

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Oran 1, Ahmed Ben Bella
Institut des Sciences et Techniques Appliquées

Thème

Conception et implémentation d'un système de suivi d'objets dans une
organisation

Par

Azrach Riham Hind

Projet présenté à l'Institut des sciences et techniques appliquées
En vue de l'obtention du diplôme de master professionnelle IM

Soutenu le 23 juin 2026 devant le jury composé de :

DAHANE Amine

MCA

Président

BEKKAR Azzedine

MCB

Examineur

AIT SI LARBI El Yasmine

MCB

Encadrant

23 Juin 2026

Sommaire

Ce travail vise à concevoir et à implémenter un système de suivi d'objets basé sur les technologies de l'Internet des Objets (IoT), afin de réaliser la localisation des équipements au sein des organisations en temps réel. Une étude a été menée concernant les concepts fondamentaux de l'IoT, les différentes techniques de suivi d'objets ainsi que les principales technologies de positionnement et de communication utilisées dans ce domaine, en environnements intérieurs et en environnements extérieurs.

Le système proposé repose sur l'utilisation du microcontrôleur ESP32 et du module GPS NEO-6M pour la géolocalisation des objets en environnement extérieur, ainsi que sur la technologie Bluetooth Low Energy (BLE) pour les communications à courte portée en environnement intérieur. Les données collectées sont stockées et gérées à l'aide de la plateforme Firebase Cloud Firestore, et une application mobile développée avec Flutter permet le suivi des objets en temps réel.

Les résultats obtenus démontrent l'efficacité du système dans la collecte, la transmission et l'affichage des données de localisation en temps réel. Cette solution contribue à améliorer la gestion des objets et à faciliter leur suivi au sein des organisations.

Mots-clés : Internet des Objets (IoT), Suivi d'objets, ESP32, GPS, Bluetooth Low Energy (BLE), Firebase, Flutter, Géolocalisation.

Abstract

This work aims to design and implement an object tracking system based on Internet of Things (IoT) technologies in order to achieve a real-time localization of equipment within organizations. First, a comprehensive study was conducted on the fundamental concepts of IoT, various object tracking techniques, and the main positioning and communication technologies used in this field, in indoor and outdoor environments.

The proposed system is based on the ESP32 microcontroller and the NEO-6M GPS module for outdoor object geolocation, while Bluetooth Low Energy (BLE) technology is used for short-range communications in indoor environments. The collected data are stored and managed using Firebase Cloud Firestore, and a mobile application developed with Flutter enables real-time visualization and tracking of objects.

The obtained results demonstrate the effectiveness of the system in collecting, transmitting, and displaying location data in real time. This solution contributes to improving asset management and facilitating equipment tracking within organizations.

Keywords: Internet of Things (IoT), Object Tracking, ESP32, GPS, Bluetooth Low Energy (BLE), Firebase, Flutter, Geolocation.

الملخص

يهدف هذا العمل إلى تصميم وتنفيذ نظام لتتبع الأجسام بالاعتماد على تقنيات إنترنت الأشياء، من أجل تحسين عملية تحديد مواقع المعدات داخل المؤسسات في الزمن الحقيقي. تم في البداية دراسة المفاهيم الأساسية لإنترنت الأشياء وتقنيات التتبع المختلفة، بالإضافة إلى أهم تقنيات التوضع والاتصال المستخدمة في هذا المجال وذلك، في البيئات الداخلية والخارجية.

اعتمد النظام المقترح على المتحكم الدقيق ESP32 ووحدة تحديد المواقع GPS NEO-6M لتحديد الموقع الجغرافي للأجسام في البيئات الخارجية، مع استخدام تقنية (Bluetooth Low Energy (BLE للاتصالات قصيرة المدى. كما تم استخدام منصة Firebase Cloud Firestore لتخزين البيانات وإدارتها، في حين تم تطوير تطبيق محمول باستخدام Flutter لعرض مواقع الأجسام ومتابعتها بشكل لحظي.

أظهرت النتائج المتحصل عليها فعالية النظام في جمع البيانات وإرسالها وعرضها في الزمن الحقيقي، مما يساهم في تسهيل عملية تتبع المعدات داخل المؤسسات.

الكلمات المفتاحية: إنترنت الأشياء، تتبع الأجسام، ESP32، GPS، Bluetooth Low Energy، Flutter، Firebase، تحديد المواقع.

Remerciements

Avant tout, je remercie Allah le Tout-Puissant de m'avoir accordé la force, la patience et la persévérance nécessaires pour mener à bien ce travail et atteindre cette étape importante de mon parcours universitaire.

Je tiens à exprimer ma profonde gratitude à mon encadrante, Madame AIT SI LARBI E., pour son accompagnement, sa disponibilité et ses précieux conseils tout au long de la réalisation de ce projet. Son soutien constant, sa rigueur scientifique et ses orientations pertinentes ont largement contribué à l'aboutissement de ce travail. Qu'elle trouve ici l'expression de mon profond respect et de ma sincère reconnaissance.

J'adresse également mes sincères remerciements aux membres du jury, Monsieur BEKKAR A. et Monsieur DAHANE A., pour l'honneur qu'ils me font en acceptant d'examiner et d'évaluer ce mémoire. Je leur suis reconnaissante pour le temps consacré à la lecture de ce travail ainsi que pour les remarques et suggestions qu'ils apporteront afin de l'enrichir.

Je souhaite également remercier l'ensemble des enseignants de l'institut des sciences et techniques (ISTA) de l'université Oran 1 Ahmed Ben Bella qui ont contribué à ma formation universitaire. Grâce à leur savoir, leur dévouement et leurs encouragements, j'ai pu acquérir les connaissances et les compétences nécessaires pour mener à bien mes études et réaliser ce projet.

Mes remerciements les plus profonds sont adressés à mes très chers parents. Aucun mot ne saurait exprimer l'immense gratitude que je leur porte. Merci pour tous les sacrifices consentis, pour votre amour inconditionnel, votre soutien sans faille et vos prières qui m'ont toujours accompagnée. Vous avez été ma force dans les moments difficiles et la source de ma motivation dans chaque étape de mon parcours. Que Dieu vous protège et vous accorde santé, bonheur et longue vie.

Je remercie également mes frères, qui ont toujours été présents à mes côtés par leurs encouragements, leur affection et leur soutien moral. Leur présence a constitué une véritable source de réconfort et de motivation.

Une pensée toute particulière est adressée à Maram, que le destin m'a offerte et que je considère comme ma propre sœur. Tu es cette sœur que ma mère n'a pas mise au monde, mais que mon cœur a choisie. Merci pour ton amour sincère, ta fidélité, ton écoute et ta présence constante à mes côtés. Merci pour tous les moments partagés, les encouragements et le soutien que tu m'as apportés. Ce travail porte aussi une part de ta présence et de ton soutien.

Je tiens à exprimer du fond du cœur mes sincères remerciements à mon fiancé Amine, pour sa patience, sa compréhension et son soutien constant. Merci d'avoir toujours cru en mes capacités, d'avoir été à mes côtés et de m'avoir encouragée dans les moments de doute et de fatigue. Tu as partagé avec moi les difficultés comme les réussites, et ta présence a été une véritable source de force, de sérénité et de confiance. Je t'adresse toute ma reconnaissance et mon affection. Que Dieu te protège et qu'il nous réunisse dans le bonheur.

Mes remerciements vont également à mes chères amies Kedjar Sara et Bouhacene Zahera, pour leur présence, leur amitié sincère et leur soutien tout au long de mon parcours. Un grand merci à Sara pour son aide précieuse, ses encouragements et sa disponibilité dans les moments où j'en avais besoin. Je remercie également Zahera, avec qui j'ai partagé les années d'études, les efforts, les moments de stress mais aussi les beaux souvenirs qui resteront gravés dans ma mémoire. Votre présence a rendu ce parcours plus agréable et plus riche.

Table des matières

Sommaire	ii
Abstract	iii
المخلص	iv
Remerciements.....	v
Table des matières	vii
Liste des abréviations.....	x
Liste des tableaux.....	xiii
Liste des figures	xiv
Liste des équations.....	xv
Introduction.....	1
Chapitre 1 Problématique et état de l’art	3
1.1 Introduction.....	3
1.2 Définition de l’Internet des objets.....	3
1.3 Historique de l’Internet des objets (IoT).....	4
1.4 Applications de l’IoT	5
1.4.1 Maison intelligente (Smart Home).....	6
1.4.2 Hôpitaux intelligents (Smart Hospitals).....	6
1.4.3 Environnements éducatifs intelligents (Smart Educational Environment)	6
1.4.4 Automatisation industrielle intelligente (Smart Industry Automation)	6
1.5 e- Avantages de l’IIoT	9
1.5.1 Transport intelligent (Smart Transportation)	10

1.5.2	Agriculture et environnement intelligents (Smart Agriculture and Environment)	10
1.6	Les défis de l'Internet des objets (IoT)	10
1.6.1	Défis technologiques	11
1.6.2	Défis de sécurité	11
1.6.3	Défis liés aux données	11
1.6.4	Défis de la vie privée et de l'éthique	11
1.6.5	Défis économiques et organisationnels	12
1.6.6	Défis sociétaux et humains	12
1.6.7	Défis environnementaux	12
1.7	Avantages et inconvénients de l'Internet des objets (IoT)	12
1.7.1	Avantages	12
1.7.2	Inconvénients	13
1.8	Suivi des objets (Object Tracking)	14
1.8.1	Défis du suivi d'objet	15
1.8.2	Avantages du suivi des objets	16
1.9	Conclusion :	16
Chapitre 2 Technologies et méthodes utilisées en IoT et en suivi des objets (Object Tracking)		17
2.1	Introduction :	17
2.2	L'architecture de l'IoT	17
2.2.1	Architecture en couches de l'IoT	18
2.2.2	Cloud Computing	18
2.2.3	Edge et Fog Computing	19
2.3	Technologies de positionnement intérieur et extérieur	21
2.3.1	Technologies de positionnement sans fil	21
2.4	Conclusion :	25
Chapitre 3 Conception et implémentation		26

3.1	Introduction :	26
3.2	Architecture globale du système	26
3.3	Choix technologiques et environnement de développement	28
3.3.1	Environnement Matériel (Hardware)	28
3.3.2	Étude de l'autonomie et gestion énergétique du système	30
3.3.3	Environnement Logiciel (Software):	32
3.3.4	Bibliothèques utilisées :	32
3.4	L'Architecture de Cloud Firestore (le back-end) [72]:	33
3.5	L'architecture de l'application Flutter [73]:	34
3.6	Tests et résultats :	34
3.6.1	Test d'acquisition de la position	34
3.6.2	Expérimentations en environnement intérieur	36
3.6.3	Visualisation et analyse du graphe :	39
3.6.4	Temps de réponse de la communication BLE	39
3.6.5	Validation expérimentale en environnement extérieur:	40
3.6.6	Développement de l'application mobile avec Flutter :	43
3.6.7	Initialisation du projet	43
3.7	Conclusion :	51
Conclusion		52
Perspectives		54
Bibliographie		55

Liste des abréviations

API	Application Programming Interface
APK	Android Package Kit
BLE	Bluetooth Low Energy
BPSK	Binary Phase Shift Keying
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESP32	Espressif 32-bit Microcontroller
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IA	Intelligence Artificielle
IDE	Integrated Development Environment
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IR	Infrared

JSON	JavaScript Object Notation
LED	Light Emitting Diode
LoRa	Long Range
LOS	Line of Sight
LPWAN	Low-Power Wide-Area Network
NFC	Near Field Communication
NMEA	National Marine Electronics Association
NoSQL	Not Only SQL
PIR	Passive Infrared Sensor
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SDK	Software Development Kit
SoC	System on a Chip
SQL	Structured Query Language
SSL	Secure Sockets Layer
STA	Station Mode
TCP/IP	Transmission Control Protocol / Internet Protocol

ToA	Time of Arrival
UART	Universal Asynchronous Receiver-Transmitter
UWB	Ultra-Wideband
VCC	Voltage at Common Collector
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Networks

Liste des tableaux

Tableau 1 - Défis identifiés et solutions potentielles pour l'IoT industriel [22]	8
Tableau 2 - Les connexions effectuées	30
Tableau 3 - Relation entre la puissance du signal RSSI, la distance estimée et la qualité de la connexion BLE.	38
Tableau 4 - Extrait des données de localisation enregistrées en temps réel	43

Liste des figures

Figure 1 - Histoire de l'IoT [5]	5
Figure 2 - Architecture du cloud computing [45]	19
Figure 3 - Architecture de l'informatique en périphérie (edge computing) [45]	19
Figure 4 - Architecture de fog computing [45]	20
Figure 5 - Microcontrôleur ESP32.....	28
Figure 6 - NEO-6M GPS Module.....	29
Figure 7 - schéma de câblage et d'interconnexion entre le module GPS NEO-6M et la carte ESP32.....	29
Figure 8 - Affichage des coordonnées GPS (latitude et longitude) sur le moniteur série de l'Arduino IDE.....	35
Figure 9 - Capture d'écran de l'application nRF Connect validant la réception des données géographiques via BLE.....	37
Figure 10 - Évolution temporelle du signal RSSI de l'ESP32 sur l'application nRF Connect.....	39
Figure 11 - Flux de débogage et d'acquisition des données sur le moniteur série.	41
Figure 12 - Interface Cloud Firestore.....	42
Figure 13 - Structure du code source du point d'entrée main.Dart et initialisation des services Firebase	45
Figure 14 - Code source du module de suivi et de gestion du flux de données Firestore.....	46
Figure 15 - Interface principale de l'application affichant la carte dynamique OpenStreetMap pour le suivi en temps réel.	48
Figure 16 - Vue d'ensemble du système de suivi (Matériel).....	49

Liste des équations

Équation 1 – La moyenne arithmétique.....	33
Équation 2 – La loi de décharge nominale.....	33
Équation 3 – Le modèle mathématique de propagation des ondes	40

Introduction

Les développements technologiques dans le domaine de l'informatique, des télécommunications et de l'électronique ont transformé la manière dont les systèmes et les objets interagissent avec leur environnement. L'utilisation de l'internet des objets, Internet of Things en anglais (IoT), est considérée comme une étape importante dans ce développement, en permettant la connexion d'objets physiques capables de collecter, traiter et transmettre des données en temps réel. Cette évolution a ouvert la voie à des applications innovantes dans plusieurs secteurs tels que l'industrie, la santé, le transport et l'agriculture.

Aussi, les entreprises et les organisations sont de plus en plus confrontées à des défis liés à la localisation des équipements, à la surveillance de leurs ressources et à l'optimisation de leurs processus. Notons que la localisation concerne l'intérieur et l'extérieur d'une organisation du point de vue des infrastructures et bâtiments. La perte de temps dans la recherche d'objets et le manque de visibilité en temps réel peuvent donner lieu à des coûts supplémentaires et une perte de temps et donc réduire l'efficacité des systèmes concernés. La mise au point de systèmes de suivi des objets (Object Tracking en anglais) devient alors une solution majeure.

