MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
Université Libre de Tunis
****

Projet de Fin d'Etudes

En vue de l'obtention du Diplôme National d'Ingénieur en génie électrique et informatique industriel

Elaboré par :

Khalil MEDFAI

Développement d'un système de géolocalisation d'objets connectés sur le réseau LoRaWAN

Réalisé au sein de
TreeTronix

Encadré par

Encadrant(s) universitaire(s)	**Encadrant(s) professionnel(s)**
Moez FERCHICHI	Bassem MECHERQUI

Année universitaire
2022 - 2023

Dédicaces

Je tiens à dédier ce rapport à ma famille, qui a été une source constante de soutien, d'encouragement et de motivation tout au long de ce processus. Votre amour inconditionnel, votre compréhension et votre patience ont été essentiels pour moi, et je suis extrêmement reconnaissant d'avoir à vos côtés.

À mes parents **Moncef** et **Fadhila**, vous avez toujours cru en moi et avez nourri mes ambitions. Votre soutien indéfectible m'a permis de persévérer face aux défis et de réaliser mes objectifs. Votre sagesse et vos conseils précieux m'ont guidé tout au long de ce parcours, et je vous suis infiniment reconnaissant.

Enfin, je tiens à remercier **Zouhour** du fond du cœur pour son soutien moral inestimable. Dans les moments où j'ai eu besoin de réconfort et d'encouragement, tu as été là pour moi, prêt à m'écouter et à me soutenir. Ta présence bienveillante a été un véritable soutien dans les moments de doute et d'incertitude. Je suis reconnaissant d'avoir quelqu'un comme toi à mes côtés, et je te suis profondément reconnaissant pour tout ce que tu as fait.

Je tiens à exprimer ma profonde gratitude envers tous les stagiaires qui sont devenus de chers amis pendant mon séjour commun dans ce stage. Grâce à cette expérience partagée, j'ai eu le privilège de faire leur connaissance

Remerciements

Avant de procéder à la présentation de ce rapport, j'adresse mes remerciements à tout le cadre administratif et éducatif de **L'ULT** et particulièrement à l'ensemble des enseignants, pour leur enseignement etleur formation qui m'a permis d'aborder les problématiques de manière concrète et spécialement **Mr. Aymen Slim**.

Mes vifs remerciements à **Mr. Moez Ferchichi** mon encadrant académique pour son soutien et ses conseils tout au long de la réalisation de mon projet de fin d'études.Sa disponibilité, son expertise et son dévouement ont été inestimables pour moi. Son encadrement rigoureux et exigeant m'a permis de mener à bien ce travail dans les meilleures conditions possibles.

Au terme de ce projet, je tiens à remercier vivement mon encadrant d'entreprise,

Mr. Bassem Mechergui pour sa confiance et son soutien tout au long de ce projet. Son expertise, sa disponibilité et son écoute ont été des atouts majeurs pourla réussite de ce projet. Son encadrement professionnel m'a permis d'acquérir des compétences techniques et humaines très importantes pour ma future carrière.

Pour finir, j'aimerais également exprimer ma gratitude pour l'honneur que les membres du jury me font en prendre le temps de lire et d'évaluer ce travail.

Table des matières

Dédicaces ..... *i*
Remerciements ..... *ii*
Table des matières..... *iii*
Liste des figures..... *vi*
Liste des tableaux..... *viii*
Liste des abréviations ..... *ix*
Introduction générale..... 1

CHAPITRE I. Cadre générale du projet 2
I.1 Introduction 2
I.2 Présentation de l'organisme d'accueille 2
I.2.1 Présentation de la société (3s) 2
I.2.2 Les filiales du groupe 3S..... 3
I.2.3 Présentation de la société TreeTronix 3
I.3 Présentation du projet 4
I.4 Problématique..... 5
I.5 Etude de l'existant 5
I.5.1 Exemple des solutions de géolocalisation 5
I.6 La solution proposée 8
I.7 Analyse fonctionnelle 9
I.7.1 Diagramme FAST..... 9
I.7.2 Analyse S.A.D.T..... 9
I.8 Spécification des besoins 11
I.8.1 Spécification des besoins fonctionnels 11
I.8.2 Spécification des besoins non fonctionnels 11
I.9 Conclusion 12

CHAPITRE II. GENERALITES SUR L'IOT ET LORAWAN..... 13
II.1 Introduction 13
II.2 L'internet des objets 13
II.2.1 Définition 13
II.2.2 Un Objet connecté 14
II.2.3 L'architecteur IOT 14
II.2.4 Les protocoles de communication pour L'IOT 15
II.2.4.1 Les principaux protocoles de communication pour des objet (IoT) 16
II.2.4.2 Etude comparative des protocoles IOT..... 17
II.3 LoRa..... 18
II.4 LoRaWAN 19

