1.2.2.1 Connexion et Déconnexion

MQTT se sert de connexions persistantes entre les clients et le broker, et exploite pour cela les protocoles de réseau garantissant un bon niveau de fiabilité comme TCP.

Avant de pouvoir envoyer des commandes de contrôle, un client doit au préalable enregistrer auprès du broker, ce qui se fait avec la commande **CONNECT**.

Divers paramètres de connexion peuvent alors être échangés, comme les identifiants du client ou encore le mode de persistance souhaité. Le broker doit confirmer au client que son inscription a bien été prise en compte, soit indiquer qu'une erreur est survenue en renvoyant un **CONNACK** accompagné d'un code de retour

Il existe une commande **PINGREQ** permettant de faire savoir au broker que le client est toujours actif, le broker répondra de son côté avec un **PINGRESP** pour indiquer au client que la connexion est toujours active.

Lorsque le client veut se déconnecter, il envoie au préalable une commande **DISCONNECT** au broker pour lui faire part de son intention. Dans le cas contraire, le broker considérera la déconnexion comme anormale et agira en conséquence, en envoyant le message de volonté **WILL** au nom du client à tous les abonnés. Un message que le client peut spécifier et conservera l'indicateur dans la charge utile de message **CONNECT**. [7]

1.2.2.2 Abonnements et Publications

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

:

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

Si la qualité de service requise est supérieure à zéro, des messages seront échangés pour confirmer la prise en charge de la publication.

1.2.3 Qualité de service

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

❖ Au plus une fois

QoS=0

- Le message est distribué une fois tout au plus, ou n'est pas distribué du tout. Sa distribution via le réseau n'est pas accompagnée d'un accusé de réception.
- Le message n'est pas stocké. Le message peut être perdu si le client est déconnecté ou si le serveur échoue.
- QoS=0 est le mode de transfert le plus rapide. Il est parfois appelé "autonome après diffusion".
- Le protocole MQTT ne nécessite pas que les serveurs envoient les publications avec QoS=0 au client. Si le client est déconnecté au moment où le serveur reçoit la publication, cette dernière peut être supprimée en fonction du serveur. Le service de télémétrie (MQXR) ne supprime pas les messages envoyés avec QoS=0. Ils sont stockés en tant que messages non persistants et sont uniquement supprimés en cas d'arrêt du gestionnaire de files d'attente.

❖ Au moins une fois

QoS=1

- QoS=1 est le mode de transfert par défaut.
- Le message est toujours distribué au moins une fois. Si l'expéditeur ne reçoit pas d'accusé de réception, le message est renvoyé avec l'indicateur DUP défini jusqu'à la réception de l'accusé de réception. En conséquence, le destinataire peut recevoir le même message à plusieurs reprises, et le traiter plusieurs fois.
- Le message doit être stocké localement au niveau de l'expéditeur et du destinataire jusqu'à ce qu'il soit traité.
- Le message est supprimé du destinataire après son traitement. Si le destinataire est un courtier, le message est publié pour ses abonnés. Si le destinataire est un client, le message est distribué à l'application de l'abonné. Une fois que le message est supprimé, le destinataire envoie un accusé de réception à l'expéditeur.
- Le message est supprimé de l'expéditeur une fois qu'il a reçu un accusé de réception de la part du destinataire.

❖ Une seule fois

QoS=2

- Le message est toujours distribué une seule fois.
- Le message doit être stocké localement au niveau de l'expéditeur et du destinataire jusqu'à ce qu'il soit traité.
- QoS=2 est le mode de transfert le plus sûr, mais le plus lent. Il faut au moins deux paires de transmissions entre l'expéditeur et le destinataire avant la suppression du message côté expéditeur. Le message peut être traité côté destinataire après la première transmission.
- Dans la première paire de transmissions, l'expéditeur transmet le message et reçoit un accusé de réception du destinataire indiquant qu'il a stocké le message. Si l'expéditeur ne reçoit pas d'accusé de réception, le message est renvoyé avec l'indicateur DUP défini jusqu'à la réception de l'accusé de réception.
- Dans le cadre de la seconde paire de transmissions, l'expéditeur dit au destinataire qu'il peut terminer le traitement du message, "PUBREL". Si l'expéditeur ne reçoit pas d'accusé de réception du message "PUBREL", le message "PUBREL" est renvoyé jusqu'à la réception de l'accusé de réception. L'expéditeur supprime le message qu'il a enregistré lors de la réception de l'accusé de réception au message "PUBREL"
- Le destinataire peut traiter le message lors de la première ou de la seconde phase à condition de ne pas traiter à nouveau le message. Si le destinataire est un courtier, il publie le message pour les abonnés. Si le destinataire est un client, il livre le message à l'application de l'abonné. Le destinataire renvoie un message d'achèvement à l'expéditeur indiquant qu'il a terminé le traitement du message.

1.2.4 Sécurité

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

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

Un client authentifie un broker à l'aide du protocole SSL. Un broker MQTT authentifie un client à l'aide du protocole SSL, d'un mot de passe, ou des deux.

1.3 Description informelle du protocole

Dans cette section, nous définissons le format binaire des paquets spécifié par la norme MQTT.

1.3.1 Représentations de données

- ❖ **Bits** : Les bits dans un octet sont étiquetés de 7 à 0. Le bit numéro 7 est le bit le plus important, le bit le moins significatif est affecté au bit 0.
- ❖ **Valeurs de données entières** : Les valeurs de données entières sont 16 bits : l'octet d'ordre supérieur précède l'octet d'ordre inférieur. Cela signifie qu'un mot de 16 bits est présenté sur le réseau comme octet le plus significatif (MSB), suivi de l'octet le moins important (LSB).
- ❖ **Chaînes codées enUTF-8** : Les chaînes de caractères doivent toutes être encodées en utf8, et préfixées par leur longueur sur deux octets, ces chaînes sont donc limitées à une longueur de $2^{16} - 1 = 65535$ octets.

1.3.2 Structure d'un paquet de contrôle MQTT

Le protocole MQTT fonctionne en échangeant une série de paquets de contrôle MQTT d'une manière définie. Cette section décrit le format de ces paquets. Un paquet de contrôle MQTT comprend jusqu'à trois parties, toujours dans l'ordre suivant:

- Entête fixe
- CONNECT
- CONNACK
- PUBLISH
- SUBSCRIBE
- SUBACK
- UNSUBSCRIBE
- UNSUBACK
- PINGREQ, PINGRESP
- DISCONNECT

1.3.3 La synchronisation des requêtes envoyées au broker

Le tableau suivant décrit les méthodes du client MQTT qui envoient une demande au broker.

À moins que le client d'application fixe un délai d'attente indéfinie, le client attend indéfiniment jamais pour le broker. Si le client se bloque, il est soit un problème de programmation d'application, ou un défaut dans le client MQTT.

Méthode	Synchronisation	intervalle d'attente	délai
Connect	Temps d'attente pour une connexion à établir avec le broker	La valeur par défaut de 30 secondes, ou tel que défini par un paramètre, lance alors une exception	
Disconnect	Temps d'attente pour le client MQTT pour terminer les travaux qu'il doit faire, et pour la session TCP / IP pour déconnecter		
Subscribe UnSubscribe	Attend la fin de la méthode abonner ou désabonner		
Publish	Retourne immédiatement au fil d'application après avoir passé la demande au client MQTT.	Aucun	

Tableau 4 : Comportement de synchronisation des méthodes qui donnent lieu à des requêtes au broker

1.3.4 L'identifiant de client MQTT

Le protocole MQTT définit un « identifiantclient » (ID client) qui identifie de manière unique un client dans un réseau. En termes simples, lors de la connexion à un broker, un client doit spécifier une chaîne unique qui n'est pas utilisée actuellement et ne sera pas utilisée par un autre client qui se connectera au broker MQTT. Il existe plusieurs façons de choisir un identifiant client. Voici quelques exemples :

- Un capteur installé dans un emplacement particulier peut utiliser le code d'emplacement comme ID du client.

- Un appareil mobile possédant une capacité de réseau peut choisir l'adresse MAC ou un identifiant d'appareil unique comme ID de client mais MQTT restreint la longueur d'ID de client à 23 caractères. Par conséquent, il y aura une situation où l'identification du client doit être raccourcie.

En raccourcissant l'identifiant du client, il faut s'assurer que l'ID client n'est pas identique à toute autre ID client utilisée dans le réseau. Afin de garder l'identifiant court et unique, il faut présenter un mécanisme de génération d'identifiant fiable. Par exemple, créer un identifiant client à partir du périphérique MAC 48 bits adresse. Si la taille de la transmission n'est pas un problème critique, et utiliser les 17 octets restants pour rendre l'adresse plus facile à administrer, comme un texte lisible par l'utilisateur dans l'identifiant.

Maintenant, essayons de comprendre les implications de deux clients obtenant le même identifiant client.

Le broker MQTT surveille les messages en attente d'être envoyés à un client en fonction de l'identifiant du client. Ainsi, si un client utilise QoS 1 ou QoS 2 et abonné à n'importe quel sujet et déconnecté du broker, le broker sauvera les messages arrivés pour le client pendant qu'il était déconnecté. Une fois que le client se reconnecte, le broker enverra ces messages au client. Si un autre appareil MQTT utilise le même ID de client et se connecte au broker, le broker enverra les messages qu'il avait sauvegardés.

Un autre scénario lié aux identifiants clients est la connexion en double. Disons qu'un périphérique particulier à l'aide de l'ID client DeviceA est connecté au courtier. Si un autre client vient avec le même ID de client DeviceA, le broker peut décider s'il doit permettre au nouveau client de se connecter et de déconnecter le client existant ou de garder l'ancienne connexion en vie et interdire le nouveau client. Il s'agit d'une fonctionnalité facultative d'un broker MQTT.

1.3.5 Callbacks

Le modèle de programmation client MQTT utilise des threads en profondeur. Les threads découplent une application client MQTT, autant qu'ils le peuvent, des retards dans la transmission de messages vers et depuis le broker. Les publications, les jetons de livraison et les événements perdus de connexion sont livrés aux méthodes d'une classe de rappel qui implémente MQTTCallback. [9]

2 Les plateformes de l'IDO

2.1 Définition d'une plateforme

Une plateforme d'IDO est un ensemble de services permettant de collecter, stocker, corréler, analyser et exploiter les données.

Une plateforme est donc un ensemble d'APIs et drivers de communication, de bases de données, de services de traitements et de calculs et bien souvent un web service pour générer des tableaux de bords de visualisation et d'exploitation.

Afin de rester compétitives, les entreprises doivent innover et créer davantage de ² valeur pour augmenter leur chiffre d'affaire ou pour améliorer leur productivité.

La mise en place d'une plateforme IDO permet notamment de :

- Proposer de nouveaux services à vos équipes métiers et/ou à vos clients
- De mieux surveiller votre parc d'objets connectés
- De vous fournir des indicateurs pour faciliter vos prises de décision

2.2 Choix d'une plateforme

Une plateforme d'IDO est un ensemble de services et chaque plateforme dispose d'avantages spécifiques en fonction de leurs services. Selon nous, il existe deux types de plateformes :

- Les propriétaires,
- Les open-source.

Nous allons peser le pour et le contre de chacune de ces solutions pour que vous puissiez déjà orienter vos choix. Ensuite nous développerons les caractéristiques essentielles et les points de vigilance à retenir.

2.3 Plateforme Propriétaire vs. Open-Source

Communément, les plateformes **propriétaires** sont des solutions de type PaaS (Plateforme As A Service). La plupart du temps, ces plateformes sont hébergées dans le Cloud. Aussi, le modèle économique de cette solution est une souscription

récurrente à un service (OPEX). L'avantage de ce modèle est le partage de responsabilité puisque finalement, c'est votre prestataire qui a la charge de la fourniture et du maintien opérationnel des environnements.

A contrario, les solutions **Open-source** sont des plateformes qui demandent des connaissances et des ressources assez conséquentes. En effet, avec ce type de plateforme, c'est à vous de développer l'ensemble des services, parfois aider par la communauté, mais aussi de maintenir les outils, infrastructure et software. D'ailleurs, d'un point de vue time-to-market cela peut rallonger les délais de livraison. Cependant, vous conserver une totale maîtrise de l'outil.

	Avantages	Inconvénients
Propriétaires	<ul style="list-style-type: none"> ➤ Risques partagés ➤ Maintien en condition opérationnelle externalisé ➤ Modèle économique OPEX 	<ul style="list-style-type: none"> ➤ Pas de maîtrise sur les développements ➤ Pérennité du prestataire
Open - source	<ul style="list-style-type: none"> ➤ Maîtrise des outils ➤ Localisation des environnements 	<ul style="list-style-type: none"> ➤ Besoin en ressources important ➤ Coût supporté à l'investissement

Tableau 5: Les avantages et les inconvénients des plateforme Propriétaires et Open-source

2.4 Critères du choix d'une plateforme

2.4.1 La gestion des devices

Les modèles de capteurs et d'objets connectés se multiplient et se diversifient. Une plateforme IdO doit ainsi avoir vocation à connecter et gérer l'ensemble de ces appareils hétérogènes sur une interface unique.

Le volume exponentiel d'objets à gérer engendre aussi un nouveau besoin. Celui de pouvoir créer sans contraintes des interactions entre ces objets hétérogènes, d'où le critère essentiel d'interopérabilité à prendre en ligne de compte.