En effet, le suivi des objets permet de détecter, localiser et suivre un objet dans un environnement interne ou externe à l'organisation. Grâce à des solutions technologiques, il est possible de suivre un objet, de reconstituer sa trajectoire, d'analyser son comportement et même de prédire ses déplacements futurs.

Notons que, en utilisant l'IoT, et donc en combinant les données issues de capteurs basés sur le GPS (Global Positioning System), le RFID (Radio Frequency Identification), caméras, etc. et en ajoutant des techniques de traitement, il est possible de concevoir des systèmes de suivi performants, capables de fonctionner en temps réel dans des environnements variés. Ces

systèmes peuvent être utilisés dans plusieurs domaines tels que la gestion logistique et les systèmes industriels.

Dans le but de proposer une solution de suivi des objets d'une organisation, la première étape a été d'étudier les méthodes de suivi des objets qui existent dans la littérature pour pouvoir proposer un système de localisation en temps réel. L'implémentation du système a été réalisée ensuite.

Afin de montrer notre apport, ce mémoire est structuré en trois chapitres :

- Le premier chapitre présente les concepts fondamentaux de l'Internet des Objets et ses domaines d'application, le concept de suivi des objets et un état de l'art des solutions utilisées dans l'application de ce concept.
- Le deuxième chapitre est consacré à l'étude des méthodes de suivi des objets et des technologies de positionnement utilisées. Nous présentons aussi les technologies choisies dans notre projet tout en argumentant nos choix.
- Le troisième chapitre concerne l'implémentation du système proposé. Il détaille les outils utilisés, l'architecture mise en place, les étapes de développement ainsi que la présentation et l'analyse des résultats obtenus.

Une conclusion et des perspectives seront présentées à la fin de ce mémoire.

Chapitre 1

Problématique et état de l’art

1.1 Introduction

L’aptitude de suivre des objets avec précision et fiabilité est devenue un enjeu important dans certains secteurs tels que la logistique et l’industrie. La demande en hausse de solutions de suivi d’objets a conduit à des avancées dans les technologies de localisation, grâce à l’intégration de l’Internet des Objets (IoT). Ce qui permet de collecter et de transmettre les données de position en temps réel, répondant aux exigences de surveillance continue [1].

Dans ce chapitre, nous présenterons une vue globale de l’Internet des Objets, en abordant ses concepts fondamentaux, ses architectures et ses domaines d’application, avant de nous focaliser sur les systèmes de suivi d’objets, leurs principes de fonctionnement et leur importance dans les systèmes intelligents modernes. Nous présenterons également un état de l’art pour pouvoir positionner notre contribution.

1.2 Définition de l’Internet des objets

L’Internet des Objets est un réseau d’objets physiques connectés, équipés de capteurs, de logiciels et d’une connectivité réseau. Ces objets, étant par exemple des appareils, des capteurs..., peuvent collecter, échanger des données et interagir avec leur environnement, généralement de manière autonome.

Un des défis les plus importants est de garantir une communication éco- énergétique entre les objets, étant mobiles et fonctionnant sur batterie [2].

Nous précisons que nous avons choisi d'utiliser dans ce mémoire l'abréviation anglaise IoT pour l'Internet des Objets vu l'utilisation courante de ce terme dans le domaine de l'enseignement et de la recherche.

1.3 Historique de l'Internet des objets (IoT)

Concernant l'Internet des objets, nous avons une évolution progressive, passant de simples appareils connectés à une technologie globale.

En effet, ce concept remonte au début des années 1980 avec l'apparition du premier dispositif utilisant l'IoT, qui consiste en une machine distributrice de boissons à l'Université Carnegie Mellon en Pennsylvanie [3].

Dans les années 1990, les avancées réalisées dans le domaine de l'identification par radiofréquence (RFID) sont considérées comme une base technologique pour le développement des applications IoT.

En 1999, Kevin Ashton a introduit le terme « Internet of Things » au sein des laboratoires Auto-ID du Massachusetts Institute of Technology (MIT), afin de décrire un système dans lequel des objets physiques sont connectés à Internet en utilisant des capteurs [4].

Dans les années 2000, l'intégration des technologies RFID, des communications sans fil et des systèmes micro électromécaniques (MEMS) a permis la surveillance et le contrôle à distance des objets connectés.

Pendant les années 2010, l'IoT est devenu un concept considérablement utilisé, vu la hausse de l'utilisation des smartphones, la baisse des coûts des capteurs et la vulgarisation des réseaux Internet à haut débit. Des initiatives telles que le concept Internet of Everything proposé par Cisco ont ajouté les personnes et les processus à l'IoT [4].

Aujourd'hui, l'Internet des Objets est intégré dans plusieurs secteurs tels que la santé, l'agriculture, l'industrie et les villes intelligentes [4].

Les tendances actuelles se concentrent sur l'utilisation, entre autres, de l'IA (Intelligence Artificielle) pour la prise de décision locale, ainsi que sur la résolution des défis liés à la normalisation, à la sécurité et à la protection de la vie privée [5].

La figure 1 montre les étapes les plus importantes de l'histoire de l'IoT.

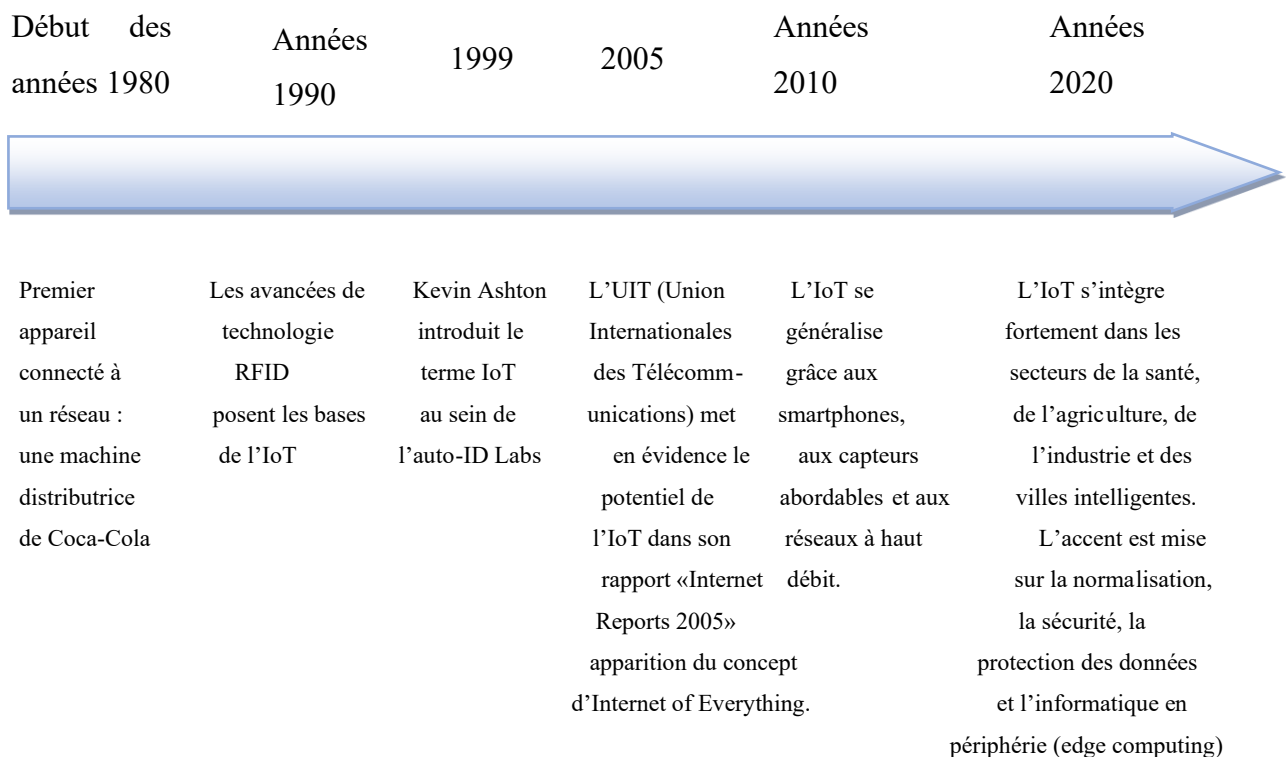


Figure 1 - Histoire de l'IoT [5]

1.4 Applications de l'IoT

Concernant l'utilisation de l'IoT, les appareils utilisés actuellement sont presque tous des appareils intelligents, notons que l'IoT permet la communication entre ces dispositifs afin qu'ils puissent partager de l'information [6]. Aussi, les applications intelligentes avec des capacités autonomes peuvent améliorer notre qualité de vie [7]. Nous citons dans ce qui suit quelques applications de l'IoT :

1.4.1 Maison intelligente (Smart Home)

À l'intérieur d'une maison, l'IoT permet de contrôler des accessoires à distance selon les besoins: contrôle des appareils électroménagers, suivi du compteur d'énergie, sécurité domestique, ...

À l'extérieur d'une maison, les capteurs peuvent surveiller les facteurs environnementaux comme l'intensité lumineuse, l'humidité, la température, le taux d'eau dans le sol, ... [8].

1.4.2 Hôpitaux intelligents (Smart Hospitals)

L'IoT est également utilisée dans le domaine de la santé. En effet, les hôpitaux peuvent fournir des bracelets intelligents flexibles équipés de RFID aux patients, pour surveiller des paramètres tels que le rythme cardiaque, la pression artérielle, la température ... [9].

1.4.3 Environnements éducatifs intelligents (Smart Educational Environment)

Les dispositifs basés sur l'IoT sont aussi utilisés dans le secteur éducatif [8] pour enregistrer la présence des étudiants et du personnel, suivre les livres de bibliothèques, identifier le matériel de laboratoires, ...

1.4.4 Automatisation industrielle intelligente (Smart Industry Automation)

Dans ce contexte, les capteurs constituent un élément essentiel dans les systèmes de climatisation, de sécurité et d'autres installations dans les usines. Notons qu'un système sans fil offre la flexibilité nécessaire pour ajuster la configuration selon les besoins. Il s'agit ici d'un sous-réseau IoT [2].

Notons que l'industrie 4.0 est un concept introduit en Allemagne en 2011 [10]. Dans ce contexte, pour obtenir un environnement de production axé sur les besoins du client et visant

une amélioration continue, les machines deviennent des entités capables de collecter, analyser et fournir des recommandations basées sur les données [11].

Aussi, le fait d'automatiser et de numériser un environnement de production crée une chaîne de valeur pouvant améliorer la performance du système en question et gérer les risques. L'usine intelligente évolue donc vers un modèle de production innovant et développé du point de vue technologique [12].

Notons qu'il existe un lien entre l'IoT et l'industrie 4.0. En effet, l'étude de l'IoT est essentielle pour comprendre l'effet du big data et l'analyse de données sur les réseaux cyber-physiques et les processus industriels dans le cadre de l'industrie 4.0 [13].

Aussi, les machines peuvent automatiser des tâches, ce qui était impossible lors des révolutions industrielles précédentes [14]. L'utilisation de l'IoT est également importante pour les applications impliquant des écosystèmes interconnectés, comme les installations de fabrication intelligentes [15].

Grâce à ces données, les organisations peuvent identifier et corriger rapidement les erreurs ou inefficacités, comme par exemple dans leurs chaînes d'approvisionnement, ce qui permet d'atteindre une efficacité continue du point de vue financier et opérationnel [16]. L'intégration de l'IoT permet donc aux entreprises industrielles de prendre des décisions commerciales adéquates [17]. Nous présentons dans ce qui suit quelques notions que nous avons jugées importantes dans le contexte de l'IIoT (Industrial Internet of Things).

a- Architecture de l'IIoT

L'architecture de l'IIoT peut être perçue de façon générale comme un système contenant plusieurs niveaux:

Niveau 1 : Les composants physiques ;

Niveau 2 : Les méthodes de communication ;

Niveau 3 : l'agrégation des données ;

Niveau 4 : Stockage des données et analyse ;

Niveau 5 : Interface utilisateur [18]

Notons que les systèmes cyber physiques peuvent être définis comme des technologies permettant de gérer des systèmes interconnectés de composants physiques et leurs capacités vues du côté informatique [19].

b- Le Big Data dans l'IIoT

Le développement rapide de l'IIoT a entraîné une augmentation importante de la quantité de données produites. Pour garantir un traitement efficace, une prise de décision en temps réel et une automatisation par rapport à cette augmentation, l'utilisation de l'analyse des Big Data, de l'apprentissage automatique (Machine learning en anglais, ML) et de l'IA est nécessaire [20]. Les Big Data correspondent aux données volumineuses et complexes collectées à partir de capteurs industriels par exemple, caractérisés eux-mêmes par leur volume, vitesse et diversité élevés [21]. Pour gérer ces données, le cloud computing, l'edge computing et le fog computing sont utilisés pour permettre un traitement et un stockage en temps réel.

d- Défis de recherche et solutions potentielles pour l'IoT industriel

La mise en œuvre de solutions basées sur l'IoT en industrie fait face à plusieurs défis. Le tableau 1 montre quelques défis retenus dans le contexte de recherche de solutions pour l'IIoT.

Tableau 1 - Défis identifiés et solutions potentielles pour l'IoT industriel [22]

Défis	Solutions potentielles
Fiabilité et résilience	Capacité des systèmes IIoT de pouvoir continuer à fonctionner malgré des interruptions ou des pannes.
Sécurité et confidentialité	Disponibilité de mécanismes de sécurité robustes tels que les systèmes de détection d'intrusion... Protection des données.
Analyse coût-bénéfice	Évaluer les gains à long terme grâce à des modèles d'analyse coût-bénéfice ...

Gestion et analyse des données	Utiliser des solutions de stockage et des plateformes analytiques évolutives pour la gestion et l'exploitation des données IIoT de manière efficace.
Interaction homme-machine	Améliorer l'interaction grâce à des interfaces utilisateur intuitives ...
Impact environnemental	Utiliser des composants recyclables, des solutions de conception efficaces ...
Efficacité énergétique	Prolonger la durée de vie des dispositifs IIoT avec du matériel économe en énergie et des protocoles de communication de basse consommation ...
Traitement des données en temps réel	Réduire la latence avec l'edge et le fog computing pour traiter les données près de la source. Étudier des algorithmes et des méthodes de traitement pour la prise de décision en temps réel.

1.5 Les avantages de l'IIoT

L'IIoT offre plusieurs avantages, nous en citons :

- Permet de prévoir des pannes, en utilisant des données générées en temps réel par les systèmes basés sur IIoT.
- Les technologies IIoT permettent de détecter les problèmes potentiels concernant les équipements et les installations.
- Vu que les produits sont connectés à internet, l'IIoT permet aux fabricants de recueillir et analyser des données.
- En utilisant l'IIoT, les acteurs d'une chaîne logistique tels que les fournisseurs, les fabricants et les clients peuvent utiliser des systèmes de gestion des actifs pour le suivi de la position et l'état des produits par exemple.