2.4.2 La gestion de la connectivité

L'agnosticité technologique est un premier élément essentiel à prendre en compte. Aujourd'hui les protocoles de communication évoluent et se multiplient. Pour autant, cette hétérogénéité des réseaux ne doit pas être un frein. Aussi est-il impératif pour une plateforme IdO de pouvoir gérer l'ensemble de ces réseaux. **3G/4G, LoRa, Sigfox**, LTE-M, Narrow Band... autant de solutions qui permettent de connecter les objets. Une plateforme IdO se doit donc de supporter un large éventail de standards de communication.

2.4.3 Visualisation de la donnée

La capacité d'une plateforme IdO à stocker, analyser et rediriger les données vers des interfaces de visualisation dédiées est un point majeur.

La qualité des algorithmes permettant une analyse fine de la donnée, l'éventail des possibilités des tableaux de bords pour visualiser l'information... sont autant de facteurs à prendre en compte. Toute la valeur est en effet dans la capacité à capter des données brutes pour les restituer en informations intelligentes. De plus, la plateforme IdO se doit de s'adapter à l'utilisateur en offrant une expérience utilisateur personnalisée et optimisée, et non l'inverse comme on le retrouve souvent sur les solutions existantes.

2.4.4 Sécurité des objets et des données

Une plateforme IDO doit garantir la sécurité et la confidentialité des données de bout en bout, de la captation jusqu'à la restitution.

Enfin la question de la sécurité est un enjeu majeur. Protéger les données sensibles, empêcher les hackers de pirater les systèmes d'information, la pression est croissante dans le domaine. Aussi est-il nécessaire de choisir une plateforme IoT qui va garantir un haut niveau de sécurité et permettre d'atténuer les risques. Une sécurisation de bout en bout en somme de la couche matérielle à la couche logicielle.

2.4.5 Le degré d'agilité

Au regard de l'évolution sans cesse plus rapide des types d'objets, des technologies... être agile, fonctionner en mode itératif, être évolutif est une nécessité. Une plateforme IdO doit avoir la capacité de s'adapter rapidement et garantir une solution pérenne à ses utilisateurs. Elle doit assurer en somme d'un retour sur investissement rapide pour les entreprises sans avoir besoin d'investir massivement dans une solution qui mettra du temps à être rentabilisée.

Toute porte à croire que la plateforme IdO parfaite serait une plateforme hybride à la croisée entre plateforme ouverte, qui nécessite une personnalisation et donc un certain temps dans sa mise en œuvre, et plateforme fermée qui présente une rigidité certaine.

Chapitre 4 :

Applications WSN

1 INTRODUCTION

Les réseaux de capteurs sans fil (RCSFs) sont constitués d'un grand nombre de noeuds de capteurs qui sont généralement alimentés par batterie et conçu pour fonctionner pendant une grande période. Les domaines d'application sont nombreux et variés, tel que le domaine environnementale, médicale et militaire. L'atout majeur de ce dispositif est un déploiement à grande échelle sans aucune maintenance.

Les capteurs n'ont pas besoin d'une infrastructure établie pour parvenir à transmettre des données vitales à l'étude de l'environnement. Il est nécessaire également de garantir une bonne qualité de service, car les réseaux de capteurs sans fils doivent intégrer des mécanismes qui permettent aux utilisateurs de prolonger la durée de vie du réseau en entier, car chaque nœud est alimenté par une source d'énergie limitée et généralement irremplaçable. C'est pourquoi, il est nécessaire d'optimiser la consommation d'énergie à tous les niveaux de conception de ce type de réseau. Par conséquent, la minimisation de la consommation d'énergie est un facteur de conception des plus importants dans les réseaux de capteurs

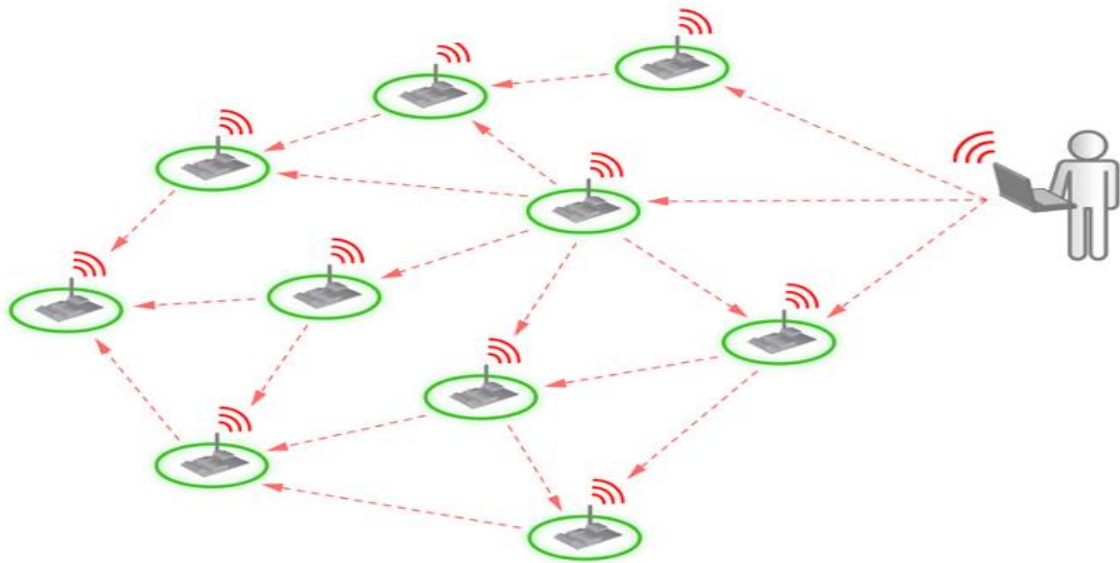


Figure 4.1: Un réseau

2Interface Arduino IDE

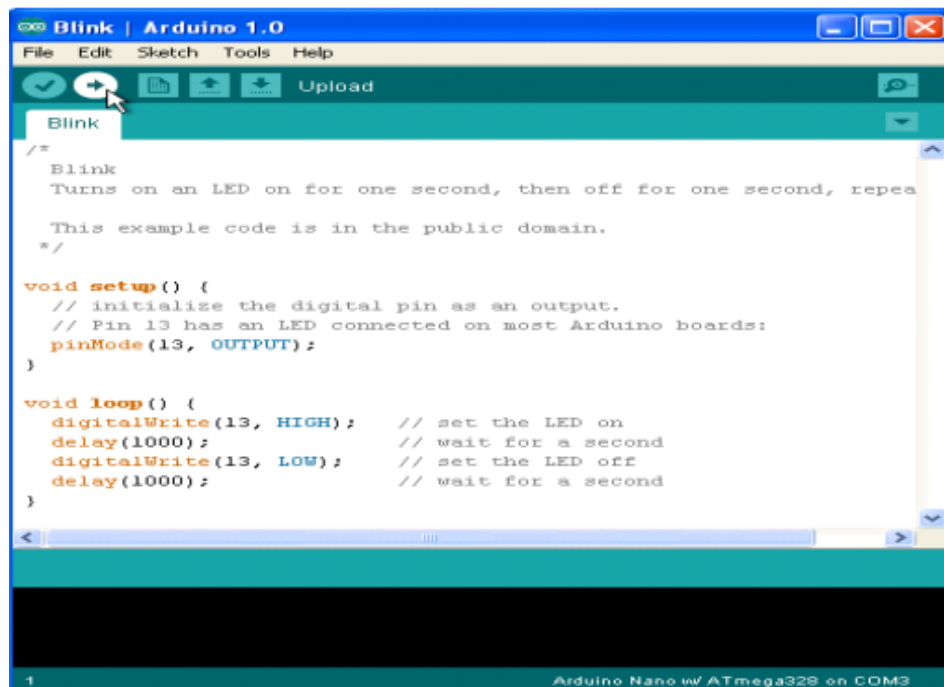
Arduino fournit un environnement de développement (IDE) avec un éditeur de source, les opérations de compilation et de chargement dans la mémoire du microcontrôleur étant ramenées à des clics sur des boutons dans l'IDE (très simple). La communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par Arduino).

Structure d'un projet Arduino L'outil impose de structurer l'application de façon spécifique. Le compilateur utilisé est AVR GCC (compilateur C/C++ pour processeur AVR).

Le programme principal (fonction main) est imposé, non modifiable, et décrit ci-dessous.

Les seules parties que l'on développe spécifiquement sont :

- **La fonction Setup ()** : doit contenir les initialisations (times, interruptions...)
- **La fonction Loop ()** : fonction répétée indéfiniment.



```
Arduino 1.0
File Edit Sketch Tools Help
Upload
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeats.
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}

1 Arduino Nano w/ ATmega328 on COM3
```

Figure 4.2 : Fonction Setup et Loop

3 Communication entre l'interface et la carte

3.1 Bibliothèques

Le logiciel Arduino, basé sur le langage C++, possède des bibliothèques de développement riches. L'exécution s'effectue d'une manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

Avant d'ajouter les bibliothèques qu'on veut utiliser, on est obligé de certifier notre carte ESP32 avec l'interface IDE.

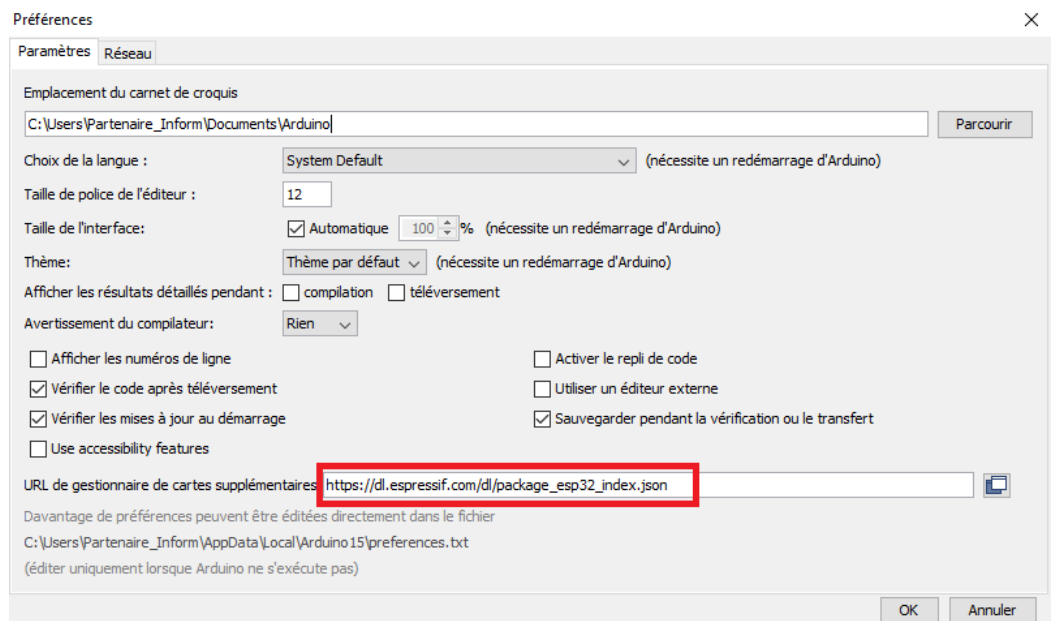


Figure 4.3: Paramètre Arduino BLE

Cette opération va nous permettre de trouver les bibliothèques souhaitées dans le gestionnaire.

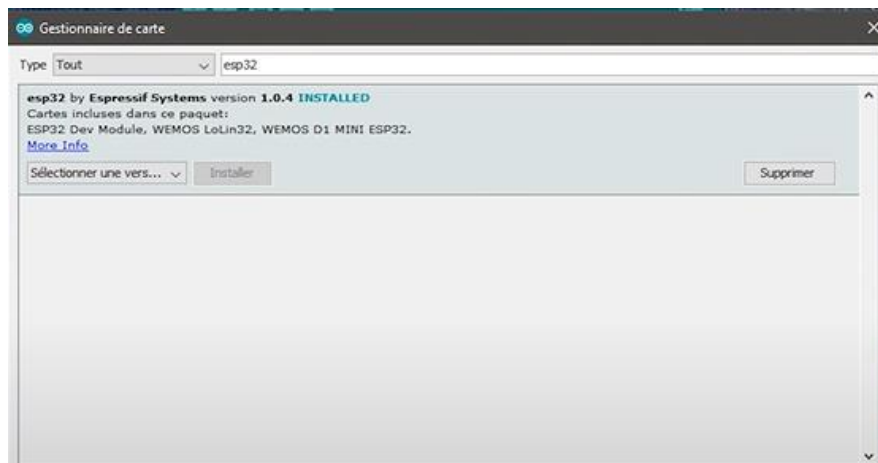


Figure 4.4: Gestionnaire de la carte

4 Serveur et client BLE

4.1 Définition du BLE

Bluetooth LowEnergy, BLE en abrégé, est une variante de Bluetooth à économie d'énergie. L'application principale du BLE est la transmission à courte distance de petites quantités de données (faible bande passante). Contrairement au Bluetooth qui est toujours activé, le BLE reste en mode veille en permanence, sauf lorsqu'une connexion est initiée.

Cela fait qu'il consomme très peu d'énergie. BLE consomme environ 100 fois moins d'énergie que Bluetooth (selon le cas d'utilisation).

4.2 Caractéristiques

La caractéristique appartient toujours à un service et c'est là que les données réelles sont contenues dans la hiérarchie (valeur). La caractéristique a toujours deux attributs : la déclaration de caractéristique (qui fournit des métadonnées sur les données) et la valeur de caractéristique.

De plus, la valeur caractéristique peut être suivie de descripteurs, qui élargissent davantage les métadonnées contenues dans la déclaration de caractéristique.

Les propriétés décrivent comment interagir avec la valeur caractéristique. Fondamentalement, il contient les opérations et les procédures qui peuvent être utilisées avec la caractéristique :

- Diffuser
- Lire
- Ecrire sans réponse
- Écrivez
- Avertir
- Indiquer
- Écritures signées authentifiées
- Propriétés étendues

4.3 Fonctionnement

Le BLE dispose de différents types de connexions. Un objet connecté en BLE peut avoir jusqu'à 4 fonctions différentes :

- Le « Broadcaster » : il peut faire office de serveur. Ainsi, il a pour objectif de transmettre régulièrement des données à un appareil, mais il n'accepte aucune connexion entrante.
- « L'Observer » : Dans un deuxième temps, l'objet peut seulement écouter et interpréter les données envoyées par un « broadcaster ». Dans cette situation-là, l'objet ne peut pas envoyer de connexions vers le serveur.
- Le « Central » : souvent un smartphone ou une tablette. C'est un élément qui interagit de deux façons différentes : soit en mode advertising, soit en mode connecté. Il est alors le dirigeant et c'est de lui que part l'échange de données.
- Le « Peripheral » : il accepte les connexions du central et lui envoie des données de manière périodique. Ce système a pour objectif de packager les données de façon universelle via le protocole afin qu'elles soient comprises par les autres périphériques.

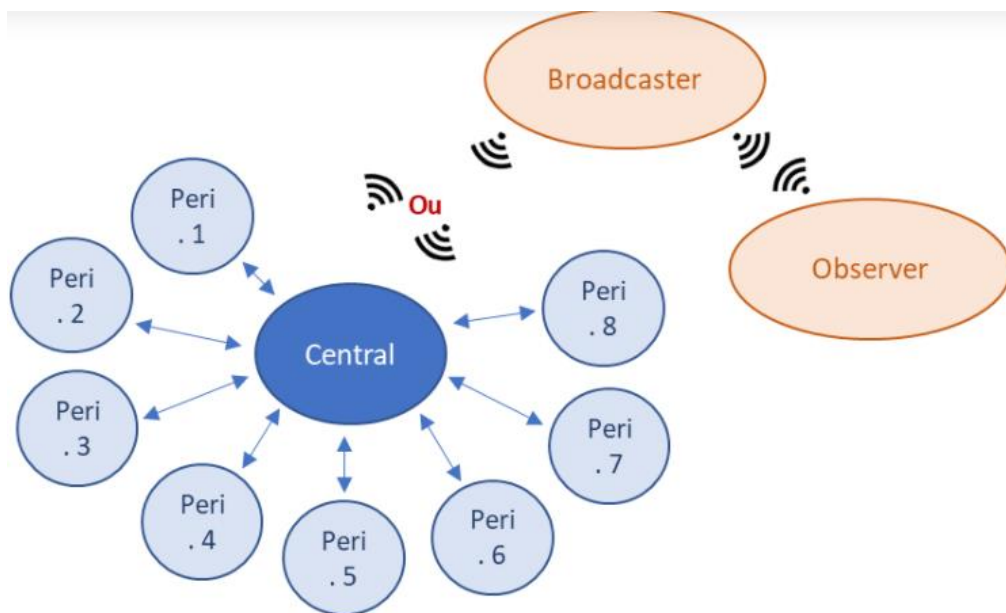


Figure 4.5: La disposition du BLE

4.4 Avantages du BLE

Chaque protocole a ses avantages et ses inconvénients. Voici quelques avantages du BLE :

- Pour l'utilisateur : En effet, il est issu d'un consortium de géant de l'industrie (IBM, Microsoft, Intel ...) ce qui lui confère une adoption et un fort crédit de stabilité économique. Il est déjà intégré dans tous les smartphones, ce qui facilite l'utilisation.
- Coté sécurité : le protocole permet de gérer plusieurs niveaux de sécurité.

4.5 Exemple

Dans cette application, nous allons connecter la carte ESP avec un smartphone.

Et pour réaliser ce travail, on a besoin de créer un UUID.

C'est un identifiant unique codé sur 128 bits, Dans sa représentation textuelle canonique, les 16 octets d'un UUID sont représentés comme 32 nombres hexadécimaux en minuscules séparés par 4 tirets sous la forme suivante:

Exemple : 123e4567-e89b-12d3-a456-426614174000

.....-...-M....-N....-.....

Les 4 bits de M et les bits de 1 à 3 de N déterminent le format de l'UUID

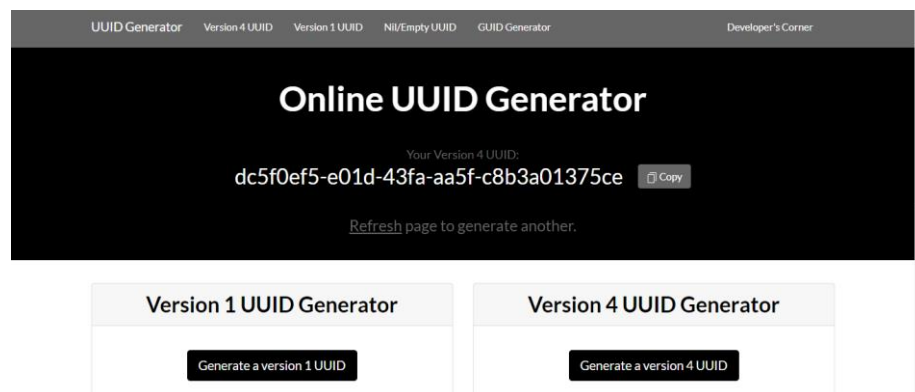


Figure 4.6 : Générateur de UUID

Le système de communication est Full-duplex car l'opération se fait dans les deux sens.

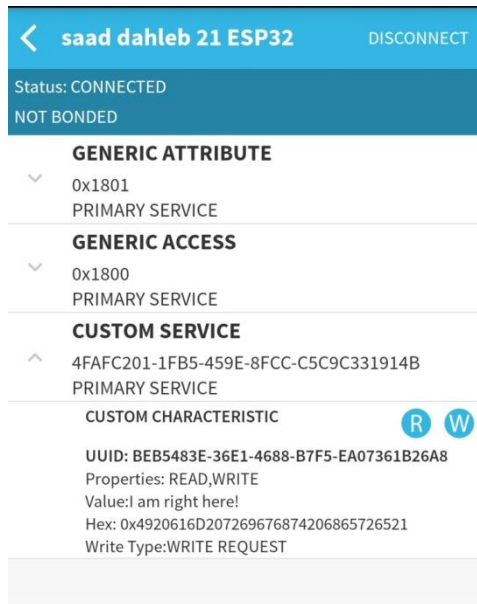


Figure 4.7: Connexion entre le téléphone et la carte

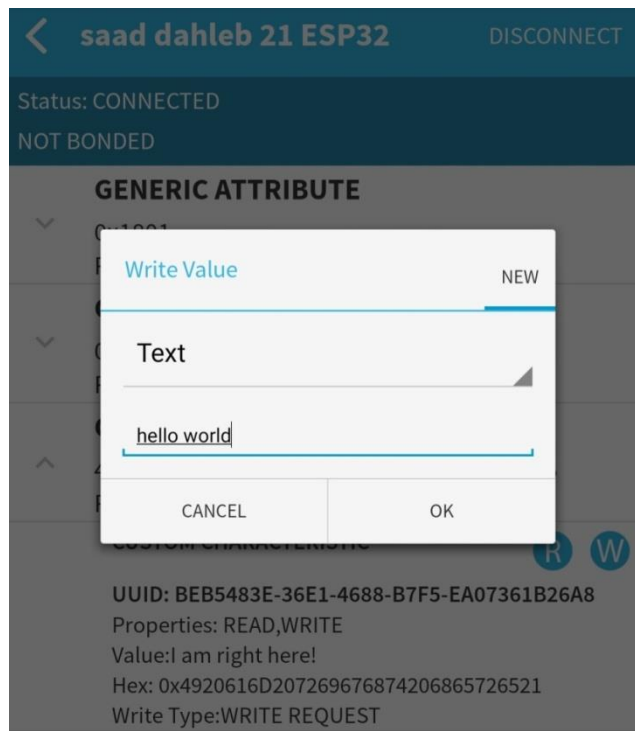


Figure 4.8: Connexion entre le téléphone et la carte

```
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
Starting BLE work!
Characteristic defined! Now you can read it in your phone!
*****
New value: hello world
*****
```

Figure 4.9: La réception du message

5 Communication bidirectionnelle ESP-NOW entre les cartes ESP32

5.1 Présentation d'ESP-NOW

ESP-NOW est un protocole de communication sans connexion développé par Espressif qui permet la transmission de paquets courts. Ce protocole permet à plusieurs appareils de communiquer entre eux sans utiliser le Wi-Fi.

Il s'agit d'un protocole de communication rapide qui peut être utilisé pour échanger de petits messages (jusqu'à 250 octets) entre les cartes ESP32. ESP-NOW est très polyvalent et vous pouvez avoir une communication unidirectionnelle ou bidirectionnelle dans différents arrangements.

Deux cartes ESP32 échangeront les lectures des capteurs (avec une portée en champ libre jusqu'à 220 mètres ~ 722 pieds).



Figure 4.10: ESP NOW

5.2 Obtenir l'adresse MAC de la carte

Pour envoyer des messages entre chaque carte, nous avons besoin de connaître leur adresse MAC . Chaque carte a une adresse MAC unique

Après avoir mis le code, appuyez sur le bouton RST/EN et l'adresse MAC devrait s'afficher sur le moniteur série.

The image shows two screenshots of the Arduino IDE serial monitor. The top screenshot shows a code editor with the following code:

```
#include "WiFi.h"

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_MODE_STA);
  Serial.println(WiFi.macAddress());
}

void loop(){}
}
```

The serial monitor output shows the following text:

```
ets Jun  8 2016 00:22:57
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:0:0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
7C:9E:BD:FA:00:F8
```

The bottom screenshot shows the same code editor with the following code:

```
#include "WiFi.h"

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_MODE_STA);
  Serial.println(WiFi.macAddress());
}

void loop(){}
}
```

The serial monitor output shows the following text:

```
ets Jun  8 2016 00:22:57
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:0:0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
08:3A:F2:A9:75:E4
```

Figure 4.11: Adresse MAC

5.3 Code ESP-NOW de communication bidirectionnelle ESP32

```
#include <esp_now.h>
#include <WiFi.h>

// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0x08, 0x3A, 0xF2, 0xA9, 0x75, 0xE4};

// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
  char a[32];
  int b;
  float c;
  String d;
} struct_message;

// Create a struct_message called myData
struct_message myData;
```

```
16:20:07.723 ->
16:20:07.723 -> Last Packet Send Status:      Delivery Success
16:20:09.691 -> Sent with success
16:20:09.724 ->
16:20:09.724 -> Last Packet Send Status:      Delivery Success
16:20:11.700 -> Sent with success
16:20:11.700 ->
16:20:11.700 -> Last Packet Send Status:      Delivery Success
16:20:13.703 -> Sent with success
16:20:13.703 ->
16:20:13.703 -> Last Packet Send Status:      Delivery Success
```

```
#include <esp_now.h>
#include <WiFi.h>

// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0x7C, 0x9E, 0xBD, 0xFA, 0x00, 0xF8};

// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
  char a[32];
  int b;
  float c;
  String d;
  bool e;
} struct_message;

// Create a struct_message called myData
struct_message myData;
```

```
15:20:18.156 -> Last Packet Send Status:      Delivery Success
15:20:20.166 -> Sent with success
15:20:20.166 ->
15:20:20.166 -> Last Packet Send Status:      Delivery Success
15:20:22.174 -> Sent with success
15:20:22.174 ->
15:20:22.174 -> Last Packet Send Status:      Delivery Success
15:20:24.184 -> Sent with success
15:20:24.184 ->
15:20:24.184 -> Last Packet Send Status:      Delivery Success
15:20:26.159 -> Sent with success
15:20:26.159 ->
15:20:26.159 -> Last Packet Send Status:      Delivery Success
```

Figure 4.12: La réception des paquets avec succès

6 Établissement d'une connexion WI-FI avec la carte esp32

6.1 Contrôle de LED via WEB

Dans la partie suivante nous allons montrer comment Contrôler une LED à partir d'un réseau local.

Tout d'abord on doit afficher l'adresse IP attribuée à la carte ESP dans le moniteur série après avoir mis le code dans IDE

```
17:47:18.898 -> Connecting to red1
17:47:19.512 -> .....
17:47:21.521 -> WiFi connected.
17:47:21.521 -> IP address:
17:47:21.521 -> 192.168.137.240
```

Figure 4.13: Affichage de l'adresse IP

Sur la barre de recherche du navigateur, on ajoute l'adresse attribuée 192.168.137.240

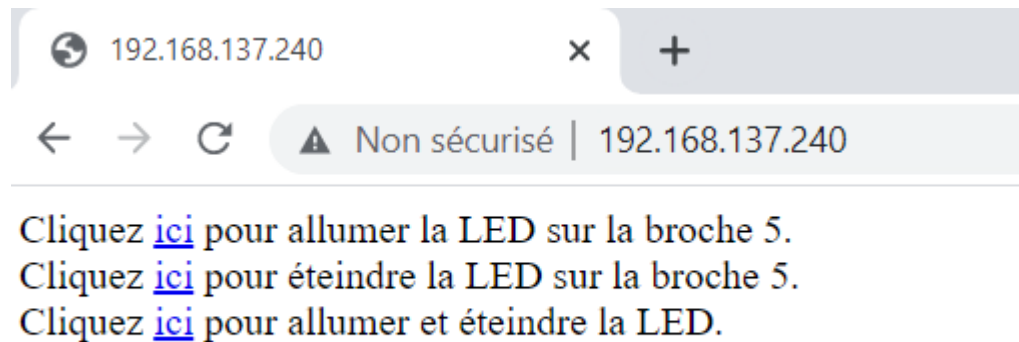


Figure 4.14: Page de contrôle

6.2 Mesure de qualité du signal (RSSI)

A propos du RSSI : Les mesures RSSI représentent la qualité relative d'un signal reçu sur un dispositif. RSSI indique le niveau de puissance reçu après une possible perte au niveau de l'antenne et du câble. Plus la valeur RSSI est élevée, plus le signal n'est fort. Lorsque mesuré avec des chiffres négatifs, si le chiffre est proche de zéro ceci signifie qu'il existe un meilleur signal. Par exemple : -50 est un assez bon signal, -75 est un signal plutôt raisonnable et -100 n'est pas un signal du tout.