- La possibilité d’avoir des alertes instantanées, envoyées dans le cas où les produits sont endommagés ou risquent de l’être, ce qui permet d’intervenir de façon immédiate ou de mener des actions de prévention [23].

1.5.1 Transport intelligent (Smart Transportation)

L’objectif d’un système de transport intelligent est de superviser les villes intelligentes grâce à l’utilisation de technologies de communication avancées. Les conditions météorologiques défavorables peuvent affecter négativement les systèmes de transport conventionnels. La mise en œuvre d’un système [24] intégrant la technologie RFID permet par exemple le suivi, la surveillance et l’identification intelligents des véhicules.

Aussi, les applications en temps réel des systèmes de navigation par satellite et des capteurs sont également possibles pour les véhicules, ... Notons que les capteurs de ces véhicules peuvent fournir des données en temps réel pour déterminer par exemple si une maintenance est nécessaire [25].

1.5.2 Agriculture et environnement intelligents (Smart Agriculture and Environment)

Il est nécessaire d’améliorer les méthodes agricoles pour satisfaire les besoins d’une population en produits agricoles. La technologie utilisant des capteurs intégrés peut être employée dans le domaine de l’agriculture pour détecter et transmettre des informations importantes concernant l’environnement. Nous précisons que la température, l’humidité, la pression et d’autres variables sont des paramètres courants de l’environnement [26].

1.6 Les défis de l’Internet des objets (IoT)

L’Internet des Objets fait face à plusieurs défis majeurs qui sont à la fois technologiques, sécuritaires, économiques, sociétaux et environnementaux [27].

1.6.1 Défis technologiques

L'un des principaux défis de l'IoT réside dans l'hétérogénéité des objets connectés, des protocoles de communication et des plateformes logicielles. Nous avons aussi les contraintes matérielles des objets connectés, telles que la faible capacité de calcul, la mémoire limitée et la consommation énergétique [27].

1.6.2 Défis de sécurité

Les objets connectés exposent les systèmes à des risques tels que le piratage, l'usurpation d'identité et la falsification des données. D'où la nécessité de se doter de solutions de sécurisation des communications, d'authentification des objets et de protection des infrastructures.

En effet, une faille au niveau d'un dispositif peut compromettre un réseau. La protection de l'infrastructure est donc importante [28].

1.6.3 Défis liés aux données

Notons dans ce contexte que l'IoT génère de grandes quantités de données. La gestion, le stockage et le traitement de ces dernières représentent un défi important pour les concepteurs de ces systèmes [27].

1.6.4 Défis de la vie privée et de l'éthique

Les objets connectés peuvent collecter des données personnelles relatives aux habitudes, aux comportements et à la localisation des utilisateurs. Ce qui mène à poser des questions d'ordre éthique concernant le respect de la vie privée [27].

1.6.5 Défis économiques et organisationnels

L'utilisation de l'IoT implique des coûts importants. Aussi, les entreprises doivent fournir des efforts pour adapter leurs modèles économiques et leurs processus organisationnels afin d'exploiter efficacement les données issues de l'IoT [27].

1.6.6 Défis sociétaux et humains

Les changements induits par l'IoT dans la vie des individus transforment les métiers et les modes de travail, ce qui demande l'acquisition de nouvelles compétences. Nous citons aussi le problème de fracture numérique entre les populations ayant accès aux technologies et celles qui n'en ont pas [27].

1.6.7 Défis environnementaux

La propagation des objets connectés donne lieu à une augmentation des déchets électroniques et de la consommation énergétique.

1.7 Avantages et inconvénients de l'Internet des objets (IoT)

L'interconnexion des objets qui échangent des données mène à des opportunités, tout en posant des inconvénients, nous citons ces points dans ce qui suit [29]:

1.7.1 Avantages

- **Automatisation**

L'IoT permet d'automatiser des tâches, ce qui améliore la productivité dans les environnements industriels par exemple.

- **Surveillance à distance**

L'IoT permet de surveiller et de contrôler des systèmes ou des équipements à distance.

- **Amélioration de la précision**

Les dispositifs IoT fournissent des données fiables en temps réel, permettant d'améliorer la précision des opérations et d'optimiser la gestion des ressources.

- **Stimulation de l'innovation**

L'IoT favorise le développement de nouvelles applications et services intelligents dans plusieurs secteurs tels que la santé, l'agriculture, l'énergie, les transports et l'industrie.

1.7.2 Inconvénients

- **Problèmes de sécurité**

La hausse du nombre d'objets connectés rend les systèmes plus exposés aux cyberattaques et au piratage des données.

- **Atteinte à la vie privée**

La collecte de données personnelles peut poser des problèmes de confidentialité concernant la protection de la vie privée des utilisateurs.

- **Complexité technique**

Les systèmes IoT reposent sur des architectures technologiques complexes qui nécessitent des compétences spécialisées pour leur gestion et leur maintenance.

- **Coûts de mise en œuvre**

La mise en œuvre d'une installation IoT peut être coûteuse vu les investissements nécessaires en capteurs, dispositifs connectés, réseaux déployés et systèmes spécialisés dans la sécurité.

1.8 Suivi des objets (Object Tracking)

Le suivi des objets est une technique utilisée pour assurer la localisation et la surveillance d'un objet dans un environnement donné. Nous précisons que le suivi d'objets se fait selon deux catégories selon l'environnement en question :

- Un suivi d'objets à l'intérieur d'une organisation (indoor) ;
- Un suivi d'objets à l'extérieur d'une organisation (outdoor).

Nous présentons dans ce qui suit un état de l'art concernant les deux catégories :

1.8 Etat de l'art :

Nous retrouvons dans la littérature plusieurs travaux dédiés à l'élaboration de solutions pour le suivi d'objets. En ce qui concerne le suivi en environnements externes, nous avons :

Nous en citons [30] qui concerne la création d'un système basé sur le GPS et le suivi de passagers en utilisant l'ESP8266 NODE MCU Bord comme serveur. Ce module interagit avec un module GPS et des unités GSM/GPS reliées à une antenne GPS. L'étude concerne l'analyse des retards de transmission des données, en se concentrant sur le temps de traitement, la durée de transmission, le temps de réponse du serveur, la latence du réseau et les besoins en énergie. L'interface de l'application a été mise en œuvre en utilisant Flutter et l'API OpenStreetMap, la partie Backend a été établie en utilisant Firebase.

Aussi, dans [31], les auteurs ont implémenté un système de suivi d'objets en utilisant la technologie GPS. Ce projet a permis d'ajouter à la localisation l'adresse correspondante en utilisant la méthode de reverse geocoding pour la conversion. Les auteurs ont suivi une méthodologie qui se base sur la réception des signaux du module GPS placé sur l'objet suivi, émis par le satellite, le microcontrôleur transforme les signaux en coordonnées et les envoie à la base de données Google Firebase puis la transmission des coordonnées à l'application de suivi d'objets.

Nous citons également [32], où les auteurs présentent un système intégrant l'IoT en utilisant le microcontrôleur ESP32 et des capteurs gyroscopes pour permettre la détection de vibrations, des capteurs de charge pour la surveillance du poids, un GPS pour la localisation et le réseau GSM pour la communication. Une application mobile a également été développée en utilisant Flutter pour le monitoring en temps réel et l'envoi de notifications d'alerte.

D'autres travaux ont utilisé la technologie GPS ainsi qu'une solution indoor afin de résoudre des problèmes de localisation, nous citons les travaux de [33] où la technologie GPS-Lora a été utilisée pour le développement d'une application de sécurité de stationnement de véhicules, avec notification instantanée. Un contrôleur ESP32 a été utilisé équipé d'un module GPS neo-6M et d'un transceiver LoRa SX1278, qui transmet les coordonnées d'un véhicule à une base de données Firebase en temps réel. Une application mobile basée sur Flutter récupère les données, visualise la position du véhicule sur Google Maps et déclenche des notifications lorsque le véhicule sort de zones de géo repérage prédéfinies.

Concernant l'amélioration des performances, nous citons [34] où les auteurs ont utilisé un microcontrôleur ESP32 dans l'edge computing pour évaluer les gains de performance obtenus grâce à l'intégration d'un serveur basé sur l'edge computing et du cloud. Une infrastructure logicielle complémentaire pour le serveur cloud a été conçue afin de prendre en charge le traitement en périphérie (edge). Un modèle d'apprentissage profond pour la reconnaissance d'objets a été entraîné et déployé sur le serveur edge. Les résultats expérimentaux montrent que le serveur edge réduit l'utilisation de la bande passante (donc la charge) et la latence, allégeant ainsi la charge sur le serveur cloud.

Pour le suivi en environnement interne, nous avons les travaux de [35] qui a servi de base.

1.8.1 Défis du suivi d'objet

Plusieurs facteurs affectent les performances d'un algorithme de suivi d'objet. Parmi ces facteurs, nous mentionnons les suivants [36]:

- Similarité de couleur entre l'objet cible et le fond
- Mouvement de caméra ;
- Rotation dans le plan de l'image ;
- Objet de petite taille ;
- Objet en mouvement rapide ;
- Nécessité d'un suivi en temps réel.
- ...

1.8.2 Avantages du suivi des objets

Le suivi des objets présente plusieurs avantages, notamment [37] :

- Estimation précise des paramètres : Le suivi permet d'estimer les paramètres de localisation, de vitesse et de distance des objets en mouvement.
- Segmentation précise : Une segmentation précise des objets de l'arrière-plan est nécessaire pour un suivi efficace.
- Amélioration des performances : Le suivi permet de détecter des discontinuités.
- Robustesse : Des méthodes comme le filtre de Kalman et le modèle de mixture gaussienne offrent des solutions robustes aux problèmes de détection et de suivi d'objets en mouvement.

1.9 Conclusion :

Nous avons présenté dans ce chapitre les notions élémentaires liées à l'internet des objets à partir de son apparition jusqu'aux utilisations récentes de cette notion.

Le suivi des objets s'inscrit dans les disciplines utilisant l'internet des objets, visant à localiser des objets en mouvement dans le temps afin de faciliter le suivi de ces derniers dans les organisations.

Chapitre 2

Technologies et méthodes utilisées en IoT et en suivi des objets (Object Tracking)

2.1 Introduction :

Vu le développement rapide des technologies numériques, de l'IoT, des réseaux sans fil et des systèmes embarqués, le suivi des objets est devenu une solution essentielle dans de nombreux secteurs. Aujourd'hui, les entreprises cherchent à automatiser leurs processus, réduire les pertes de matériel, améliorer la gestion des équipements et optimiser le temps de travail. Dans ce contexte, les systèmes de suivi d'objets représentent une technologie relevant de la stratégie d'une organisation.

Le suivi d'objet consiste à localiser un objet dans un espace donné, connaître sa position actuelle, mémoriser ses déplacements et parfois prévoir sa trajectoire future. Cette technologie peut être utilisée pour suivre des outils industriels, des véhicules, des colis, des machines mobiles, du personnel ou tout autre équipement important.

Dans les environnements industriels, il arrive fréquemment que les employés perdent du temps à chercher un outil ou un équipement déplacé. Grâce à un système de suivi intelligent, il devient possible d'identifier rapidement l'emplacement exact de l'objet recherché via une application mobile ou une interface informatique.

2.2 L'architecture de l'IoT

Le paradigme de l'IoT a été développé par rapport au contexte où l'information pouvait être collectée à partir de plusieurs sources et traitée par des technologies dans un environnement

hétérogène [38]. Dans ce cas, les fonctionnalités et les services similaires peuvent être regroupés dans la même couche dans un modèle IoT, ceci pour faciliter le développement et l'amélioration de l'architecture de chaque couche [39].

2.2.1 Architecture en couches de l'IoT

L'IoT peut être représenté par cinq couches : physique, liaison de données, réseau, transport et application [40]. Nous décrivons ces couches dans ce qui suit :

- **Couche physique (perception):** Cette couche détecte les caractéristiques physiques de l'environnement grâce à des capteurs comme RFID, WSN (Wireless Sensor Networks) ou GPS. Elle convertit les données en signaux numériques pour la transmission [41].
- **Couche de liaison de données:** Cette couche gère la détection des erreurs, la synchronisation des trames, l'adressage des émetteurs/récepteurs et la prévention des collisions. Les protocoles dépendent du contrôle d'accès, de la vitesse, de la topologie et de la consommation d'énergie [42].
- **Couche réseau:** Assure le routage des données à travers des équipements tels que des switches, routeurs et des technologies comme 3G, 4G, 5G, Wi-Fi (Wireless Fidelity), ZigBee.
- **Couche de transport:** Travaille avec la couche application pour transmettre les données sans erreur et garantit l'intégrité, la fiabilité et l'ordre des paquets [43].
- **Couche d'application :** Point d'accès pour les développeurs IoT, permet la création d'applications.

2.2.2 Cloud Computing

Le cloud permet de traiter de grandes quantités de données générées par les objets connectés, avec flexibilité et évolutivité. Une architecture cloud IoT contient [44] :

- **Couche physique :** Collecte les données à partir des différents dispositifs.
- **Couche passerelle :** Traite et prépare les données pour le cloud.

- **Services cloud** : Appliquent des algorithmes analytiques et fournissent le stockage, les bases de données, les serveurs et les gestionnaires d'événements.

Avantages : Stockage et traitement des ensembles de données massifs, coûts réduits, faible impact écologique.

Limites : Risques liés à la sécurité, latence, bande passante limitée [44].

La figure 2 montre l'architecture du cloud computing.

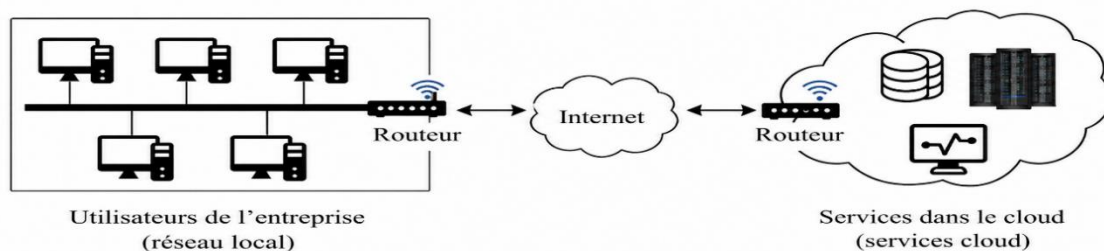


Figure 2 - Architecture du cloud computing [45]

2.2.3 Edge et Fog Computing

- **Edge computing** : Consiste en traitement des données à proximité des dispositifs, réduit la latence. La figure 4 montre l'architecture de l'informatique en périphérie (edge computing).