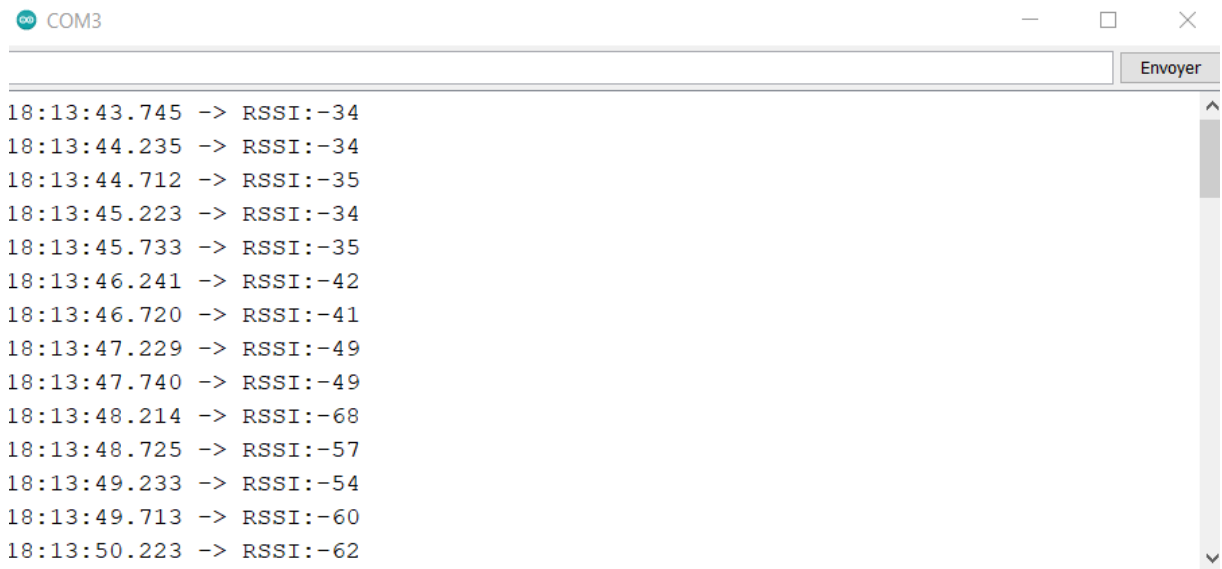
Même si RSSI et dBm sont des unités de mesure différentes, elles indiquent la force du signal. Le dBm est un taux de puissance de la puissance mesurée en tant que références à un mW (milliwatt). Alors que dBm est un indice absolu, le RSSI est un indice relatif.

Pour pouvoir mesuré le RSSI on est obligé de connecter notre carte ESP a un réseau WI-FI. Cette dernier est programmer pour indique le niveau du signal reçu

On doit d'abord insérer le nom du WIFI et le mot de passe dans le programme avec **const char* ssid = "nom du wifi » ;**

const char* password = "mtps du wifi"

On compile. Ensuite on ouvre le moniteur série dans la barres des outils, et on aura l'affichage suivant :



```
COM3
18:13:43.745 -> RSSI:-34
18:13:44.235 -> RSSI:-34
18:13:44.712 -> RSSI:-35
18:13:45.223 -> RSSI:-34
18:13:45.733 -> RSSI:-35
18:13:46.241 -> RSSI:-42
18:13:46.720 -> RSSI:-41
18:13:47.229 -> RSSI:-49
18:13:47.740 -> RSSI:-49
18:13:48.214 -> RSSI:-68
18:13:48.725 -> RSSI:-57
18:13:49.233 -> RSSI:-54
18:13:49.713 -> RSSI:-60
18:13:50.223 -> RSSI:-62
```

Figure 4.15: Mesure Du RSSI

7 Communication WI-FI client-serveur entre deux carte ESP

7.1 Introduction

Dans cette partie on va montrer comment configurer une communication HTTP entre deux cartes ESP32 pour échanger des données via Wi-Fi sans connexion Internet (routeur). En termes simples, vous apprendrez à envoyer des données d'une carte à l'autre à l'aide de requêtes HTTP. Les cartes ESP32 seront programmées à l'aide d'Arduino IDE.

Une carte ESP32 agira en tant que serveur et l'autre carte ESP32 agira en tant que client. Le diagramme suivant montre un aperçu de la façon dont tout fonctionne.

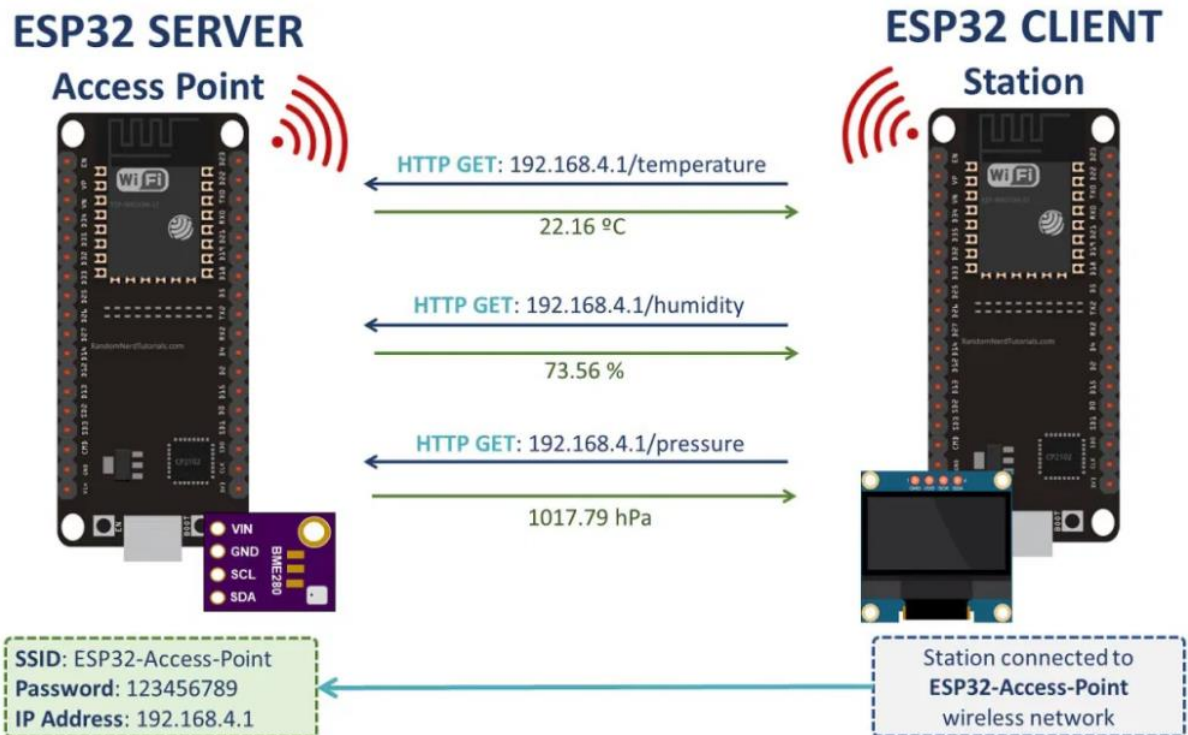


Figure 4.16: Client-serveur ESP32

7.2 Les étapes du travail

- Le serveur ESP32 crée son propre réseau sans fil. Ainsi, d'autres appareils Wi-Fi peuvent se connecter à ce réseau.
- Le client ESP32 est défini comme une station. Ainsi, il peut se connecter au réseau sans fil du serveur ESP32.
- Le client peut faire des requêtes HTTP GET au serveur pour demander des données de capteur ou toute autre information. Il suffit d'utiliser l'adresse IP du serveur pour faire une requête sur une certaine route :/Température, /humidité ou /pression.
- Le serveur écoute les demandes entrantes et envoie une réponse appropriée avec les lectures.
- Le client reçoit les lectures et les affiche sur l'écran OLED.

7.3 Serveur ESP32

Le serveur ESP32 est un point d'accès (AP), qui écoute les requêtes sur le/Température, /humidité et /pression URL. Lorsqu'il reçoit des requêtes sur ces URL, il envoie les dernières lectures du capteur BME280.

7.3.1 Diagramme schématique

BME280	ESP32
VIN	3.3V
GND	GND
SCL	GPIO 22
SDA	GPIO 21

Tableau 6 : Table de branchement (BME280 /ESP32 server)



On compile le code et on aura l'affichage suivant dans le moniteur série :

```
COM3
18:27:51.134 -> rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
18:27:51.168 -> configsip: 0, SPIWP:0xee
18:27:51.168 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_dr
18:27:51.168 -> mode:DIO, clock div:1
18:27:51.168 -> load:0x3fff0018,len:4
18:27:51.168 -> load:0x3fff001c,len:1216
18:27:51.168 -> ho 0 tail 12 room 4
18:27:51.168 -> load:0x40078000,len:10944
18:27:51.168 -> load:0x40080400,len:6388
18:27:51.168 -> entry 0x400806b4
18:27:51.436 ->
18:27:51.436 -> Setting AP (Access Point)...AP IP address: 192.168.4.1
```

Figure 4.17: La compilation du code

Maintenant, pour nous assurer qu'il est à l'écoute des demandes de température, d'humidité et de pression, nous devons nous connecter à son réseau.

Dans notre smartphone, allez dans les paramètres Wi-Fi et connectez-vous au ESP32-Access-Point. Le mot de passe est 123456789.

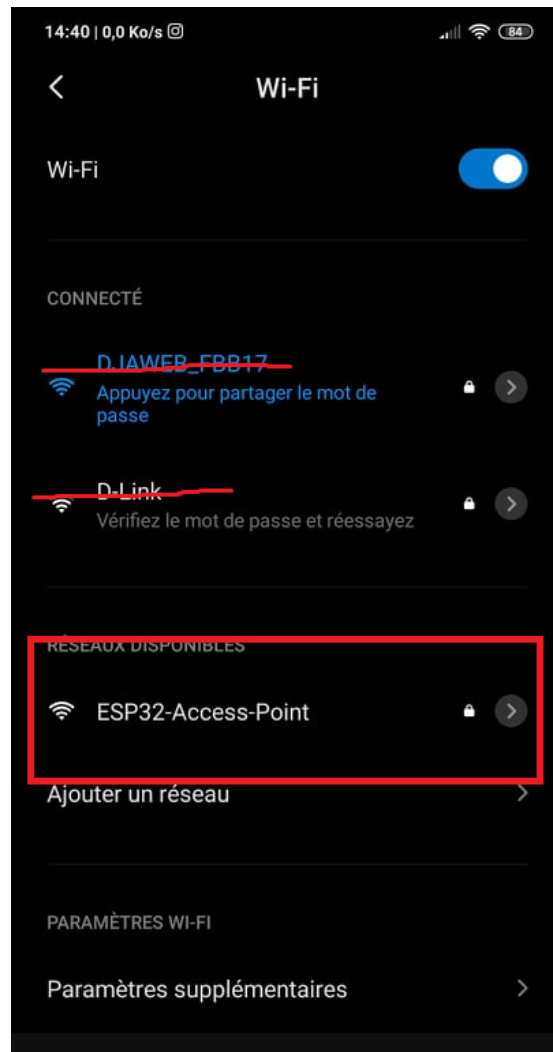


Figure 4.18: Connexion via Smartphone

Une fois connecté au point d'accès, ouvrez votre navigateur et tapez 192.168.4.1/température :

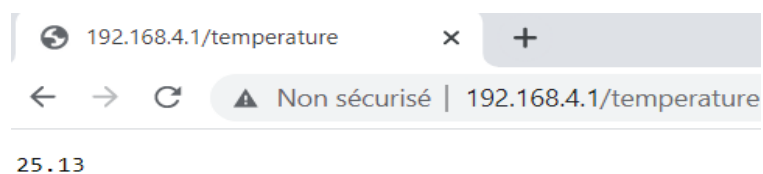


Figure 4.19: Visualisation de la température

Essayez ce chemin d'URL pour l'humidité 192.168.4.1/humidité :

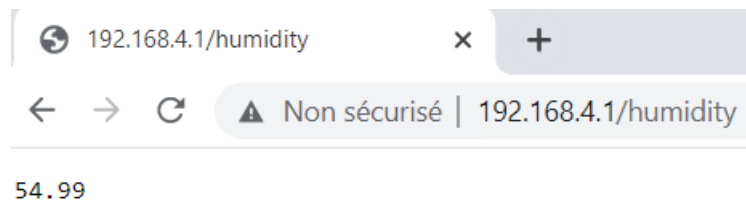


Figure 4.20: Visualisation de l'humidité

Enfin, allez à 192.168.4.1/pression URL :

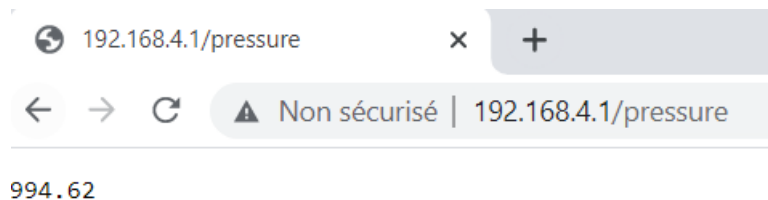


Figure 4.21: Visualisation de la pression

7.4 Client ESP32 (Station)

Le client ESP32 est une station Wi-Fi qui est connectée au serveur ESP32. Le client demande la température, l'humidité et la pression au serveur en faisant des requêtes HTTP GET sur le/Température, /humidité, et /pression Itinéraires d'URL. Ensuite, il affiche les lectures sur un écran OLED.