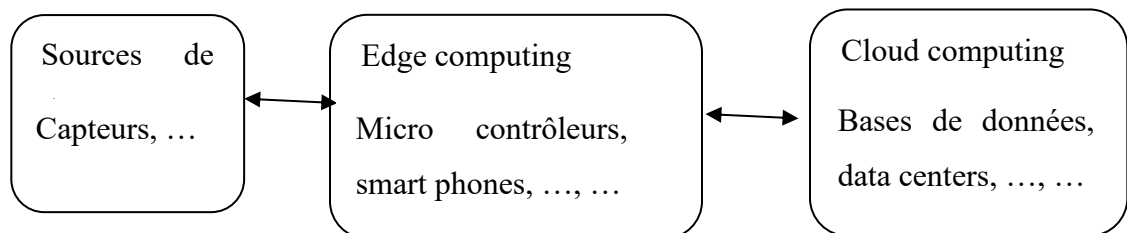


Figure 3 - Architecture de l'informatique en périphérie (edge computing) [45]

- **Fog computing** : Approche hybride combinant les ressources locales et le cloud. Elle prend en charge les éléments suivants:

- **Surveillance** : Suivi de l'énergie, de la disponibilité et de la performance.
- **Prétraitement** : analyse et filtre les données avant l'envoi au cloud.
- **Stockage à court terme** : Concernant la sauvegarde, la redistribution et le cache.
- **Sécurité** : Permet le chiffrement et la protection contre les accès non autorisés.

Avantages : prise de décision en temps réel et sécurité.

Défis : hétérogénéité, charge non déterministe, liens imprévisibles, gestion des ressources en temps réel [45]. La figure 3 montre l'architecture du fog computing.

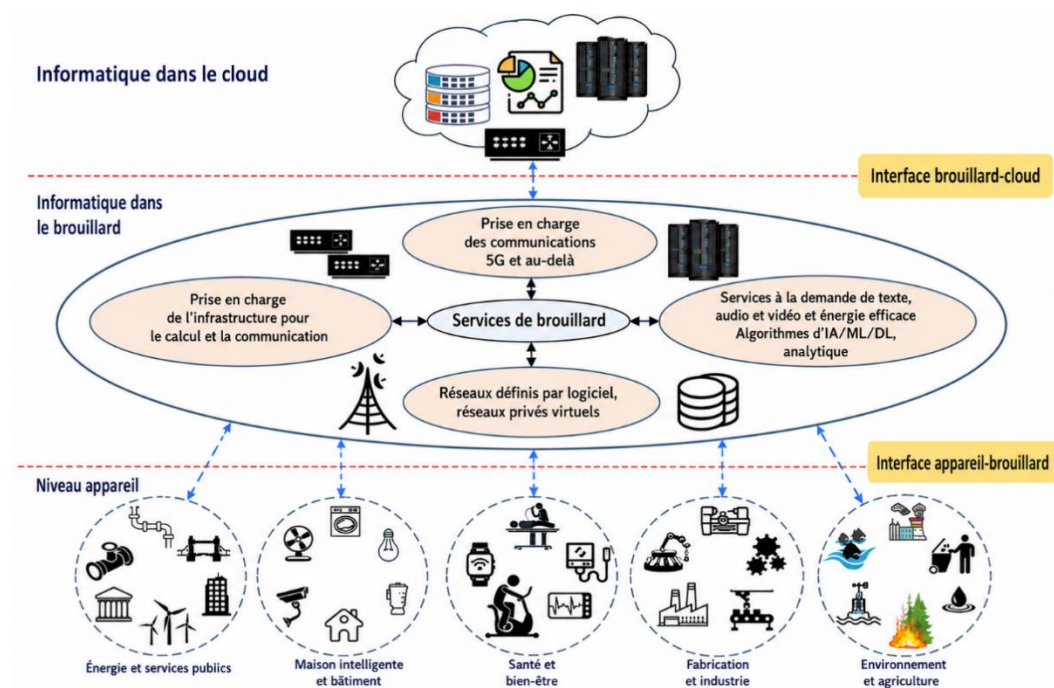


Figure 4 - Architecture de fog computing [45]

2.3 Technologies de positionnement intérieur et extérieur

Ces différents types de technologies sans fil présentent des caractéristiques. La dispersion, la réflexion, la diffraction et la propagation sont des facteurs qui influencent les signaux [46].

La représentation se base sur la portée de ces technologies.

Nous nous sommes inspirés de [47] pour le choix des technologies à utiliser dans ce projet, où nous avons une comparaison entre les technologies de positionnement.

En effet, les auteurs citent plusieurs solutions pour le suivi d'objets dans un environnement intérieur. Ces solutions sont: l'utilisation du RFID, sensible à la température, ce qui ne correspond pas à notre contexte, RFID/Bluetooth dont la précision est faible, RFID/INS dont le coût est assez élevé, UWB qui est une technologie en développement, Zigbee : ayant une faible précision, LANDMARC, Compass ayant une précision limitée, whereNet dont le coût des tags est élevé, wifi, NRF et Bluetooth.

Vu que nous sommes dans un contexte d'une organisation dont :

- La zone est conçue en plusieurs secteurs (dans la zone elle-même),
- Un même objet peut être utilisé dans plusieurs entrepôts ou ateliers.

Nous avons choisi d'utiliser une solution qui allie le suivi d'objets en intérieur et en extérieur. Pour cela nous avons opté pour l'utilisation de l'ESP32, qui permet d'avoir une connectivité Wifi, pour un positionnement GPS à l'extérieur et une connectivité Bluetooth, pour un positionnement en intérieur.

2.3.1 Technologies de positionnement sans fil

1) Wi-Fi

Une technologie de communication sans fil locale principalement destinée à l'accès à Internet via un point d'accès fixe ou un hotspot. Le Wi-Fi est une marque déposée de la Wi-Fi Alliance, responsable des tests et de la certification de la conformité des équipements à la norme IEEE 802.11.

Les normes IEEE 802.11 (Wi-Fi) utilisent les bandes 2,4 GHz, 5 GHz et 60 GHz pour atteindre des débits de l'ordre de dizaines ou centaines de Mbps, avec une portée modérée mais une

consommation d'énergie élevée [48]. Cette norme possède plusieurs révisions : 802.11a, 802.11b et 802.11g, généralement incompatibles entre elles.

En 2016, la Wi-Fi Alliance a annoncé Wi-Fi HaLow basé sur la technologie IEEE 802.11ah. HaLow fonctionne en dessous de 1 GHz, permettant une connectivité à plus longue portée et à faible consommation par rapport au Wi-Fi classique.

Ainsi, HaLow permet des applications à faible consommation d'énergie telles que les maisons intelligentes, l'automobile, la santé numérique, l'industrie, l'agriculture et les villes intelligentes.

HaLow étend le Wi-Fi à la bande 900 MHz avec une connectivité basse consommation pour les objets connectés et capteurs. La portée est presque doublée par rapport au Wi-Fi actuel. HaLow améliore aussi la robustesse dans des environnements difficiles avec des obstacles comme les murs.

Le Wi-Fi est de plus en plus utilisé dans les environnements intérieurs tels que les hôtels, cafés, aéroports et bâtiments, permettant la localisation et l'identification d'objets et de personnes pour des applications basées sur la position [49].

2) Infrarouge

La technologie infrarouge est utilisée dans des dispositifs transmettant des données via la lumière infrarouge. Certaines entreprises combinent RF (Radio Frequency) et infrarouge pour obtenir la position d'un objet.

Les balises émettent des signaux infrarouges et RF contenant un identifiant unique pour le suivi des objets ou des équipements [50].

Le RF permet une localisation générale, tandis que l'infrarouge fournit une résolution plus précise.

Si un autre dispositif reçoit le signal infrarouge d'un appareil, ces dispositifs sont probablement dans la même zone, car l'infrarouge ne traverse pas les murs. De plus, la lumière infrarouge peut être réfléchiée par les murs, donc la ligne de vue directe (LOS) n'est pas toujours nécessaire.

3) RFID

Ce système est utilisé pour stocker et récupérer des données via la transmission électromagnétique d'un circuit intégré par radiofréquence.

Le signal transmis peut traverser la majorité des matériaux de construction, offrant donc une bonne portée en intérieur. Sa vitesse de propagation est élevée, environ 320 m/s.

Cette technologie a une large gamme de fréquences comparée à l'infrarouge et aux ultrasons. Cela dit, des interférences peuvent apparaître à cause des métaux ferromagnétiques ou des émissions radio proches. Des facteurs comme la température, l'humidité ou la structure du bâtiment peuvent aussi impacter la propagation [51].

4) NFC

La NFC (Near Field Communication) [52] est une technologie de communication sans fil à très courte portée (quelques centimètres), basée sur le RFID. Elle fonctionne à 13,56 MHz.

Elle utilise les variations du champ magnétique pour la transmission des données et possède deux modes: actif et passif. Cette technologie permet une communication bidirectionnelle.

5) Ultrasonics

Ces systèmes fonctionnent à haute fréquence pour le positionnement en environnement intérieur. Ils permettent d'obtenir une estimation de la position des objets avec une vitesse de propagation du son de 343 m/s et offrent une bonne précision. Ils nécessitent des balises ultrasoniques servant d'émetteurs ou de récepteurs [53].

Les avantages sont la simplicité et le faible coût. L'inconvénient est que les ultrasons ne traversent pas les murs [54].

6) GPS

Le GPS fonctionne grâce à un réseau de 24 satellites en orbite autour de la terre à une altitude d'environ 20 200 km, avec des trajectoires synchronisées afin de couvrir la surface terrestre [55].

7) RADAR

Le RADAR est un système de localisation et de suivi en intérieur basé sur la technologie WaveLAN de la norme IEEE 802.11 [56]. Au niveau d'une station de base, RADAR mesure le niveau du signal reçu (RSS) ainsi que le rapport signal/bruit des signaux envoyés par les dispositifs sans fil.

8) Bluetooth

Le Bluetooth est une technologie sans fil à courte portée utilisant des ondes radio, qui fonctionne dans la bande de fréquences industrielle, scientifique et médicale (ISM). Sa version la plus récente, 4.0, permet un débit de transmission pouvant atteindre 24 Mbit/s, sans nécessiter de licence, et fonctionne spécifiquement dans la plage de fréquences comprise entre 2,4 GHz et 2,485 GHz. Le Bluetooth est le nom de la spécification IEEE 802.15.1, définissant une norme mondiale pour la transmission sans fil d'informations entre des dispositifs, via une liaison radiofréquence sécurisée, à courte portée et sans licence [57], [58].

9) ZigBee

ZigBee est le nom donné à des protocoles basés sur la norme IEEE 802.15.4, conçue pour créer des réseaux de communication sans fil à faible débit de données, fonctionnant à différentes fréquences comprises entre 868 MHz et 2,4 GHz [59], [60].

La fréquence de communication la plus utilisée est de 2,4 GHz, qui permet d'avoir des débits allant jusqu'à 40 Kbps sur 16 canaux différents. ZigBee fonctionne aussi à 915 MHz (aux États-Unis, 20 Kbps sur 10 canaux) ou à 868 MHz (en Europe, 20 Kbps sur un seul canal).

Il s'agit d'une des rares technologies de communication sans fil orientées vers la surveillance de l'environnement et la collecte de données à l'aide de capteurs, tout en optimisant la consommation d'énergie et les coûts [61], [62].

10) LoRa

LoRa, abréviation de Long Range, est une spécification d'un réseau étendu à faible consommation d'énergie (LPWAN) proposée par la LoRa Alliance [63]. Elle est destinée à la communication de dispositifs peu coûteux, à faible consommation et alimentés par batterie [62].

11) Sigfox

Sigfox [64] est un opérateur de réseau LPWAN qui propose des solutions de connectivité IoT de bout en bout, basées sur ses technologies propriétaires. Il met en œuvre ses propres stations de base équipées de radios cognitives définies par logiciel et les connecte à des serveurs back-end via un réseau basé sur IP.

2.4 Conclusion :

Ce chapitre a permis d'explorer les différentes méthodes de suivi des objets ainsi que les technologies associées utilisées dans les systèmes modernes. Nous avons d'abord défini le concept de suivi d'objets et mis en évidence son importance dans plusieurs domaines d'application tels que l'industrie.

Ensuite, nous avons présenté les principales approches de suivi, notamment le suivi.

Enfin, nous avons étudié les technologies de positionnement et de communication (comme le GPS, le Wi-Fi, le RFID et le Bluetooth) qui permettent d'effectuer le suivi d'objets dans des environnements réels, dans le cadre des applications IoT.

Cette étude théorique constitue une base essentielle pour la suite de ce projet. Elle nous permet d'effectuer les choix techniques adoptés dans ce projet.

Chapitre 3

Conception et implémentation

3.1 Introduction :

Après l'étude théorique présentée dans les chapitres précédents, cette phase consiste à réaliser le système de suivi d'objets proposé. L'objectif est de développer un prototype capable de localiser un objet à l'aide du GPS en environnement extérieur, tout en exploitant la technologie Bluetooth Low Energy (BLE) pour les expérimentations de communication en environnement intérieur. Les données collectées sont transmises vers une base de données NoSQL de Firebase pour être visualisées à distance via une application mobile développée sous Flutter.

3.2 Architecture globale du système

L'architecture de la solution développée adopte une structure en couches, permettant une bonne modularité, évolutivité et séparation des fonctions en suivant l'architecture proposée par [65]. Chaque couche joue un rôle spécifique dans le processus global de localisation, de transmission et de visualisation des données.

1) Couche de perception (Matériel)

Cette couche représente la partie physique du système et est responsable de l'acquisition des données. Elle est constituée de :

La carte ESP32 agit comme unité centrale de traitement embarquée.

- **Le module GPS NEO-6M**, chargé de capter les signaux satellites et de fournir les coordonnées géographiques (latitude, longitude).

2) Couche de Communication (Transport)

Cette couche assure le transfert des données entre le système et les plateformes externes. Correspond à la couche réseau dans [65].

Deux technologies de communication sont utilisées :

- **Wi-Fi (ESP32 → Firebase) :**

Utilisé pour l'envoi des données vers le cloud. Il permet une transmission à longue distance et en temps réel vers la base de données Firebase.

- **Bluetooth Low Energy (BLE) :**

Utilisé pour la communication locale et les tests à courte portée entre l'ESP32 et un smartphone. Cette technologie permet de transmettre les données de manière rapide et à faible consommation énergétique.

- **Application nRF Connect :**

Utilisée comme outil de test pour visualiser les données envoyées via BLE par l'ESP32. Elle permet de vérifier le bon fonctionnement de la communication Bluetooth avant l'intégration finale dans l'application mobile.

3) Couche de Service (Cloud)

Cette couche est assurée par Firebase, qui centralise le stockage des données via sa base de données en temps réel (Cloud Firestore) et gère l'authentification des utilisateurs. Elle correspond à la couche de traitement dans [65].

4) Couche Application (Interface)

Cette couche correspond à l'interface utilisateur finale. Elle est développée avec Flutter et offre une interface ergonomique permettant le suivi cartographique et le contrôle du système.

3.3 Choix technologiques et environnement de développement

3.3.1 Environnement Matériel (Hardware)

- **ESP32** : L'ESP32 est une gamme de microcontrôleurs SoC (System on a Chip), développés par Espressif, qui intègrent des fonctionnalités sans fil Wi-Fi et Bluetooth ainsi qu'un processeur double cœur [66].