OLED	ESP32
NIV/CCV	3.3V
GND	GND
SCL	GPIO 22
SDA	GPIO 21

Tableau 7 : table de branchement (ESP client / LCD)

On entre le code dans l'IDE et on compile le résultat s'affiche comme suivant :

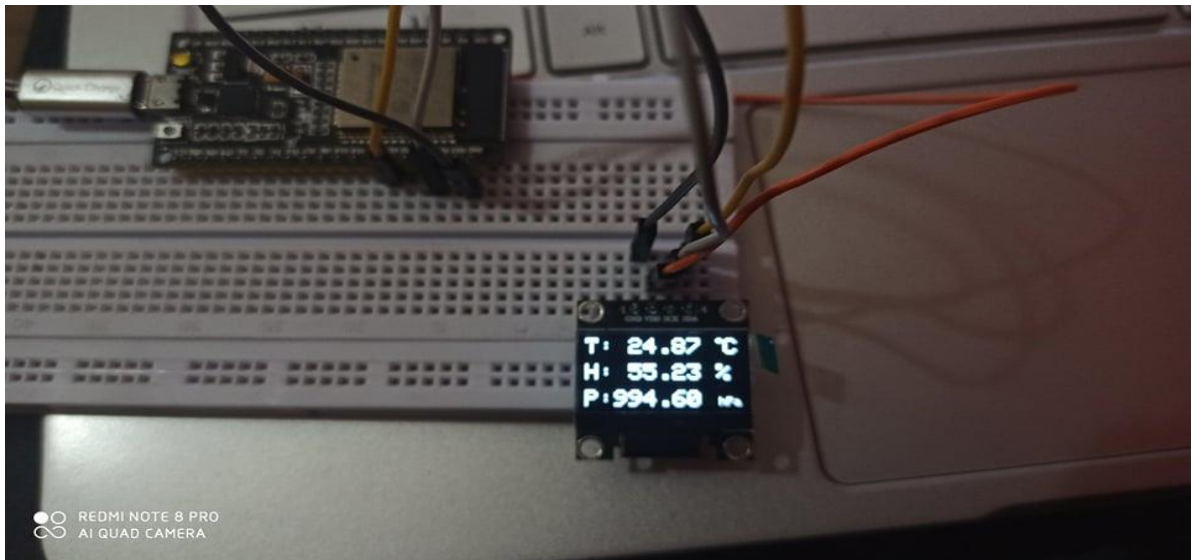


Figure 4.22: Visualisation de la pression, la température et l'humidité via LCD

7.5 Conclusion

Dans ce travail, nous avons appris à envoyer des données d'un ESP32 à une autre carte ESP32 via Wi-Fi à l'aide de requêtes HTTP sans avoir recours à Internet. À des fins de démonstration, nous avons montré comment envoyer les lectures du capteur BME280, mais on peut utiliser n'importe quel autre capteur pour envoyer d'autres données

Chapitre 5 :

Applications Cayenne

1 Introduction

MyDevices a développé une plateforme IDO robuste permettant aux sociétés de connecter des objets, de visualiser des données, d'appliquer des règles sophistiquées et d'interagir avec leurs clients connectés de manière efficace. Avec Cayenne, MyDevices étend sa plateforme agnostique de connectivité et d'objets connectés (lancée en octobre 2015) à la communauté des développeurs et des créateurs, sous la forme d'un outil facile d'utilisation grâce à la fonction « Glisser-déposer »

2 Historique

Lancée en début d'année par MyDevices, la division Internet des objets (IdO) du groupe français d'édition de logiciels Avanquest, la plate-forme de création et de développement de projets IdO Cayenne existe désormais dans une version LoRa, du nom de la technologie de communication radio longue portée et basse consommation pour réseaux M2M et IdO de Semtech. L'outil Cayenne, qui aurait été adopté par près de 100 000 développeurs, est présenté comme un moyen simple pour les entreprises de connecter des objets (dès lors qu'ils sont répertoriés dans l'outil), de visualiser des données, d'appliquer des règles sophistiquées et d'interagir avec leurs clients connectés de manière efficace. A ce titre, si l'on en croit myDevices, la reconnaissance automatique des objets et des capteurs lui permettrait d'être opérationnelle en quelques minutes, des widgets de type glissé-déposer facilitant la visualisation des données en temps réel et historiques en ligne ou à partir d'une application mobile. La plate-forme dispose aussi de déclencheurs, permettant à un capteur, un moteur ou un actionneur de lancer une action sur un autre objet, et d'alertes de seuils qui envoient des e-mails ou des SMS selon des critères personnalisables. Et ce avec la possibilité de programmer la mise en route et l'arrêt de lumières, moteurs, soupapes et actionneurs à des jours et à des heures données. Avec Cayenne for LoRa, les développeurs et les entreprises qui veulent bâtir une solution LoRa peuvent désormais utiliser gratuitement l'outil pour prototyper, développer et déployer rapidement une application pour l'Internet des objets, indique MyDevices, qui est par ailleurs membre de l'alliance LoRa. Parallèlement, n'importe quel constructeur d'équipements LoRa peut ajouter la plate-forme Cayenne à ses produits et la mettre à disposition de ses clients. Selon MyDevices, plusieurs fabricants intègrent déjà leurs capteurs, modules ou passerelles compatibles avec le protocole LoRaWAN au sein de l'environnement Cayenne, comme les taiwanais AcSiP et GlobalSa

3 À propos de MyDevices

MyDevices développe des plateformes middleware et des applications pour l'Internet des objets afin de « simplifier le monde connecté ». Les solutions personnalisées de MyDevices permettent aux sociétés qui fabriquent, vendent et prennent en charge des objets connectés de connecter les produits, de gérer les données et d'interagir avec leurs clients efficacement. MyDevices est la première plateforme du genre à proposer une solution d'objets connecté backend pour les entreprises et une solution frontend interconnectée pour l'utilisateur final. Basée à Los Angeles, MyDevices est une division du groupe Avanquest.

3.1 Le rôle de la plateforme Cayenne

La solution Cayenne dispose de capacités révolutionnaires comme :

- La reconnaissance automatique des objets et capteurs lui permettant d'être opérationnelle en 7 minutes
- Des widgets *Glisser-déposer* pour visualiser facilement les données en temps réel et historiques en ligne ou à partir d'une application mobile
- Des déclencheurs permettant à un capteur, un moteur ou un actionneur de lancer une action sur un autre objet
- Des alertes de seuils qui envoient des e-mails ou des SMS selon des critères personnalisables
- La possibilité de programmer la mise en route et l'arrêt de lumières, moteurs, soupapes et actionneurs à des jours et à des heures données

D'un point de vue commercial, Cayenne peut être utilisé pour effectuer des prototypages efficaces, optimiser les prévisions commerciales ou encore assurer la continuité du flux de travail. À la maison, Cayenne permet par exemple de contrôler, de manière simple et fluide, la domotique, la sécurité, les capteurs et l'éclairage. Outre une interface utilisateur intégrée et intuitive, son plus gros avantage est de permettre d'entreprendre des projets IdO simples et de réduire de manière significative le temps et les frais associés à des projets IdO plus complexes

3.2 Les Principales caractéristiques

- Une application mobile pour configurer, suivre et contrôler les objets et les capteurs partout où l'on se trouve
- Une configuration intuitive qui connecte Pi à Internet, aux capteurs, actionneurs et extensions en quelques minutes seulement

- Un moteur de règles pour déclencher des actions sur les objets
- Un tableau de bord personnalisable avec des widgets de visualisation à Glisser-déposer
- Une programmation des lumières, moteurs et actionneurs
- Une commande GPIO configurable depuis une application mobile ou un tableau de bord
- Un accès distant instantané depuis un téléphone ou un ordinateur

3.3 Fonctionnement de Cayenne

Cayenne est la première plate-forme de son genre, constructeur de projet IdO drag and drop qui permet aux développeurs de créer et héberger rapidement leurs projets d'objets connectés. Cayenne a été conçu pour l'Internet des objets. Il peut contrôler le matériel à distance, il peut afficher des données de capteurs, il peut stocker des données, analyser et faire beaucoup d'autres choses cool.

Il y a plusieurs éléments importants dans la plate-forme :

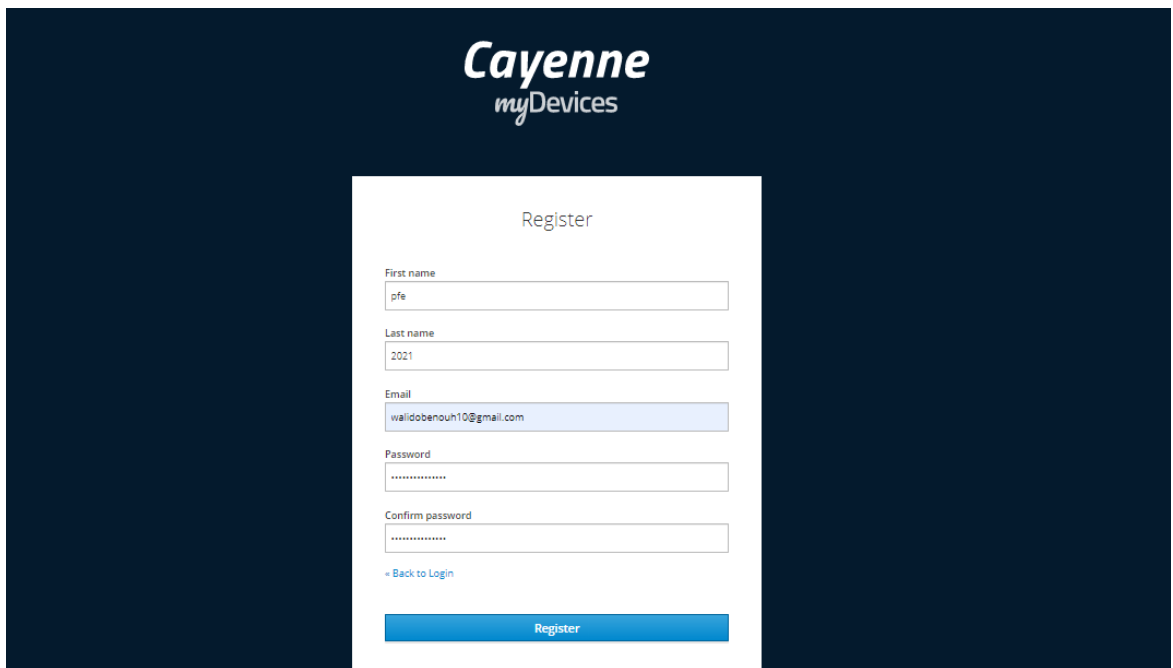
- ❖ **App Cayenne** - configurer et contrôler vos projets IdO avec des widgets drag and drop à partir d'une application mobile.
- ❖ **Portail en ligne Cayenne** - Utilisez un navigateur pour configurer et contrôler vos projets IdO.
- ❖ **Cayenne Cloud**- En charge du traitement et du stockage des données de l'objet, et également en charge de la gestion des commandes utilisateurs, des actions, des déclencheurs et des alertes sur les capteurs.
- ❖ **Agent Cayenne** - permet la communication entre le serveur et le matériel pour la mise en œuvre des commandes entrantes et sortantes, les actions, les déclencheurs et les alertes.

Chaque fois que vous appuyez sur un bouton de l'application Cayenne ou du portail en ligne, l'action est traitée à travers le Cloud de Cayenne et trouve son chemin au matériel correspondant. Cela fonctionne également dans le sens opposé. Vous pouvez utiliser l'application mobile de Cayenne ou le portail en ligne, c'est à vous de jouer. Toutes les modifications apportées au matériel via l'application mobile sont reflétées lors de l'affichage du portail en ligne et vice versa.

4 Création d'un compte

D'abord on doit s'inscrire à un compte gratuit.

Pour commencer à créer le compte, on doit se rendre a la page d'inscription Cayenne. Sur cette page d'inscription, on va entrer les informations suivantes : nom, email et un mot de passe



The image shows the registration page for Cayenne myDevices. The page has a dark blue background with the Cayenne myDevices logo at the top. The registration form is centered and contains the following fields:

- First name: pfe
- Last name: 2021
- Email: walidobenouh10@gmail.com
- Password: [masked]
- Confirm password: [masked]

There is a link for "Back to Login" and a blue "Register" button at the bottom of the form.

Figure 5.1: Page d'inscription à Cayenne

Une fois le compte a été créé, on doit choisir la carte souhaitée.