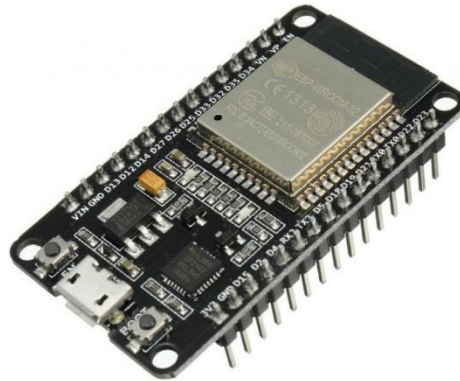


Figure 5 - Microcontrôleur ESP32

- **GPS NEO-6M** : Le module GPS NEO-6M est un récepteur GPS compatible avec la plupart des cartes à microcontrôleur. Il permet de récupérer des données relatives à la position, à la vitesse, à l'altitude et à l'heure. Il est équipé d'une batterie de secours, d'une mémoire EEPROM externe et d'un voyant LED. Son voyant commence à clignoter si une position est déterminée [67].



Figure 6 - NEO-6M GPS Module

3.3.1.1 Schéma de connexion :

Le module GPS NEO-6M communique avec l'ESP32 via une liaison série (UART). Voici le câblage retenu :

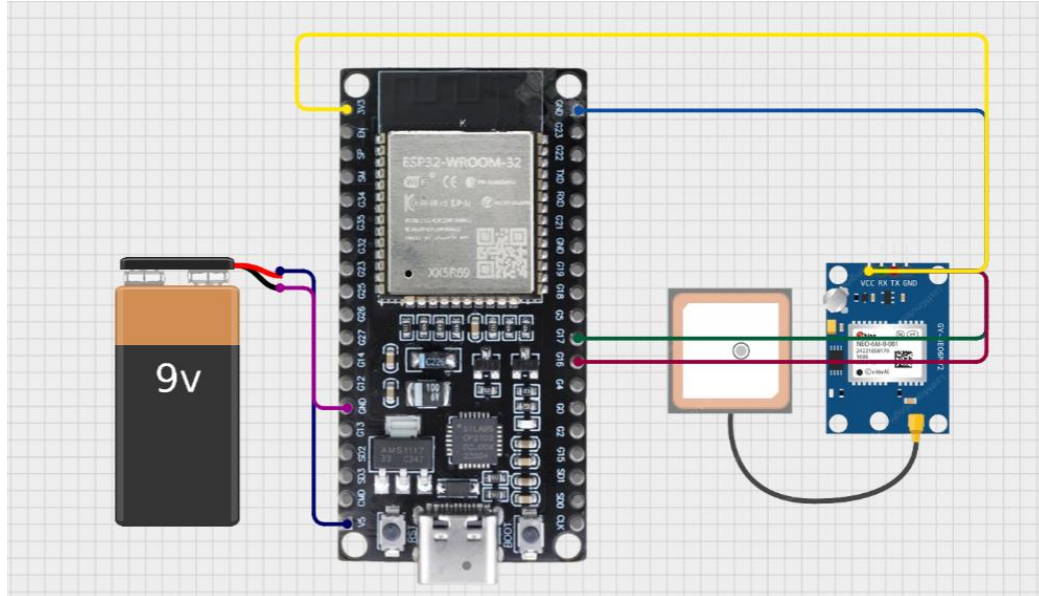


Figure 7 - schéma de câblage et d'interconnexion entre le module GPS NEO-6M et la carte ESP32

Le tableau suivant présente les connexions effectuées :

Tableau 2 - Les connexions effectuées

Module GPS NEO-6M	ESP32	Description
VCC	3.3 V	Alimentation du module GPS
GND	GND	Masse commune
TX	GPIO16 (RX2)	Transmission des données GPS
RX	GPIO17 (TX2)	Réception (configuration éventuelle)
Batterie 9 V (+)	VIN / 5V de l'ESP32	Alimentation principale de l'ESP32
Batterie 9 V (-)	GND	Masse commune du système

3.3.2 Étude de l'autonomie et gestion énergétique du système

3.3.2.1 Principe de fonctionnement par alternance de connectivité

Afin d'estimer l'autonomie du prototype alimenté par une batterie standard de 9 V (d'une capacité nominale de 500 mAh), il convient de tenir compte du comportement séquentiel des modules de communication. Le système n'active pas tous les périphériques simultanément, mais bascule d'un mode à l'autre en fonction de la disponibilité des signaux.

Mode Extérieur (Outdoor) : Le module GPS NEO-6M et le protocole Wi-Fi sont activés pour acquérir les données de géolocalisation et les transmettre directement au serveur cloud. Dans cette configuration, la consommation de courant maximale mesurée est de 145 mA.

Mode Intérieur (Indoor) : En l'absence de signal satellitaire (perte de synchronisation GPS), le module GPS passe en état de veille passive et le Wi-Fi est désactivé. Le microcontrôleur active alors le module BLE (Bluetooth Low Energy) pour assurer la diffusion des trames de proximité. La consommation globale du circuit chute à 45 mA.

3.3.2.2 Calcul théorique de l'autonomie moyenne

En considérant un cycle d'utilisation standardisé réparti équitablement entre l'environnement intérieur (**50 %**) et l'environnement extérieur (**50 %**), le courant moyen consommé ($I_{moyenne}$) par le circuit est déterminé par la moyenne arithmétique des deux états [68] :

$$I_{moyenne} = \frac{145mA + 45mA}{2} = 95mA$$

Équation 1 : La moyenne arithmétique

Mode Outdoor : le GPS et le Wi-Fi fonctionnent simultanément afin d'acquérir les coordonnées géographiques et de les transmettre vers Firebase, ce qui correspond à une consommation d'environ 145 mA.

Mode Indoor : le GPS et le Wi-Fi sont désactivés, tandis que seul le module BLE reste actif pour assurer la localisation de proximité, ce qui réduit la consommation à environ 45 mA.

En appliquant la loi de décharge nominale pour le calcul de l'autonomie temporelle (T) [68]:

$$T = \frac{\text{Capacité de la batterie (mAh)}}{\text{courant moyen (mA)}} = \frac{500mAh}{95mA} \approx 5.26h$$

Équation 2 : La loi de décharge nominale

- **Terminal de test mobile :** Le terminal de test mobile est un smartphone Samsung (modèle SM-A145R) fonctionnant sous le système d'exploitation Android. Il a été utilisé pour exécuter l'application développée sous Flutter que pour héberger l'outil de diagnostic nRF Connect, permettant le test et la vérification des communications Bluetooth Low Energy (BLE).

3.3.3 Environnement Logiciel (Software):

- **Arduino IDE** : Utilisé pour la programmation du firmware de l'ESP32 en langage C/C++. Il offre une bibliothèque de pilotes pour les capteurs et la connectivité Cloud.
- **Android Studio & Flutter SDK** : utilisés pour le développement de l'application mobile principale.
- **Firestore console**: La Firestore Console centralise les services Back-end. Cloud Firestore (NoSQL orienté documents) assure, grâce à sa flexibilité et sa scalabilité, un stockage et une lecture ultra-rapides des coordonnées géographiques envoyées par l'ESP32.
- **Application nRF Connect** : est une application mobile dédiée au test et à l'analyse des communications Bluetooth Low Energy (BLE). Dans ce projet, elle a été utilisée pour détecter la carte ESP32, mesurer la puissance du signal RSSI et visualiser les caractéristiques BLE contenant les coordonnées géographiques (latitude et longitude) diffusées par le système.

3.3.4 Bibliothèques utilisées :

Le développement du programme a été réalisé via l'environnement Arduino IDE, utilisé pour la conception et le prototypage rapide des systèmes embarqués. Pour assurer le bon fonctionnement du système de suivi proposé, plusieurs bibliothèques logicielles ont été intégrées pour la gestion du GPS, de la connectivité réseau, de la communication avec Firebase ainsi que de la technologie Bluetooth Low Energy (BLE).

3.3.4.1 Bibliothèque TinyGPS++ :

Cette bibliothèque est utilisée pour le traitement des données issues du module GPS NEO-6M. Ce dernier transmet en continu des trames de données conformes au protocole standard NMEA 0183. TinyGPS++ assure l'analyse et l'interprétation de ces trames pour extraire les informations géographiques pertinentes : la latitude et la longitude dans notre cas. Les

coordonnées obtenues sont converties en valeurs numériques exploitables par le programme pour les opérations de localisation et de transmission des données [69].

3.3.4.2 Bibliothèque Wi-Fi :

La bibliothèque WiFi intégrée à l'ESP32 permet la connexion du microcontrôleur à un réseau sans fil en mode station (STA). Cette connexion est obligatoire pour transmettre les données de localisation vers la plateforme Firebase en temps réel [70].

3.3.4.3 Bibliothèque Firebase ESP Client :

Les coordonnées GPS acquises sont automatiquement stockées dans le cloud sous forme de documents grâce à cette bibliothèque [71].

3.3.4.4 Bibliothèques de communication BLE (BLEDevice, BLEServer et BLEUtils) :

Pour l'implémentation de la technologie Bluetooth Low Energy (BLE), plusieurs bibliothèques dédiées à l'ESP32 ont été utilisées. Ces dernières permettent d'exploiter le contrôleur radio intégré pour assurer la communication sans fil à courte portée, adaptée aux environnements intérieurs [72].

- **BLEDevice** : Assure l'initialisation et la configuration du module Bluetooth intégré à l'ESP32.
- **BLEServer** : Permet de configurer l'ESP32 en tant que périphérique BLE capable de diffuser sa présence et de répondre aux demandes de connexion des équipements clients.
- **BLEUtils** : Fournit des fonctions utilitaires destinées à la création de services et de caractéristiques BLE, et à la gestion des données publicitaires (Advertising Data).

3.4 L'Architecture de Cloud Firestore (le back-end) [73]:

Nous précisons que Firestore organise les données en collections et en documents.

La collection principale (tracking): C'est le point d'entrée ou le dossier racine de l'application IoT. Elle regroupe les équipements de l'organisation qui possèdent un module de traçage.

Le document de l'équipement (object_01): À l'intérieur de la collection racine, chaque objet possède son propre document unique.

La Sous-Collection Dynamique (positions) : C'est la structure pour la sauvegarde historique. Si on stocke toutes les coordonnées dans le document, l'espace sera saturé en quelques jours. Pour résoudre ce problème, nous créons une sous-collection. Chaque fois que l'ESP32 envoie une nouvelle position (toutes les 10 secondes), Firestore génère un nouveau document à l'intérieur de cette sous-collection.

3.5 L'architecture de l'application Flutter [74]:

Nous précisons dans ce qui suit les composants de l'architecture basée sur Flutter.

Le mécanisme du StreamBuilder : Au lieu de fermer et rouvrir la connexion pour vérifier s'il y a de nouvelles positions (Polling), Flutter garde le canal ouvert. Dès qu'un document est ajouté par l'ESP32 dans Firestore, le StreamBuilder intercepte immédiatement l'événement et reconstruit automatiquement la vue de l'application en moins de 200 millisecondes.

Le contrôle du WebViewController (WebView) : Les cartes OpenStreetMap et la bibliothèque Leaflet.js fonctionnent avec les technologies du Web front-end (HTML, CSS et JavaScript). Pour les intégrer proprement dans une application mobile, Flutter utilise le package `webview_flutter`.

3.6 Tests et résultats :

3.6.1 Test d'acquisition de la position

Nous avons commencé par valider le fonctionnement matériel du module GPS NEO-6M et sa communication série avec la carte ESP32. L'objectif est de s'assurer que le module réussit à

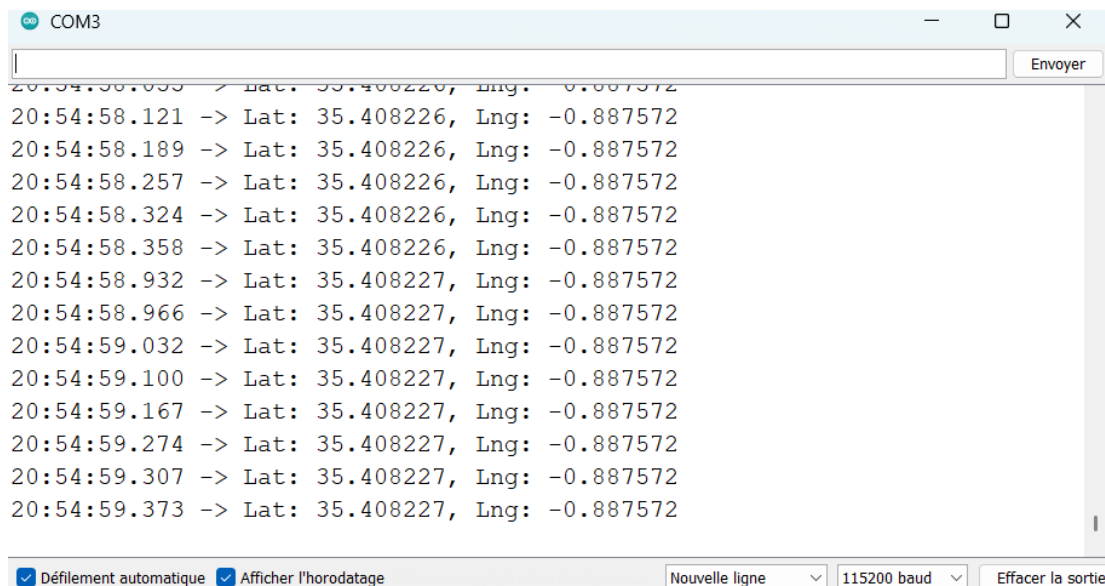
obtenir un verrouillage satellitaire (GPS Fix) et que l'ESP32 est capable d'intercepter, de filtrer et de décoder correctement ces informations.

Pour cela, nous avons développé et téléversé un programme de test initial dans l'environnement Arduino IDE. Ce script utilise la bibliothèque TinyGPS++ et initialise l'interface série matérielle UART2 sur les broches GPIO 16 (RX2) pour la réception et GPIO 17 (TX2) pour la transmission, afin d'écouter le flux continu des données du module GPS.

Quand le prototype est alimenté et placé dans un environnement extérieur, nous avons surveillé deux indicateurs qui confirment le succès de l'opération :

Indicateur matériel (Hardware) : La diode LED intégrée au module NEO-6M a commencé à clignoter de façon régulière. Ce signal confirme que le module a établi un verrouillage stable avec au moins quatre satellites, garantissant la précision des données géographiques.