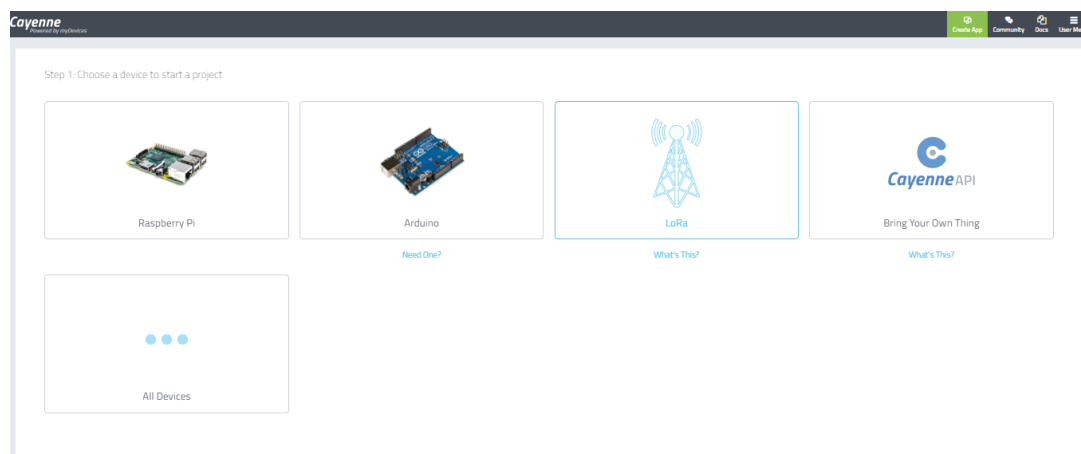


Figure 5.2: Page d'accueils Cayenne

- ❖ Comme vous voyez, l'ESP32 n'est pas trouvable sur la page. On va l'ajouter via [Cayenne API]

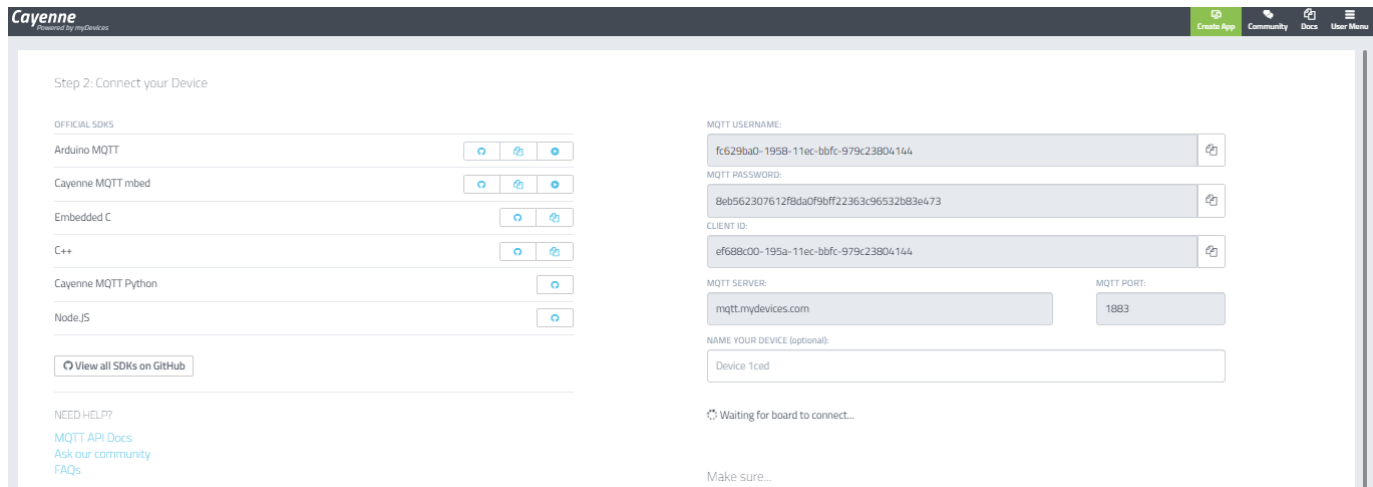


Figure 5.3: Page des données de la carte

- ❖ La plateforme nous a donné des identifiants pour définir notre carte ESP32 :

- MQTT USERNAME :
- MQTT PASSWORD :
- CLIENT ID :
- MQTT SERVER :
- MQTT PORT :
- NAME YOUR DEVICE (optionnel) : nommé la carte.

- ❖ On doit choisir le SDKS (Arduino MQTT) ensuite, on doit télécharger les bibliothèques (Cayenne Arduino MQTT Library).

Une fois les bibliothèques sont téléchargées, on doit les ajouter dans l'IDE (Outils → Gérer les bibliothèques → Cayenne MQTT dans la barre de recherche)

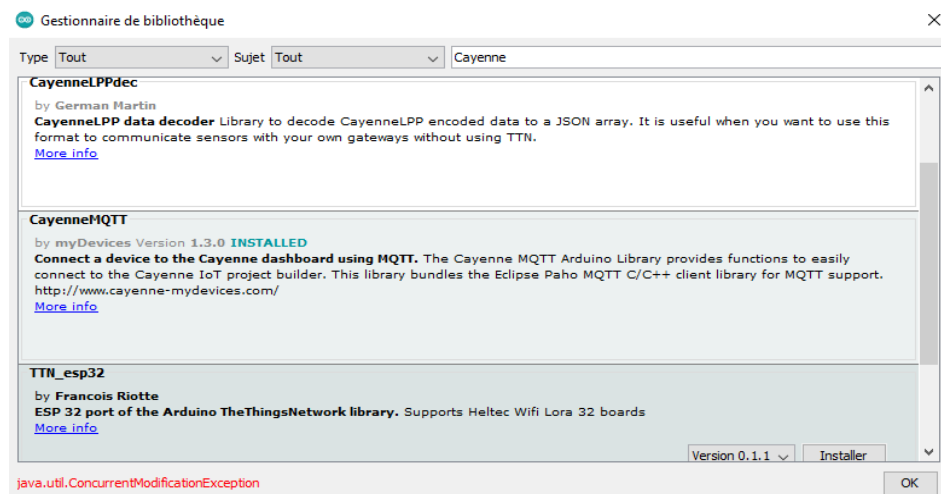


Figure 5.4: Installation des bibliothèques

- ❖ Une fois les étapes précédentes sont réalisées, on relie la carte avec le wifi et notre compte Cayenne.

```
// WiFi network info.  
char ssid[] = "walid";  
char wifiPassword[] = "walidol23";  
  
// Cayenne authentication info. This should be obtained from the Cayenne Dashboard.  
char username[] = "e3fc2aa0-d4e7-11eb-8779-7d56e82df461";  
char password[] = "dd196cb33383aec6fcf98a1e311857e28c69213e";  
char clientID[] = "f3c1a460-d4e7-11eb-883c-638d8ce4c23d";
```

Figure 5.5: Intégration des données dans le code IDE

- ❖ Après avoir importé le code dans la carte,

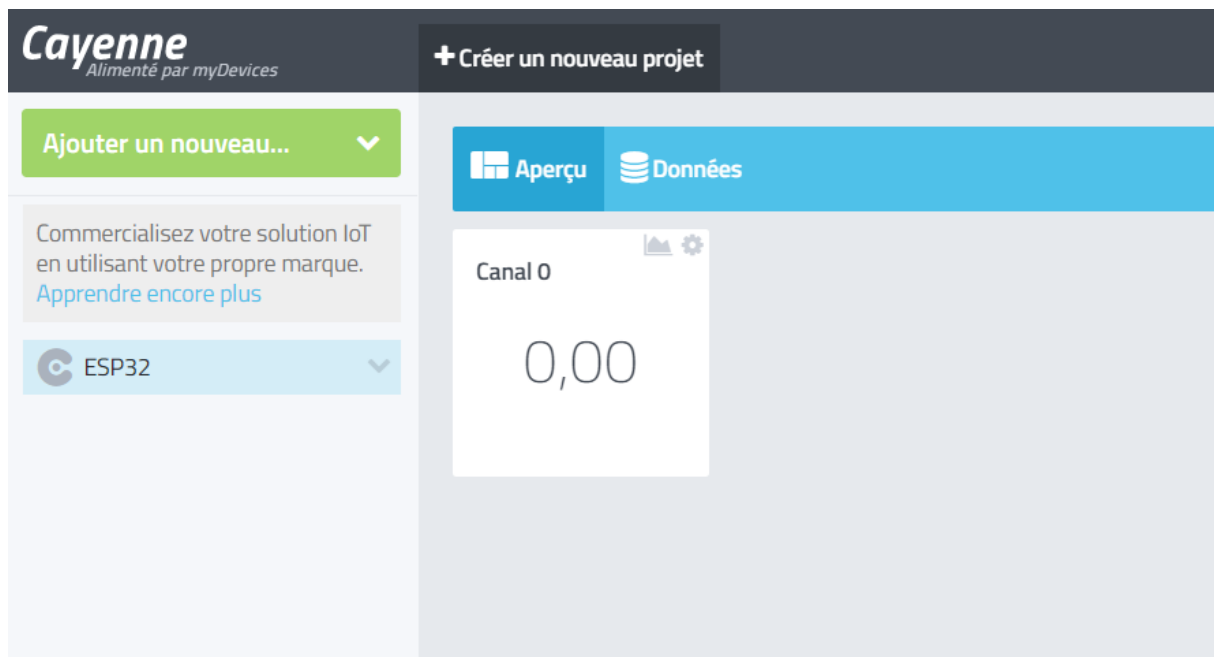


Figure 5.6: Tableau de bord Cayenne par défauts

5 Création d'un appareil Widget

Dans cette partie, on va créer des Widgets qui vont nous permettre de visualiser, contrôler notre carte à travers Cayenne MQTT.

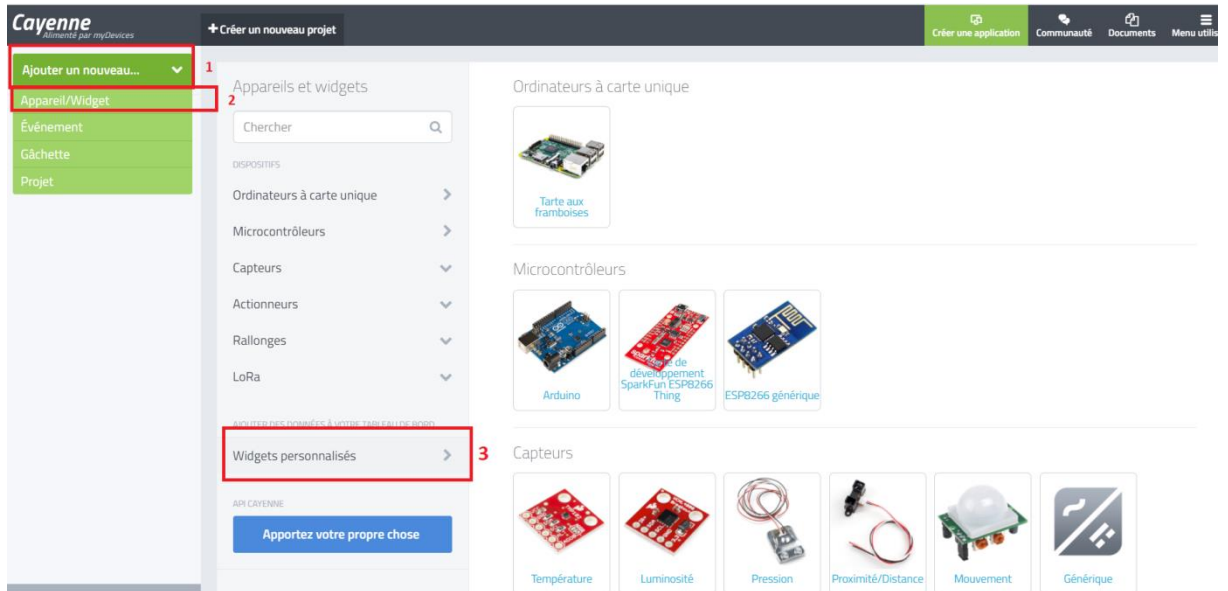


Figure 5.7: Création d'un nouveau Widget

Cette communication (Carte x plateforme) se fait par internet a travers des paquets MQTT.

5.1 Touche

Dans cette application, on va s'intéresser au port (PIN15), ce dernier va produire un signal qui va être transmis de la carte vers la plateforme.

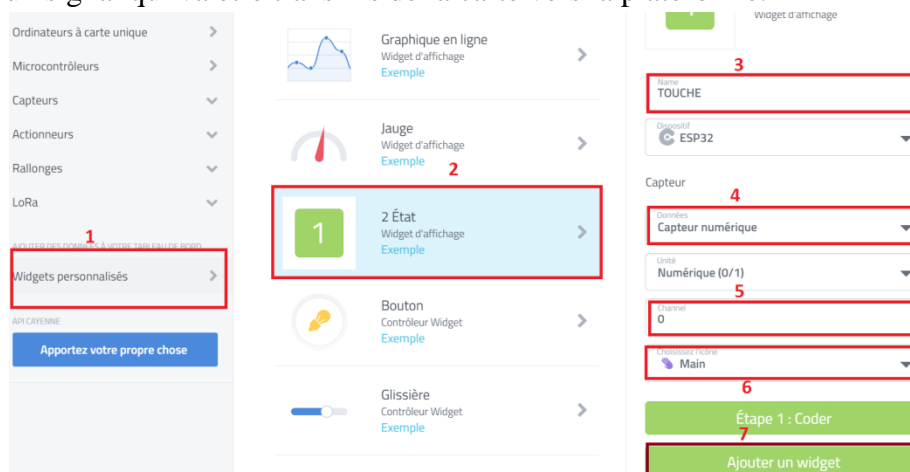


Figure 5.8: Paramètres du widget touche

- Etape 1 : Création d'un widget personnalisée
- Etape 2 : Type de widget
- Etape 3 : On nomme de Widget
- Etape 4 : Type de capteur (capteur numérique)
- Etape 5 : Affectation de notre widget a une chaine (0)
- Etape 6 : le choix d'icône
- Etape 7 : Finalisation

❖ Explications : Quand on touche le port 15, on va remarquer que le Widget va être allumé, ça veut dire que la communication entre le port et notre widget est faite.

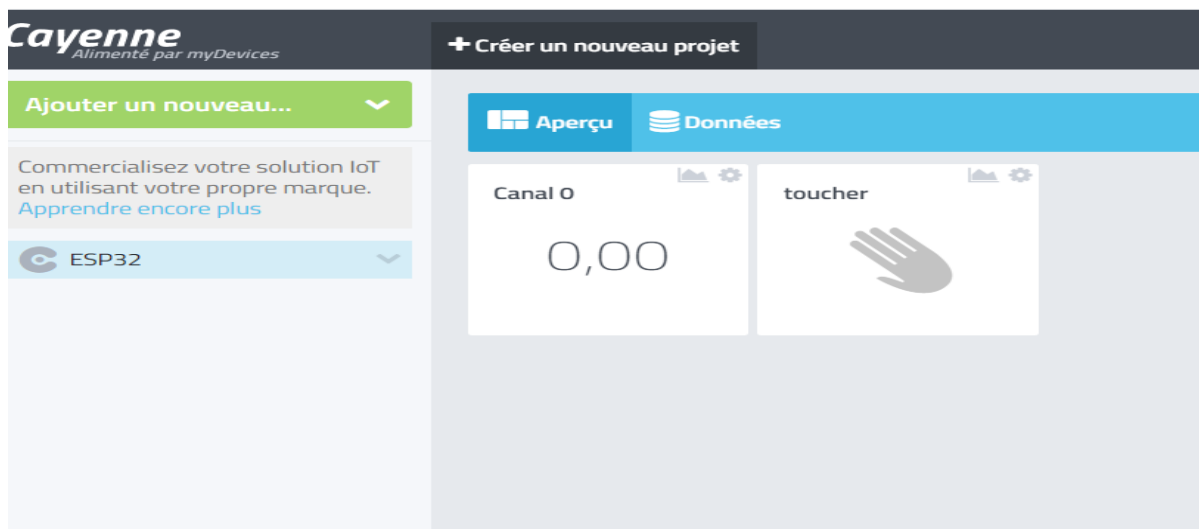


Figure 5.9: L'icône du widget touche sans toucher le port dédié

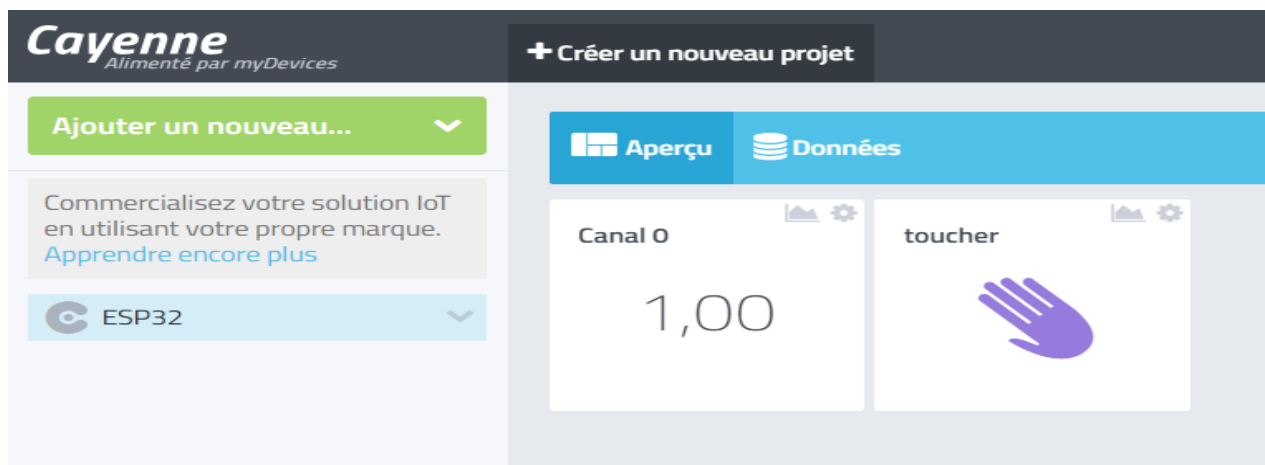


Figure5.10: L'icône du widget quand on touche le port

5.2 LED :

Dans cette application on va contrôler la LED qui est relié avec le port 5 depuis Cayenne MQTT.

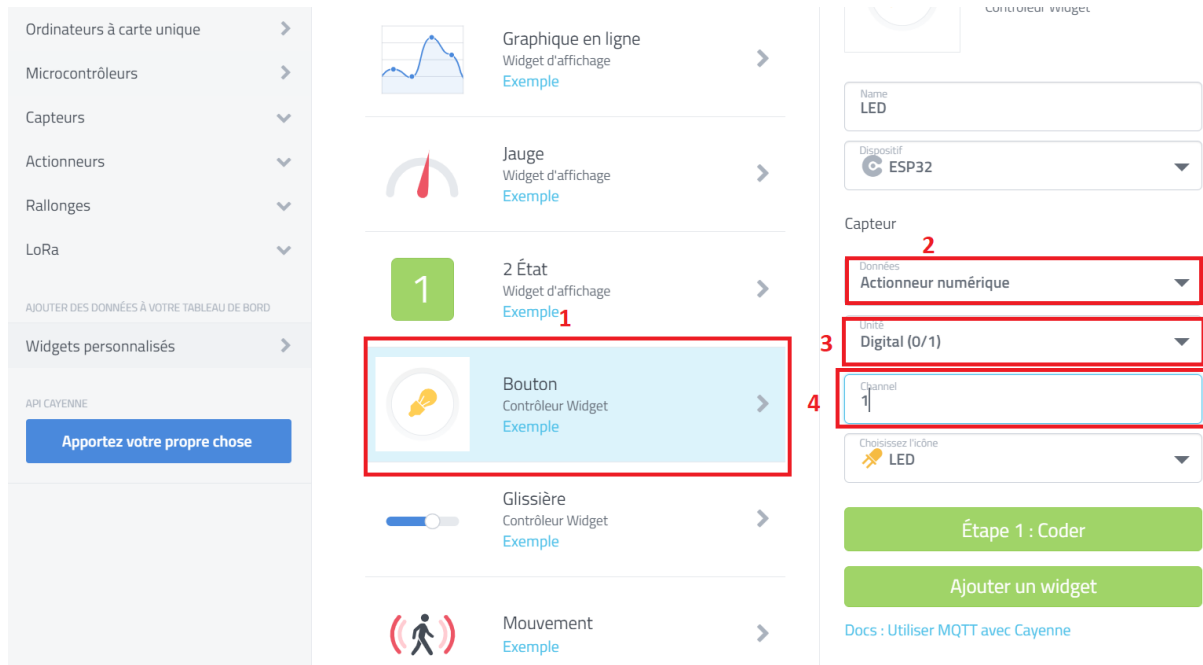


Figure 5.11: Paramètres du widget LED

❖ Pour cette application, la création de ce widget reste la même avec quelque changement à faire :

- Etape 1 : Type de Widget (bouton)
- Etape 2 : Type de capteur (actionneur numérique)
- Etape 3 : Unité (Digital 0/1)
- Etape 4 : affectation a une chaine (1)
- Etape 6 : le choix d'icône
- Etape 7 : Finalisation

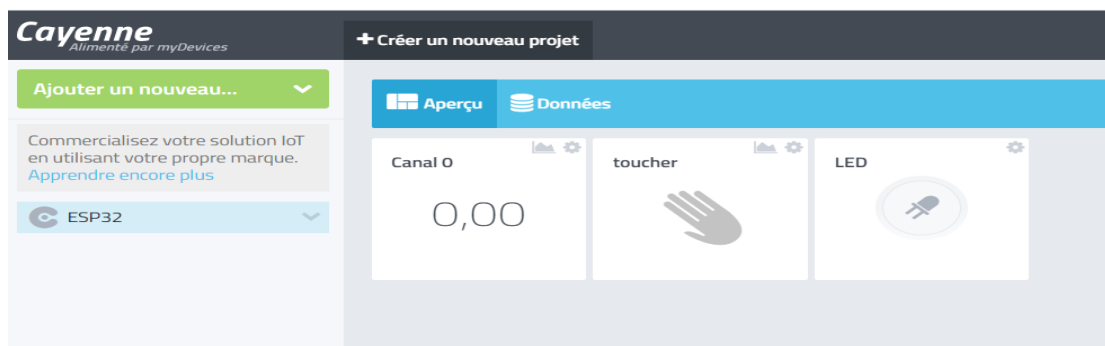


Figure 5.12: Tableau de bord Cayenne après avoir ajouté les Widgets

- De là, on peut allumer ou éteindre la LED avec un simple clic sur l'icône LED.

5.3 L'utilisation d'une boucle dans la plateforme Cayenne MQTT

Une boucle est une structure qui permet de répéter les mêmes instructions plusieurs fois.

Dans cette application, on va essayer de relier les deux applications précédentes à l'aide d'une boucle. Ce travail va nous permettre de contrôler et visualiser en même temps la touche et La LED.

5.3.1 Création d'une boucle

- La création se fait sur la plateforme.



Figure 5.13: Ajouter un gâchette boucle

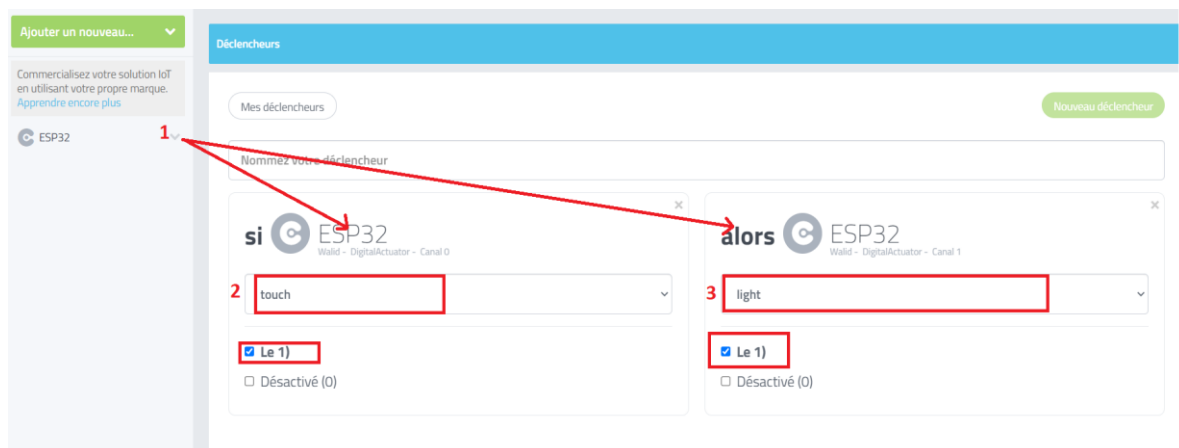


Figure 5.14: Paramètres de la gâchette boucle

- Pour commencer, on doit Glisser l'icône ESP32 vers les deux conditions [Si Alors].

En suite choisir la condition suivante :

- Si la touche est égale 1 Alors la LED est égale : 1.

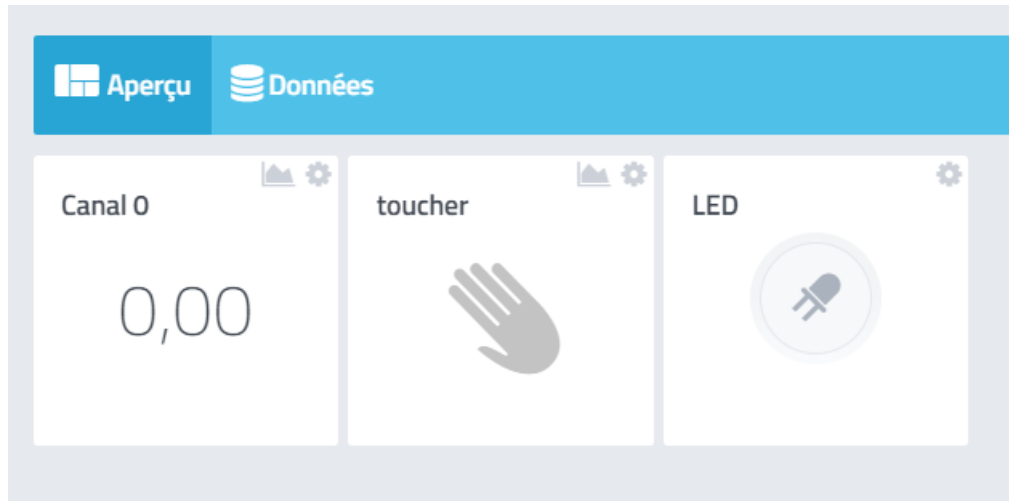


Figure 5.15: Les Widgets sans touché le port

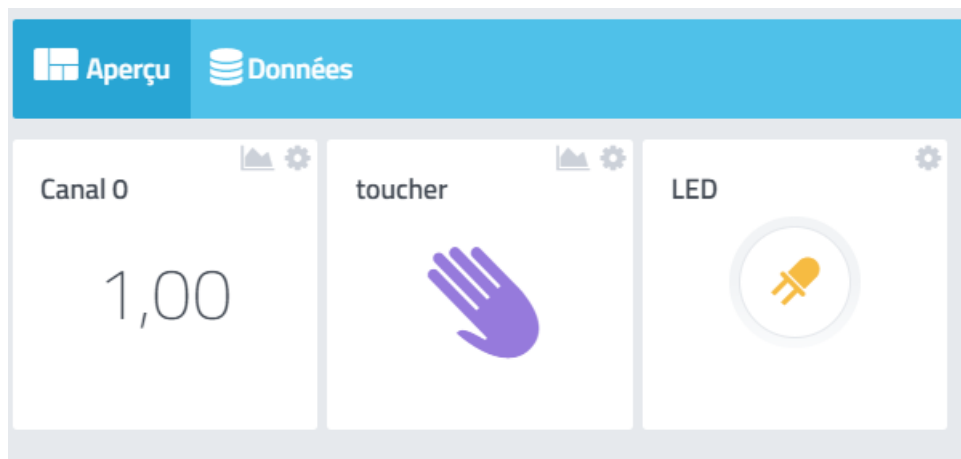


Figure 5.16: Les Widgets après avoir touché Le port

5.3.2 L'utilisation d'un déclencheur dans la plateforme Cayenne MQTT :

Les déclencheurs permettent d'inclure des expressions dans vos flux. Vous pouvez créer des déclencheurs qui sont basés sur le temps ou qui sont liés à un événement qui se produit sur un appareil.