Indicateur logiciel (Software): Le code pour n'afficher que les coordonnées (latitude et longitude) de manière stable et continue. Le résultat obtenu est représenté dans la figure 7:



The screenshot shows the Arduino IDE serial monitor window titled 'COM3'. It displays a continuous stream of GPS data. Each line consists of a timestamp followed by an arrow and the coordinates 'Lat' and 'Lng'. The coordinates are consistently 35.408226 for latitude and -0.887572 for longitude. The monitor has a search bar at the top right with the button 'Envoyer'. At the bottom, there are checkboxes for 'Défilement automatique' and 'Afficher l'horodatage', both of which are checked. To the right of these are dropdown menus for 'Nouvelle ligne' and '115200 baud', and a button 'Effacer la sortie'.

```
20:54:58.033 -> Lat: 35.408226, Lng: -0.887572
20:54:58.121 -> Lat: 35.408226, Lng: -0.887572
20:54:58.189 -> Lat: 35.408226, Lng: -0.887572
20:54:58.257 -> Lat: 35.408226, Lng: -0.887572
20:54:58.324 -> Lat: 35.408226, Lng: -0.887572
20:54:58.358 -> Lat: 35.408226, Lng: -0.887572
20:54:58.932 -> Lat: 35.408227, Lng: -0.887572
20:54:58.966 -> Lat: 35.408227, Lng: -0.887572
20:54:59.032 -> Lat: 35.408227, Lng: -0.887572
20:54:59.100 -> Lat: 35.408227, Lng: -0.887572
20:54:59.167 -> Lat: 35.408227, Lng: -0.887572
20:54:59.274 -> Lat: 35.408227, Lng: -0.887572
20:54:59.307 -> Lat: 35.408227, Lng: -0.887572
20:54:59.373 -> Lat: 35.408227, Lng: -0.887572
```

Figure 8 - Affichage des coordonnées GPS (latitude et longitude) sur le moniteur série de l'Arduino IDE.

3.6.2 Expérimentations en environnement intérieur

Afin de valider la communication à courte portée (en environnement intérieur) et le protocole de transfert de données, nous avons utilisé le contrôleur radio intégré de l'ESP32 pour implémenter la technologie Bluetooth Low Energy (BLE) [75].

L'objectif à cette étape est de configurer l'ESP32 en tant que serveur BLE capable de diffuser en continu les paquets de données vers un terminal mobile, sans nécessiter de connexion Internet. Pour intercepter et valider ces paquets en temps réel, nous avons utilisé l'application de diagnostic nRF Connect for Mobile, installée sur un smartphone.

Une fois le code exécuté sur l'ESP32, l'application nRF Connect a réussi à scanner et détecter le périphérique. Comme le montre la Figure 8, nous avons pu intercepter les caractéristiques BLE contenant les valeurs de la latitude et de la longitude.



Figure 9 - Capture d'écran de l'application nRF Connect validant la réception des données géographiques via BLE

En milieu intérieur (Indoor), lorsque les signaux satellitaires GPS réels sont indisponibles, le système bascule vers une solution alternative basée sur l'estimation de la distance.

Cette solution repose sur la mesure de la puissance du signal reçu, appelée RSSI (Received Signal Strength Indicator), qui traduit la qualité de la liaison sans fil entre la carte émettrice (ESP32 dans notre cas) et l'application réceptrice (smartphone client).

Le principe physique utilisé dans ce cas est le suivant: plus l'utilisateur s'éloigne physiquement du boîtier, plus les ondes se dissipent à cause des obstacles (murs, cloisons) et de la distance, ce qui rend la valeur du RSSI plus négative (exprimée en dBm).

Pour interpréter ces signaux et estimer la distance réelle en mètres, nous utilisons l'application nRF Connect, qui s'appuie sur le modèle mathématique de propagation des ondes (Log-Distance Path Loss Model) à travers l'équation suivante [76]:

$$RSSI = -(10 * n * \log_{10}(d) + A)$$

Équation 3 : Le modèle mathématique de propagation des ondes

Où :

d : représente la distance estimée en mètres.

n : désigne le coefficient d'atténuation environnemental, représentant les pertes de propagation dues aux obstacles et aux parois en milieu intérieur (murs, obstacles)

A : correspond à la puissance du signal mesurée à un mètre de distance (calibrée à environ -59 dBm selon les spécifications de l'antenne intégrée de l'ESP32).

Relation entre le RSSI et la distance estimée :

Le tableau 3 résume les résultats des mesures expérimentales obtenues lors de nos essais :

Tableau 3 - Relation entre la puissance du signal RSSI, la distance estimée et la qualité de la connexion BLE.

Puissance du Signal (RSSI en dBm)	Distance Réelle Estimée (en mètres)	État de la Connexion / Qualité du Signal
-59 dBm	≈ 0,5 à 1 mètre	Excellente (Proximité immédiate)
-66 dBm	≈ 1,5 à 2 mètres	Bonne (Distance moyenne)
-85 dBm	≈ 3 mètres	Faible (Limite critique due aux obstacles)

3.6.3 Visualisation et analyse du graphe :

Nous avons pu suivre en temps réel l'évolution de la puissance du signal. Comme montré dans la Figure 9, nous constatons qu'en se rapprochant de la carte ESP32, le RSSI remonte vers des valeurs fortes (environ -52 dBm), alors qu'un déplacement dans le sens contraire provoque une baisse immédiate de l'intensité jusqu'à -80 dBm.



Figure 10 - Évolution temporelle du signal RSSI de l'ESP32 sur l'application nRF Connect

3.6.4 Temps de réponse de la communication BLE

Afin d'évaluer les performances de la communication Bluetooth Low Energy (BLE), plusieurs essais ont été réalisés en utilisant la carte ESP32 et l'application nRF Connect. Les tests ont montré que la détection de l'ESP32 et l'affichage des informations s'effectuent en un temps très court après l'activation du module Bluetooth.

Le temps de réponse moyen observé est d'environ 200 ms, tandis que le temps maximal ne dépasse généralement pas 1 seconde. Ce qui permet d'assurer une détection quasi instantanée du dispositif en environnement intérieur et garantit donc une bonne réactivité du système lors de la localisation de proximité.

Les résultats obtenus montrent que la technologie BLE constitue une solution adaptée aux applications de suivi en intérieur grâce à sa faible consommation énergétique et à son temps de réponse réduit.

3.6.5 Validation expérimentale en environnement extérieur:

Après avoir validé la faisabilité de la communication locale via le protocole BLE, nous avons procédé au déploiement du prototype en environnement extérieur (Outdoor).

A. Programmation de la carte et débogage sur le moniteur série :

Pour réaliser cette expérimentation, nous avons développé et optimisé un script sous l'environnement Arduino IDE sur la plateforme Windows 10. Ce code source permet l'initialisation de la liaison matérielle avec le module GPS, la configuration des protocoles de connexion sans fil et l'envoi sécurisé des requêtes réseau vers les serveurs cloud de Google.

Les étapes suivantes ont été suivies pour le déploiement et la phase de diagnostic du système:

- 1) Préparation et positionnement du matériel :** Le prototype a été placé dans un espace extérieur complètement dégagé (accès direct au ciel) pour éviter la réduction ou le blocage des signaux des satellites.
- 2) Connexion et activation de la supervision :** Nous avons relié la carte ESP32 à l'ordinateur de test via un câble de données, puis nous avons ouvert le moniteur série (Serial Monitor) d'Arduino IDE pour suivre l'exécution du flux de données.
- 3) Supervision de la séquence d'initialisation :** Comme le montre la figure 10, le terminal série affiche la réussite des étapes suivantes :

- **Initialisation du module GPS (UART2) :** Configuration des broches matérielles de réception RX2 (GPIO 16) et de transmission TX2 (GPIO 17) de l'ESP32 pour communiquer avec le module GPS NEO-6M.
- **Connexion Réseau (Wi-Fi) :** Connexion automatique de la carte ESP32 à un point d'accès Wi-Fi disponible pour lui permettre d'accéder à Internet. Une fois l'authentification réussie, le moniteur série affiche l'adresse IP de la carte, confirmant que le prototype est prêt à envoyer les données au cloud.
- **Obtention de la synchronisation (GPS Fix) :** Positionnement du prototype en extérieur et attente du clignotement de la LED rouge du module GPS.
- **Lecture des coordonnées :** Nous avons ici l'affichage en temps réel sur le moniteur série de la latitude et de la longitude exactes.

```

11:56:48.625 ->
11:56:48.625 -> Initialisation de l'ESP32...
11:56:48.625 -> Série Neo-6M initialisée.
11:56:48.625 -> Connexion au Wi-Fi.....
11:56:52.207 -> Wi-Fi connecté !
11:56:52.207 -> Adresse IP de l'ESP32 : 192.168.1.125
11:56:58.611 -> GPS Fix OK -> Lat: 35.661571 | Lng: -0.627978
11:56:58.611 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:56:59.793 -> -----
11:57:08.597 -> GPS Fix OK -> Lat: 35.661569 | Lng: -0.627985
11:57:08.597 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:10.115 -> -----
11:57:18.621 -> GPS Fix OK -> Lat: 35.661568 | Lng: -0.627993
11:57:18.621 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:19.942 -> -----
11:57:28.594 -> GPS Fix OK -> Lat: 35.661565 | Lng: -0.627997
11:57:28.628 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:29.650 -> -----
11:57:38.606 -> GPS Fix OK -> Lat: 35.661566 | Lng: -0.627991
11:57:38.606 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:39.886 -> -----
11:57:48.606 -> GPS Fix OK -> Lat: 35.661568 | Lng: -0.627988
11:57:48.606 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:49.625 -> -----
11:57:58.606 -> GPS Fix OK -> Lat: 35.661571 | Lng: -0.627982
11:57:58.606 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:57:59.753 -> -----
11:58:08.599 -> GPS Fix OK -> Lat: 35.661579 | Lng: -0.627979
11:58:08.599 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:58:09.819 -> -----
11:58:18.605 -> GPS Fix OK -> Lat: 35.661591 | Lng: -0.627980
11:58:18.605 -> Envoi d'une nouvelle position à l'historique Firestore... Code HTTP : 200
11:58:19.627 -> -----

```

Figure 11 - Flux de débogage et d'acquisition des données sur le moniteur série.

B. Réception et visualisation des données en temps réel (Real-time) sur Cloud Firestore

La synchronisation des données géographiques avec la plateforme cloud s'est effectuée de manière instantanée.

À chaque période d'échantillonnage, l'ESP32 capture les dernières valeurs de positionnement fournies par le module GPS.

La figure 11 présente une capture d'écran de la console Cloud Firestore du projet. Nous y avons observé en temps réel la génération des documents de position dans la sous-collection positions à chaque déplacement physique du prototype sur le terrain, sans nécessiter un rechargement manuel de la page. Cette expérimentation valide la fiabilité et la réactivité de l'architecture IoT implémentée.

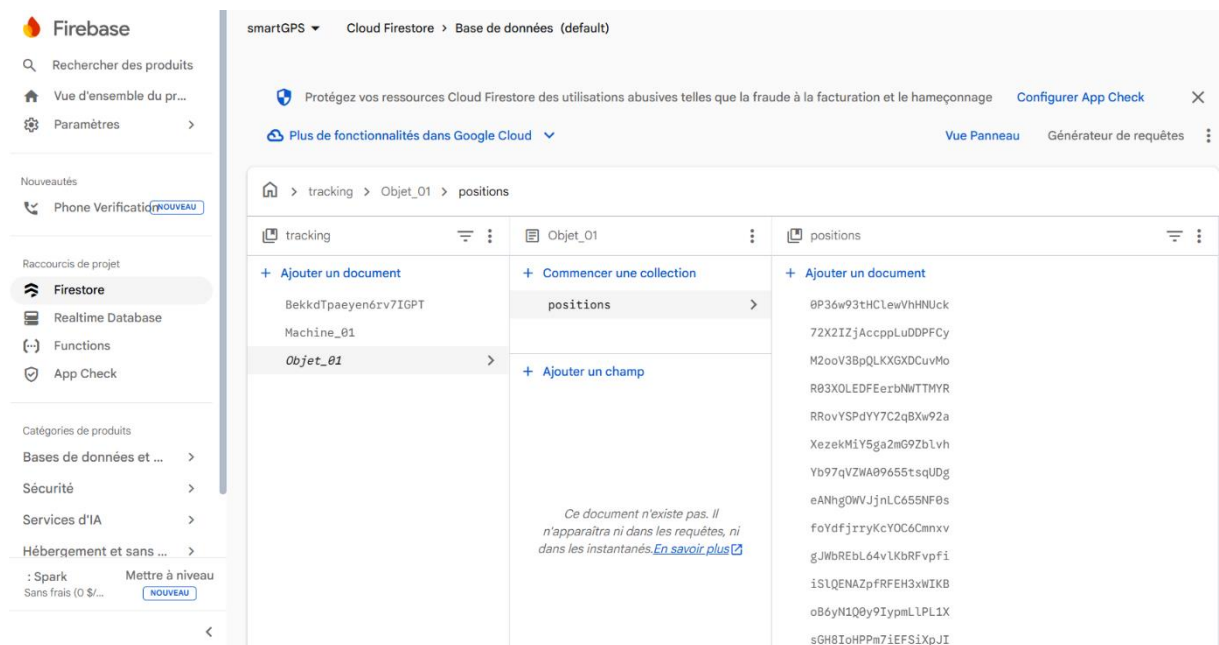


Figure 12 - Interface Cloud Firestore

Le tableau ci-dessous regroupe les cinq premières coordonnées géographiques reçues successivement par l'application pour l'équipement connecté "Objet_01", telles qu'elles sont stockées dans la base de données :

Tableau 4 - Extrait des données de localisation enregistrées en temps réel

Instance	Device Name	Latitude	Longitude	Type de Donnée
01	"Objet_01"	35.661552	0.627794-	double
02	"Objet_01"	35.661576	0.627799-	double
03	"Objet_01"	35.661580	0.627836-	double
04	"Objet_01"	35.661591	0.627764-	double
05	"Objet_01"	35.661585	0.627795-	double

3.6.6 Développement de l'application mobile avec Flutter :

Pour concevoir l'interface utilisateur et assurer le suivi cartographique en temps réel, nous avons mis en place une architecture logicielle basée sur le framework Flutter et le langage Dart.

3.6.6.1 Configuration de l'infrastructure de développement

L'installation de l'environnement de travail a nécessité le déploiement de plusieurs outils :

Le SDK Flutter et Dart : Ils intègrent les bibliothèques et les compilateurs nécessaires pour traduire le code source en instructions machine natives.

Android Studio : Cet IDE a été configuré pour piloter les outils système du SDK Android : les Command-line Tools et les Build-Tools, qui gèrent la chaîne de compilation pour le terminal physique.

3.6.7 Initialisation du projet

Afin de pouvoir utiliser le système global, il faut passer par l'étape d'initialisation de la structure du projet, qui a été effectuée en utilisant l'invite de commande (CMD) de Windows :

- **Vérification :**

La commande Flutter Doctor permet d'exécuter un outil de diagnostic qui vérifie l'intégrité du SDK Flutter, l'état de la chaîne d'outils Android (Android toolchain) et la validité des licences logicielles requises avant toute compilation.