- Dans cette application, on va essayer de relier l'application touche à un déclencheur. Ce travail va nous permettre de visualiser la touche à travers notre e-mail.
- La création se fait sur la plateforme a travers la cachette (figure 5.13)

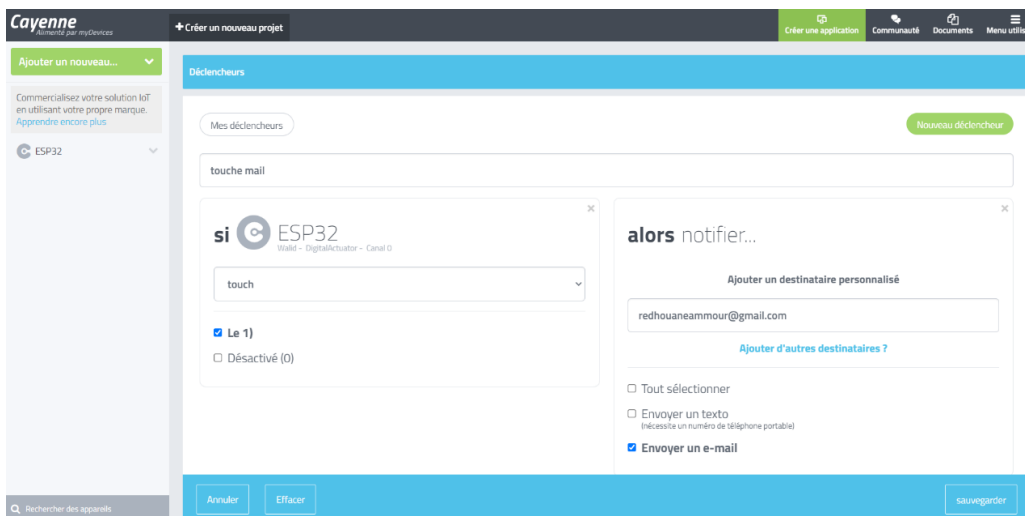


Figure 5.17: Paramètres du déclencheur

- ❖ Pour commencer, on doit Glisser l'icône ESP32 vers La condition Si.
- ❖ En suite choisir la condition suivante :
 - Si : la touche est égale à 1.
 - Alors : configure notification ensuite envoyer un e-mail on déclare notre e-mail et on sauvegarde.
 - Si on touche le port 15 on aura ce mail de la part de Cayenne :

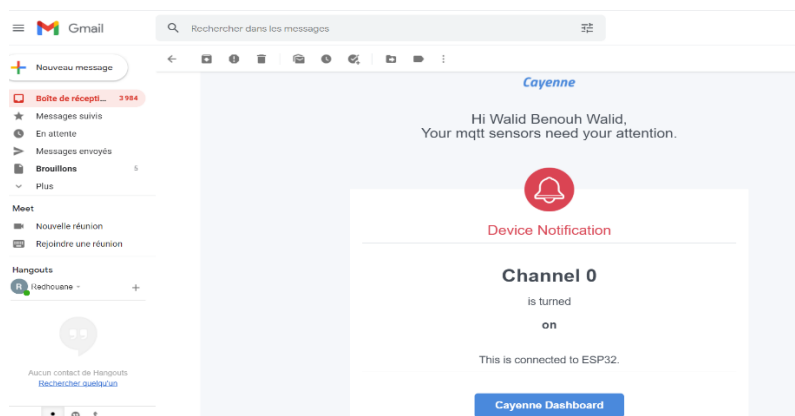
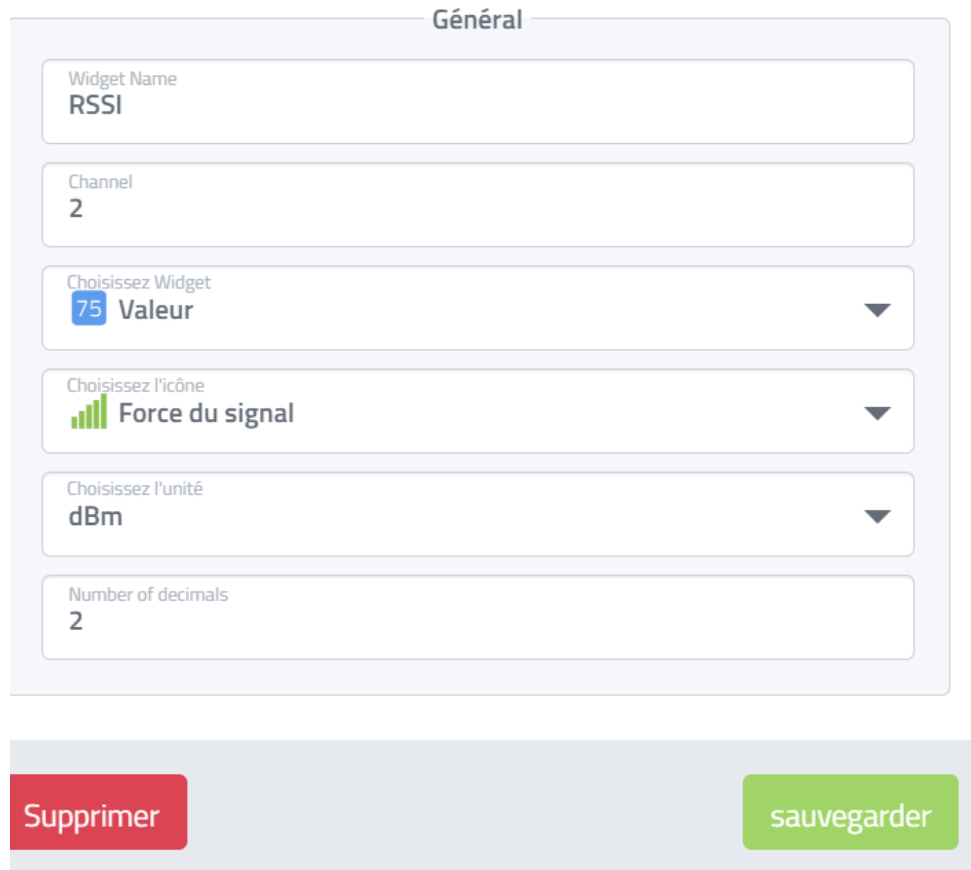


Figure 5.18: l'email reçu de la part de Cayenne

5.4 L'application RSSI :

Dans cette application, on va mesurer la puissance de signal reçu en dB.



Général

Widget Name
RSSI

Channel
2

Choisissez Widget
75 Valeur ▼

Choisissez l'icône
Force du signal ▼

Choisissez l'unité
dBm ▼

Number of decimals
2

Supprimer sauvegarder

Figure 5.19: Paramètres du widget RSSI Valeur

- ❖ Dans la figure précédente, on va remplir les paramètres suivants :
 - Le nom du Widget
 - Numéro de chaîne
 - Choix d'un widget (valeur)
 - Choix de l'icône (Force du signal)
 - L'unité (dBm)
 - Numéro de décimale (2)

Et on peut aussi régler les paramètres d'une autre façon pour obtenir un affichage (Jauge) par exemple :

Général

Widget Name
RSSI

Channel
2

Choisissez Widget
📊 **Jauge** ▼

Choisissez l'unité
dBm ▼

Number of decimals
2

Paramètres de la jauge

Veuillez saisir une plage qui ne chevauche pas une autre plage.

Plage 1	Start -70	End -100	Couleur ■ rouge ▼	
Gamme	Start -50	End -69	Couleur ■ Jaune ▼	✕
2				
Gamme	Start -7	End -49	Couleur ■ Vert ▼	✕
3				

+ [Ajouter une plage](#)

Figure 5.20: Paramètres du widget RSSI jauge

Dans la deuxième, on a choisi un autre type de Widget (Jauge).

Le réglage des paramètres du débit peut se faire de n'importe quelle manière telle que la personne doit respecter les plages.

La figure suivante est notre tableau de bord après avoir configuré les applications RSSI.

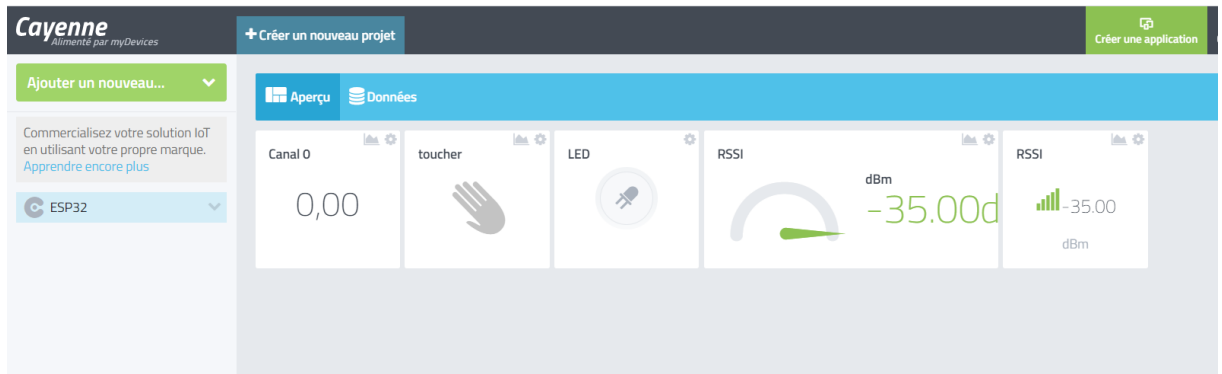


Figure 5.21: Tableau de bord final

6 Conclusion :

Cayenne se révèle un outil très puissant pour créer rapidement un tableau de bord, son utilisation simplifiera grandement la vie.

Le service Cayenne permet aussi d'afficher les courbes ou les événements sur une période, à des fins de surveillance ou d'analyse. Ce qu'on a fait dans notre travail, en gros Cayenne nous a permis de contrôler les cartes.

Conclusion Générale :

Au cours de ce mémoire, nous avons présenté au début une vue générale sur l'état de l'art de l'internet des objets, nous avons présenté par la suite les composants ainsi les protocoles de l'internet des objets et par la fin les différentes plateformes dédiées aux applications de l'IdO.

La particularité de ce travail est la réalisation physique de plusieurs systèmes de l'IdO qui présente la première expérience pour nous. Plusieurs concepts sont acquis, qui concernent non seulement la partie software mais aussi la partie hardware dont on a réalisé un système d'IoT pour gérer les dispositifs.

Les solutions proposées comportent 3 parties :

- Matériels
 - Plateforme
 - Protocole de communication
-
- ❖ Ce qui concerne la partie matérielle : On a réalisé des applications pour l'utilisation quotidienne de l'homme
 - ❖ Ce qui concerne la partie plateforme : On a connecté les cartes ESP32 avec Cayenne plateforme IdO grâce au protocole MQTT
 - ❖ Ce qui concerne les protocoles de communications : On a utilisé les protocoles suivants :
 - Wifi : pour la communication entre les deux cartes
 - MQTT : pour la communication entre le serveur et la carte
 - BLUETOOTH : pour la communication entre deux cartes.

BIBLIOGRAPHIE

- [1] Cécile Nadai. Focus on the internet of things, Aout 2017.
- [2] Luigi Atzori. The Internet of Things: A survey , Mai 2010.
- [3] Dahmane Izihome. Les objets connectés, Avril 2016.
- [4] Maxime Dechany. International Télécommunication Union, Mai 2014.
- [5] Lida Xu, Wu He, Internet of Things in Industries: A Survey, Novembre 2014.
- [6] Jena Satyabrata, Architecture of Internet of Things, Juin 2020
- [7] Farid Hiddoussi, Réalisation d'une palteforme IoT, Juillet 2019
- [8] *Open Oasis. OASIS Message Queuing Telemetry Transport (MQTT)*.
'https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt' Aout 2021
- [9] Carey Scott, Computer world duk, Best cloud Iot plateformes, Fevrier 2019
- [10] DAVE, Evans. L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?. Avril 2011