- **Génération de l'arborescence:** Nous avons procédé à la création automatique de la structure du projet en exécutant la commande : `flutter create gps_tracker`
- **Intégration de l'écosystème Cloud et gestion des dépendances (pubspec.yaml)**

Nous avons l'intégration de l'écosystème Cloud et la gestion des dépendances (pubspec.yaml),

Après avoir configuré le fichier pubspec.yaml, l'étape suivante consiste à télécharger et installer localement ces dépendances dans le projet. Pour cela, nous avons exécuté la commande de récupération des paquets via l'invite de commande : `flutter pub get`

- **Architecture logicielle et structure du code source (Dart) :**

La construction de l'interface repose sur la composition d'un arbre de widgets. L'organisation des fichiers sources au sein du répertoire `/lib` ainsi que le point d'entrée de l'application sont structurés de la manière suivante:

A. Fichier d'initialisation et de couplage Cloud: main.dart

Le rôle de ce script est de gérer le lien entre Flutter et Firebase.

Pour cela, `WidgetsFlutterBinding.ensureInitialized()` a été utilisé pour lier les composants du système à `Firebase.initializeApp()` est appelé pour établir la connexion sécurisée avec le projet Firebase en arrière-plan, comme montré dans la figure 12 :

```

import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'company_map_screen.dart';
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

```

Figure 13 - Structure du code source du point d'entrée main.Dart et initialisation des services
Firebase

B. Module de suivi géographique: company_map_screen.dart :

Ce module gère le flux de données entrantes, exploite un widget réactif StreamBuilder connecté à Cloud Firestore pour écouter en continu la sous-collection positions. Dès qu'une modification d'état survient (lorsque le module GPS envoie une nouvelle coordonnée), le contrôleur WebViewController intercepte ces nouvelles données géographiques (latitude et longitude) et les introduit dans le script JavaScript de la carte OpenStreetMap pour tracer l'itinéraire en temps réel. Le code est montré dans la figure 13:

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:webview_flutter/webview_flutter.dart';

class CompanyMapScreen extends StatefulWidget {
  const CompanyMapScreen({Key? key}) : super(key: key);

  @override
  _CompanyMapScreenState createState() => _CompanyMapScreenState();
}

class _CompanyMapScreenState extends State<CompanyMapScreen> {
  late final WebViewController _webViewController;
  bool _isMapLoaded = false;

  Stream<QuerySnapshot> getTrackingHistory() {
    return FirebaseFirestore.instance
      .collection('tracking')
      .doc('Objet_01')
      .collection('positions')
      .snapshots();
  }
}

```

Figure 14 - Code source du module de suivi et de gestion du flux de données Firestore.

- **Configuration des scripts d'assemblage (Gradle) :**

Nous avons des modifications dans le système d'assemblage Android (Gradle) :

- **Propriétés globales (android/gradle.properties) :**

La migration vers les bibliothèques d'assistance AndroidX a été ajoutée en activant Jetifier (en ajoutant les lignes android.useAndroidX=true et android.enableJetifier=true).

- **Configuration de l'application (android/app/build.gradle) :**

Le paramètre minSdk est utilisé pour assurer la compatibilité avec le composant WebView et l'exécution de la carte OpenStreetMap.

- **Connexion matérielle, compilation et déploiement final :**

Le déploiement de la solution logicielle a été réalisé sur le terminal de test (Samsung SM-A145R) :

- **Configuration du terminal:** Le mode "Options de développement" a été déverrouillé dans les paramètres du smartphone afin d'activer l'autorisation de "Débogage USB".
- **Vérification de l'interface de liaison:** Après raccordement physique de l'appareil via un câble de données, la détection et la bonne reconnaissance de la liaison hôte ont été validées à l'aide de la commande: flutter devices.
- **Compilation et installation (Run Local) :** La phase finale d'exécution a été initiée par l'instruction flutter run

Comme le montre la Figure 14, l'application s'ouvre de manière autonome sur le terminal physique, affichant la carte dynamique OpenStreetMap synchronisée en continu avec Cloud Firestore.

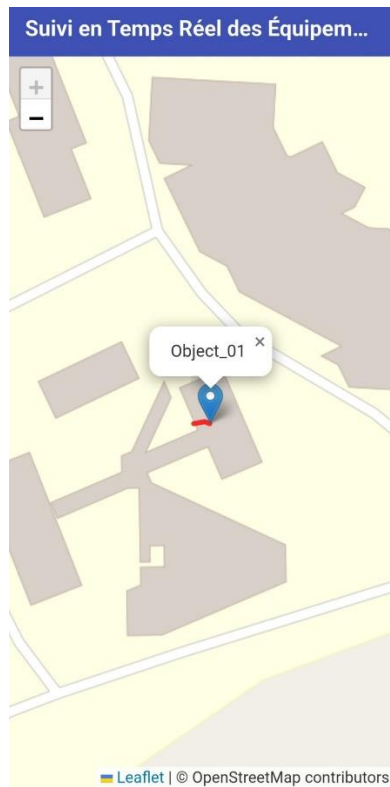


Figure 15 - Interface principale de l'application affichant la carte dynamique OpenStreetMap pour le suivi en temps réel.

Prototype final du système :

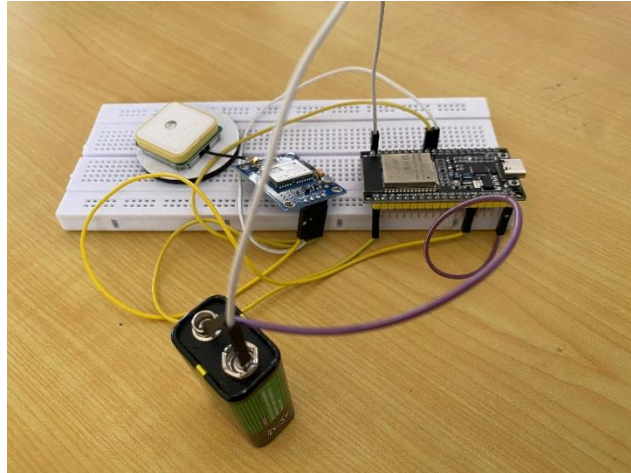


Figure 16 - Vue d'ensemble du système de suivi (Matériel)

3.6.8 Répétabilité et temps de réponse

Afin de valider les performances, la stabilité et la fiabilité de notre prototype, nous ne nous sommes pas limités à une seule expérimentation. La procédure de localisation ainsi que le transfert des données ont été reproduits au cours de dix (10) essais successifs, réalisés dans les mêmes conditions expérimentales. Les essais ont été effectués selon les deux modes de fonctionnement du système : une distance fixe de 3 mètres en environnement intérieur pour les tests utilisant le protocole Bluetooth Low Energy (BLE), et un environnement extérieur dégagé pour les essais de localisation par GPS associés à la communication Wi-Fi.

L'évaluation expérimentale s'est basée sur deux indicateurs de performance principaux :

- Le temps de réponse (latence) : il correspond au délai entre l'acquisition de l'information par le système (coordonnées GPS ou valeur RSSI) et son affichage sur l'interface de l'application mobile.

- Le taux de succès de transmission : il représente le pourcentage des données transmises, reçues et décodées correctement, sans perte d'information au cours des différentes expérimentations.

Les résultats obtenus lors des dix essais sont présentés dans le tableau suivant.

Série d'essais	Latence Outdoor (GPS / Wi-Fi vers Cloud)	Latence Indoor (BLE vers Application)	État du transfert
Essais 1 à 3	$\approx 2,4$ s	$\approx 0,8$ s	Reçu (100 %)
Essais 4 à 7	$\approx 3,1$ s	$\approx 1,2$ s	Reçu (100 %)
Essais 8 à 10	$\approx 2,8$ s	$\approx 0,9$ s	Reçu (100 %)
Moyenne calculée	$\approx 2,76$ s	$\approx 0,96$ s	Taux de succès : 100 %

Tableau 5: Résultats expérimentaux de la latence et de la répétabilité du système

3.6.9 Analyse et interprétation des résultats

L'analyse des résultats a été effectuée selon les deux modes pris en charge : en environnement intérieur et en environnement extérieur.

En mode d'environnement intérieur (BLE), le temps de réponse est faible, avec une moyenne d'environ 0,96 secondes. Cette rapidité s'explique par l'utilisation d'une liaison Bluetooth locale entre la carte ESP32 et le smartphone, sans passer par une infrastructure réseau externe. La communication directe réduit ainsi considérablement le temps de transmission des données.

En mode Outdoor (GPS / Wi-Fi), le temps de réponse moyen est d'environ 2,76 secondes. Cette durée est plus importante car différentes étapes sont nécessaires au fonctionnement du système: acquisition des coordonnées par le récepteur GPS NEO-6M, traitement des données par l'ESP32, transmission via le réseau Wi-Fi et synchronisation avec la base de données Cloud Firestore avant leur affichage dans l'application Flutter.

3.7 Conclusion :

Ce chapitre a présenté les étapes de réalisation de notre système de suivi de localisation. Les essais effectués ont permis de valider le bon fonctionnement de chaque composant: d'un côté, le suivi en environnement interne (indoor) a été effectué en utilisant le protocole BLE et validé via l'application nRF Connect. D'un autre côté, le suivi en environnement externe a été réalisé par l'acquisition des données GPS et leur affichage sur l'application mobile. Les résultats obtenus confirment la faisabilité du système proposé.

Conclusion

À travers ce projet, nous avons conçu et implémenté un système de suivi d'objets basé sur les technologies de l'Internet des objets (IoT), répondant au besoin de localisation des équipements au sein des organisations. L'objectif principal était de proposer une solution capable d'assurer le suivi des objets en temps réel en exploitant les avantages des technologies modernes.

Dans un premier temps, une étude théorique a été menée afin de comprendre les concepts fondamentaux de l'IoT, les architectures associées ainsi que les différentes technologies de positionnement et de communication utilisées dans les systèmes de suivi d'objets. Cette analyse nous a permis d'établir un état de l'art et d'identifier les solutions les plus adaptées à notre contexte et de justifier le choix des technologies retenues pour la réalisation du projet.

Ensuite, nous avons procédé à la conception de l'architecture globale du système. Le microcontrôleur ESP32 a été choisi vu ses capacités de communication Wi-Fi et Bluetooth intégrées, le module GPS NEO-6M a aussi été utilisé et a permis l'acquisition précise des coordonnées géographiques des objets suivis.

La phase d'implémentation concernait le développement du système chargé de collecter les données de localisation, puis de transmettre ces données vers la plateforme Firebase Cloud Firestore en utilisant une connexion Wi-Fi. Aussi, une application mobile développée sous Flutter a été mise en œuvre afin de permettre le suivi des positions en temps réel. L'utilisation de la technologie Bluetooth Low Energy (BLE) a également permis de réaliser des tests de communication en environnement local.

Les expérimentations réalisées ont montré que le système est capable d'acquérir, transmettre, stocker et afficher les coordonnées géographiques. Les résultats obtenus démontrent la faisabilité de la solution proposée et son aptitude à répondre aux besoins de traçabilité et de gestion des équipements dans différents environnements organisationnels. Grâce à l'intégration du cloud computing, les informations sont accessibles à distance en temps réel, ce qui permet le contrôle des objets d'une organisation.

Aussi, ce projet a permis de mobiliser et de consolider les connaissances acquises en instrumentation d'un côté, en mettant en place un système basé sur le microcontrôleur ESP32. D'un autre côté, ce projet a également permis d'apprendre à maîtriser les technologies basées sur le cloud, ce qui constitue une expérience enrichissante. Enfin, ce projet démontre concrètement l'apport de l'IoT dans la traçabilité des équipements au sein des organisations.

Perspectives

Les travaux réalisés dans ce projet peuvent être complétés par plusieurs apports, afin d'optimiser les performances du système, de renforcer son niveau de sécurité et d'enrichir son intelligence fonctionnelle.

En effet, il serait intéressant d'intégrer un système d'authentification sécurisé afin de limiter l'accès aux données de localisation uniquement aux utilisateurs autorisés.

Une autre perspective est de doter le prototype d'un module GSM pour son utilisation au sein des organisations multisites où leurs infrastructures sont géographiquement éloignées.

Du point de vue de l'autonomie du système, il faudra proposer des solutions qui pourront assister ce dernier dans le cas d'une coupure d'alimentation tout en garantissant la mobilité de ce dernier.

Aussi, l'utilisation des techniques de l'intelligence artificielle représente une voie d'amélioration intéressante pour l'exploitation des données collectées. Leur utilisation pourrait servir à identifier des comportements anormaux, prédire les trajectoires futures des objets suivis, optimiser l'utilisation des ressources à partir des données historiques disponibles.

Ces solutions pourraient permettre de rendre le système plus autonome et plus performant. Des algorithmes pourraient être développés afin de générer des alertes automatiques, analyser les habitudes de déplacement et détecter des situations inhabituelles.

Bibliographie

- [1] M S Rudramurthy and Keerthan Kumar T G. Grenze. Real-Time Object Localization and Tracking using IoT Integrated GPS Technology. International Journal of Engineering and Technology, June Issue. 2025.
- [2] A. Choudhary, Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things. 19 December 2024.
- [3] Decoding the Internet of Things, Carnegie Mellon University Computer Science Department 2023. Apr. 2023. <https://www.csd.cmu.edu/news/decoding-the-internet-of-things>.
- [4] MIT Auto-ID Lab. Mit Auto-id Laboratory. <https://autoid.mit.edu/> (consulté le 22/02/2026)
- [5] Farah Kandah,, Navigating IoT Security: Architectures, Emerging Threats, and Adaptive Countermeasures Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA ,4 June 2025.
- [6] R. Abdmeziem, D. Tandjaoui,” Internet of Things: Concept, Building blocks, Applications and Challenges, Computers and Society, Cornell University”in International Journal of Computer Applications, Volume 113, 2015.
- [7] L. Atzori, A.Iera, G. Morabito, ”The Internet of Things: A survey,” in Computer Networks - Science Direct Volume 54, Issue 15, 28 October 2010, Pages 2787-2805.
- [8] R. Choudhary, A. Vats, G. Kaur, I. Chaudhary, S. Sharma. Internet of Things: Unabridged Solution International Journal of Computer Applications (0975 – 8887) Volume 182 – No.2, July 2018.

- [9] P. Fuhrer, D. Guinard, "Building a Smart Hospital using RFID technologies," in 1st European Conference on Health (ECEH06), Fribourg, Switzerland, October 12 -13, 2006.
- [10] H. Kagermann, J. Helbig, A. Hellinger, et al., "Recommendations for Implementing the Strategic Initiative Industry 4.0: Securing the Future of German Manufacturing Industry," 2013, Final Report of the Industrie 4.0 Working Group, acatech—National Academy of Science and Engineering.
- [11] Industrial Internet Consortium, "What Is the Industrial Internet?" <https://www.iiconsortium.org/pdf/IIC-Overview.pdf> (consulté le 04/03/2026)
- [12] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems," *Manufacturing Letters* 3 (2015): 18–23.
- [13] K. E. Nolan, W. Guibene, and M. Y. Kelly, "An Evaluation of Low Power Wide Area Network Technologies for the Internet of Things," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, (Paphos, Cyprus: IEEE, 2016), 439–444.
- [14] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human- Machine- Interaction in the Industry 4.0 Era," in *12th IEEE International Conference on Industrial Informatics (INDIN)*, (Porto Alegre, Brazil: IEEE, 2014), 289–294.
- [15] A. Mosenia and N. K. Jha, "A Comprehensive Study of Security of Internet-of- Things," *IEEE Transactions on Emerging Topics in Computing* 5, no. 4 (2017): 586–602.
- [16] A. Corallo, A. M. Crespino, V. D. Vecchio, M. Lazoi, and M. Marra, "Understanding and Defining Dark Data for the Manufacturing Industry," *IEEE Transactions on Engineering Management* 70, no. 2 (2023): 700–712.

- [17] S. A. Boyer, SCADA: Supervisory Control and Data Acquisition (International Society of Automation, Research Triangle Park, NC, USA, 4th edition, 2009).
- [18] M. ALABADI, A. HABBAL, XIAN WEI. Industrial Internet of Things: Requirements, Architecture, Challenges, and Futur. Digital Object Identifier 10.1109/ACCESS.2022.3185049. IEEE Access. 2022.
- [19] Jay Lee, Behrad Bagheri, Hung-An Kao, Cyber-Physical Systems architecture for Industry, Manufacturing Letters 3 (2015) 18–23.
- [20] C. Shang and F. You, “Data Analytics and Machine Learning for Smart Process Manufacturing: Recent Advances and Perspectives in the Big Data Era,” Engineering 5, no. 6 (2019):1010–1016.
- [21] M. Khan, X. Wu, X. Xu, and W. Dou, “Big Data Challenges and Opportunities in the Hype of Industry 4.0,” in IEEE International Conference on Communications (ICC), (Paris, France: IEEE, 2017), 1–6.
- [22] Shaila Afrin, Sabiha Jannat Rafa, Maliha Kabir, Tasfia Farah, Md. Sakib Bin Alam, Aiman Lameesa, Industrial Internet of Things: Implementations, challenges, and potential solutions across various industries Computers in Industry 170 (2025) 1043172
- [23] Tunda Olembe Djamba. Notes de cours sur l’Internet industriel des objets (IIot), le smart manufacturing et l’industrie 4.0. Université de Lubumbashi, école supérieure des ingénieurs industriels, département de génie informatique et intelligence artificielle. Janvier 2024.
- [24] Evizal R, Rahim SKA. RFID vehicle plate number (e-plate) for tracking and management system. In: Proceedings of the International Conference on Parallel and Distributed Systems ICPADS, IEEE Computer Society, 2013. pp. 611–616

- [25] Derawi M, Dalveren Y, Cheikh FA. Internet-of-things-based smart transportation systems for safer roads. In: IEEE World Forum on Internet of Things, WF-IoT 2020 Symposium Proceedings. Institute of Electrical and Electronics Engineers Inc., 2020.
- [26] Singh D, Kew HP, Tiwary US, Lee HJ, Chung WY. Global patient monitoring system using IP-enabled ubiquitous sensor network. 2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009;1:524– 528
- [27] Saleh, I. « Les enjeux et les défis de l’Internet des Objets (IdO) », revue Internet des objets 1.DOI :10.21494/ISTE.OP.2017.0133, 2017.
- [28] “Internet of Things: An overview” by Internet Society
<https://www.internetsociety.org>
- [29] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- [30] Abdulla Nasir Chowdhury, Abdul Majid Nihad, Shah Masud Parvej Mamun, Aftar Ahmad Sami, (2025), Prototype and Simulation of a Real-Time GPS and PTS Architecture, International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) 31 July – 2 August 2025, Rangpur, Bangladesh
- [31] Jurusan Teknik Elektro dan Informatika – Politeknik Manufaktur Negeri Bangka Belitung. Implementasi Metode Reverse Geocoding pada Aplikasi Tracking Posisi, March 2023, Emitor Jurnal Teknik Elektro 1(1):44-49, DOI:10.23917/emitor.v1i1.21464. License. CC BY 4.0.
- [32] Abdullahi, I. & Maliki, D. & Dauda, I. & Dauda, U. & Kigbu, G.. (2025). Development of a Smart Internet of Things Based Motorcycle Theft Prevention System.

- ARID ZONE JOURNAL OF ENGINEERING, TECHNOLOGY AND ENVIRONMENT. 21. 847-856. 10.63958/azojete/2025/21/03/015.
- [33] Azad Affandi Ghadaffi Azhar, Jong Siat Ling. Development of Smart Parking Security Application Using LoRa GPS Based IoT Geo-fencing. EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING e-ISSN: 2756-8458. Vol. 6 No. 2 (2025) 199-206 <https://publisher.uthm.edu.my/periodicals/index.php/eeee> This is an open access article under the CC BY-NC-SA 4.0 license.
- [34] Chang, Yeong-Hwa & Wu, Feng-Chou & Lin, Hung-Wei. (2025). Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors. 25. 1656. 10.3390/s25061656.
- [35] P. Kriz, F. Maly, and T. Kozel. Improving Indoor Localization Using Bluetooth Low Energy Beacons. Hindawi Publishing Corporation Mobile Information Systems Volume 2016, Article ID 2083094, 11 pages <http://dx.doi.org/10.1155/2016/2083094>.
- [36] Chen, F., et al. (2022). Visual object tracking: A survey of challenges and methods. Computer Vision and Image Understanding.
- [37] Kadam, P., Fang, G., & Zou, J. J. (2024). Object tracking using computer vision: A review. Computers, 13(6).
- [38] Qiu, T.; Chen, N.; Li, K.; Atiquzzaman, M.; Zhao, W. How Can Heterogeneous Internet of Things Build Our Future: A Survey. IEEE Commun. Surv. Tutorials 2018, 20, 2011–2027.
- [39] Mashal, I.; Alsaryrah, O.; Chung, T.Y.; Yang, C.Z.; Kuo, W.H.; Agrawal, D.P. Choices for interaction with things on Internet and underlying issues. Ad Hoc Netw. 2015, 28, 68–90.
- [40] Mansour, M.; Gamal, A.; Ahmed, A.I.; Said, L.A.; Elbaz, A.; Herencsar, N.; Soltan, Internet of Things: A Comprehensive Overview on Protocols, Architectures,

- Technologies, Simulation Tools, and Future Directions A. Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions. *Energies* 2023, 16, 3465.
- [41] Colakovic, A.; Hadžialic, M. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Comput. Networks* 2018, 144, 17–39.
- [42] Farooq, M.U.; Waseem, M.; Mazhar, S.; Khairi, A.; Kamal, T. A review on internet of things (IoT). *Int. J. Comput. Appl.* 2015, 113, 1–7.
- [43] Vashi, S.; Ram, J.; Modi, J.; Verma, S.; Prakash, C. Internet of Things (IoT): A vision, architectural elements, and security issues. In *Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 10–11 February 2017; pp. 492–496.
- [44] Gupta, S.; Gupta, A.; Shankar, G. Cloud Computing: Services, Deployment Models and Security Challenges. In *Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 7–9 October 2021; pp. 414–418.
- [45] Swamy s, Narasimha. (2020). An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access*. VOLUME 8. 188082-188134. 10.1109/ACCESS.2020.3029847.
- [46] S. Lee and J. Song, "Use of coded infrared light as artificial landmarks for mobile robot localization," *IEEE/RSJ Int. Conf. on Intell. Robots & Syst.*, pp. 1731-1736, 2007, doi: 10.1109/IROS.2007.4399600. 2007.
- [47] R. Ali, R. Liu, Y. He, A. Nayyar, and B. Qureshi, "Systematic Review of Dynamic Multi-Object Identification and Localization: Techniques and Technologies," *IEEE Access*, vol. 9, pp. 3474-3493, 2021.

- [48] A. Yassin et al., "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," in *IEEE Commun. Surv. & Tutor.*, vol. 19, no. 2, pp. 1327-1346, Secondquarter 2017, doi: 10.1109/COMST.2016.2632427.
- [49] A. Farhat and K. C. Cheok, "Improving adaptive network fuzzy inference system with Levenberg-Marquardt algorithm," 2017 Annual IEEE Int. Sys. Conf. (SysCon), 2017, pp. 1-6, doi: 10.1109/SYSCON.2017.7934787.
- [50] J. Ryu, M. Irfan and A. Reyaz, "A Review on Sensor Network Issues and Robotics", *J. of Sensors*, vol. 2015, pp. 1-14, 2015. doi: 10.1155/2015/140217.
- [51] D. Ni, "Traffic Sensing Technologies," *Traffic Flow Theory.*, 3-17, 2016.
- [52] Hong-Shik Kim and Jong-Suk Choi, "Advanced indoor localization using sensor and digital compass," 2008 Int. Conf. on Cont., Auto. & Syst., pp. 223-226, 2008, doi: 10.1109/ICCAS.2008.4694553.
- [53] S. Holm, "Ultrasound Positioning Based On Time-Of-Flight And Signal Strength," *Int. Conf. on Indoor Posit. & Indoor Navig. (IPIN).*, 2012.
- [54] A. Alarifi et al., "Ultra-Wideband Indoor Positioning Technologies: Analysis and Recent Advances", *Sensors*, vol. 16, no. 5, p. 707, 2016. doi: 10.3390/s16050707.
- [55] M. A. Al-Ammar et al., "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms," 2014 Int. Conf. on Cyberworlds, pp. 245-252, 2014, doi: 10.1109/CW.2014.41.
- [56] B. Lashkari, J. Rezazadeh, R. Farahbakhsh and K. Sandrasegaran, "Crowdsourcing and Sensing for Indoor Localization in IoT: A Review," in *IEEE Sensors J.*, vol. 19, no. 7, pp. 2408-2434, 1 April, 2019, doi: 10.1109/JSEN.2018.2880180.

- [57] D. Dardari, P. Closas and P. M. Djurić, "Indoor Tracking: Theory, Methods, and Technologies," in *IEEE Trans. on Vehic. Tech.*, vol. 64, no. 4, pp. 1263-1278, April 2015, doi: 10.1109/TVT.2015.2403868.
- [58] J. Whitteker, Y. Le Helloco and B. Breton, "Evaluation of the Field on a Uniform Array of Knife Edges Using Edge Reflection," *IEEE Trans. on Ants. & Propag.*, vol. 55, no. 3, pp. 997-999, March 2007, doi: 10.1109/TAP.2007.891870.
- [59] S. K. Gharghan, R. Nordin, A. M. Jawad, H. M. Jawad and M. Ismail, "Adaptive Neural Fuzzy Inference System for Accurate Localization of Wireless Sensor Network in Outdoor and Indoor Cycling Applications," in *IEEE Access*, vol. 6, pp. 38475-38489, 2018, doi: 10.1109/ACCESS.2018.2853996.
- [60] S. Mahdi, S. Gharghan and M. Hasan, "FPGA-Based neural network for accurate distance estimation of elderly falls using WSN in an indoor environment", *Measurement*, vol. 167, p. 108276, 2021. doi: 10.1016/j.measurement.2020.108276.
- [61] S. Gharghan, R. Nordin, A. Jawad, H. Jawad and M. Ismail, "Adaptive Neural Fuzzy Inference System for Accurate Localization of Wireless Sensor Network in Outdoor and Indoor Cycling Applications", *IEEE Access*, vol. 6, pp. 38475-38489, 2018. doi: 10.1109/access.2018.2853996.
- [62] V. Truong, A. Nayyar and S. Ahmad Lone, "System Performance of Wireless Sensor Network Using LoRa–Zigbee Hybrid Communication", *Comp., Mater. & Continua*, vol. 68, no. 2, pp. 1615-1635, 2021. doi: 10.32604/cmc.2021.016922.
- [63] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha and W. Hu, "Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings," in *IEEE Internet of Things J.*, vol. 7, no. 1, pp. 298- 310, Jan. 2020, doi: 10.1109/JIOT.2019.2946900.

- [64] F. Febriyandi, A. S. Arifin and M. I. Nashiruddin, "Sigfox Based Network Planning Analysis for Public Internet of Things Services in Metropolitan Area," 2020 IEEE Int. Conf. on Industry 4.0, Artifi. Intell., & Comm. Tech. (IAICT), pp. 21-27, 2020, doi: 10.1109/IAICT50021.2020.9172012.
- [65] Fujun Qiu, Ashwini Kumar, Jiang Hu, Poorva Sharma, Yu Bing Tang, Yang Xu Xiang, and Jie Hong. A Review on Integrating IoT, IIoT, and Industry 4.0: A Pathway to Smart Manufacturing and Digital Transformation., Wiley. IET Information Security. Volume 2025, Article ID 9275962, 16 pages.
- [66] Espressif Systems, "ESP32 Series Datasheet," v4.3, Espressif Systems, 2023. [En ligne]. Disponible sur : https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (consulté le 07/06/2026)
- [67] u-blox, "NEO-6: u-blox 6 GPS Modules Data Sheet," u-blox Holding AG, Doc. No. GPS.G6-HW-09005, 2011. https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf (consulté le 07/06/2026)
- [68] T. Linden et T. B. Reddy, "Handbook of Batteries", 4ème édition, McGraw-Hill, New York, 2011.
- [69] M. Hart, "TinyGPS++: A full-featured NMEA parsing library for Arduino," Arduiniana, 2013. [En ligne]. Disponible sur : <https://github.com/mikalhart/TinyGPSPlus> (consulté le 07/06/2026)
- [70] Espressif Systems, "WiFi Library for ESP32," Arduino-ESP32 Core Repository, v2.0.x, 2021. [En ligne]. Disponible sur : <https://github.com/espressif/arduino-esp32/tree/master/libraries/WiFi> (consulté le 07/06/2026)
- [71] Mobizt (Suwattana), "Firebase ESP Client: Firebase SDK for ARM/AVR/ESP8266/ESP32," v4.x.x, 2023. [En ligne]. Disponible sur : <https://github.com/mobizt/Firebase-ESP-Client> (consulté le 07/06/2026)
- [72] N. Kolban and Espressif Systems, "ESP32 BLE Arduino Library (BLEDevice, BLEServer, BLEUtils)," Arduino-ESP32 Core Package, 2018. [En ligne]. Disponible

- sur : <https://github.com/espressif/arduino-esp32/tree/master/libraries/BLE> (consulté le 07/06/2026)
- [73] Google, "Firebase Documentation: Cloud Firestore Data Model," Firebase Google, 2024. [En ligne]. Disponible sur : <https://firebase.google.com/docs/firestore/data-model> (consulté le 07/06/2026)
- [74] Google, "Flutter Documentation: Architectural overview," Flutter dev, 2024. [En ligne]. Disponible sur : <https://docs.flutter.dev/resources/architectural-overview> (consulté le 07/06/2026)
- [75] Bluetooth Special Interest Group, "Core Specification Version 5.0 - Bluetooth Low Energy," Bluetooth SIG, 2016. Core Specification | Bluetooth® Technology Website. <https://www.bluetooth.com/specifications/specs/core-specification-5-0/> (consulté le 07/06/2026)
- [76] System Bluetooth Components TBluetooth LERssi To Distance Method, DocWiki Embarcadero – RAD Studio Athens, 2025. Available at: <https://docwiki.embarcadero.com/Libraries/Athens/en/System.Bluetooth.Components.TBluetoothLE.RssiToDistance> (consulté le 07/06/2026)