II.5 Architecture du réseaux LoRaWAN 21
II.5.1 Les devices (dispositifs) LoRaWAN 21
II.5.2 Les Gateway (Passerelles) LoRaWAN 21
II.5.3 Le Network serveur (serveur web) 22
II.5.4 L'Application Serveur (serveur d'application) 22
II.5.5 Serveur LoRaWAN 22
II.5.6 La Plateforme IOT 22
II.6 Les Classes du réseau LoRaWAN 23
II.6.1 Classe A 23
II.6.2 Classe B 23
II.6.3 Classe C 24
II.7 Activation d'un End-device 25
II.7.1 La méthode d'activation ABP 26
II.7.2 La méthode d'activation OTAA 27
II.8 Plateforme utilisée au niveau du serveur LoRaWAN 28
II.8.1 Exporter nos données du srveur LoRaWAN 29
II.8.2 Enregistrement du device 30
II.8.3 Créer un profile de routage Cayenne 30
II.8.4 Ajouter un End device à Cayenne 31
II.9 Conclusion 32

CHAPITRE III. TESTS ET REALISATION D'UN PROTOTYPES DE GEOLOCALISATION 33
III.1 Introduction 33
III.2 Matériels nécessaires 33
III.2.1 Choix de la carte de développement 33
III.2.2 Carte de développement STM32 B-L072Z-LRWAN1 35
III.2.3 Shield Dragino LoRa/GPS 37
III.2.4 Un analyseur Logique 39
III.3 Logiciels utilisés 40
III.3.1 Choix de l'environnement de développement 40
III.3.2 STM32CubeMX 41
III.3.3 STMCubeIDE 41
III.3.4 PuTTY 42
III.4 Installation de la bibliothèque I-CUBE-LRWAN 42
III.5 Configuration de la bibliothèque LoRaWAN 42
III.6 Transmission d'un simple message vers le serveur LoRaWAN 45
III.7 Visualisation de la sortie sur le serveur LoRaWAN 48
III.8 Programmation du module de géolocalisation 49
III.9 Conclusion 53

CHAPITRE IV. CONCEPTION D'UNE CARTE ELECTRONIQUE DE GEOLOCALISATION 54
IV.1 Introduction 54
IV.2 Logiciel de conception 54
IV.3 Conception du schéma électrique 55

IV.3.1 Microcontrôleur STM32L072CZ 55
IV.3.2 Module LoRa RFM95 56
IV.3.3 Le module RT8059 57
IV.3.4 Le module CN3052A 58
IV.3.5 Connecteur USB micro B 58
IV.3.6 Module GPS L70 59
IV.4 Routage du schéma électrique 60
IV.5 Phase de test 61
IV.6 Conclusion 63
Conclusion générale 64
Bibliographie 65
Annexe 1 Disposition supérieur de B-L072Z-LRWAN1 66
Annexe 2 Disposition supérieure deB-L072Z-LRWAN1 67
Annexe 3 Caractéristiques de STM32L072CZ 68
Annexe 4 Les pins de STM32L072CZ 69
Annexe 5 Les protocoles de communication pour les systèmes embarqués 70
Annexe 6 Schématique de la carte 1/2 71
Annexe 7 Schématique de la carte 2/2 72
Annexe 8 Routage de la carte 73
Annexe 9 Le schéma d'implantation de la carte 74

Liste des figures

Figure 1 – Logo de l'entreprise 3S 2
Figure 2 – Logo Treetronix 3
Figure 3 – Secteurs d'activités de Treetronix 4
Figure 4 – Relais GPS Tracker De Suivi De Voiture 6
Figure 5 – AirTag d'Apple 7
Figure 6 – Architecture de la solution proposée 8
Figure 7 – Le Diagramme FAST 9
Figure 8 – l'IoT 14
Figure 9- Etude comparative des protocoles IoT 17
Figure 10 - les principaux avantages communs 17
Figure 11 – LoRa 19
Figure 12 – LoRaWAN 19
Figure 13 – Structure globale d'un réseau LoRaWAN 21
Figure 14 – Fenêtres de réception RX1 et RX2 pour un dispositif de classe A 23
Figure 15 – Fenêtres de réception pour un dispositif de classe B 24
Figure 16 – Fenêtres de réception pour un dispositif de classe C 25
Figure 17 – Consommation d'énergie et capacités en downlink des dispositifs LoRaWAN 25
Figure 18 – DevAddr, NwkSkey et AppSkey en ABP 26
Figure 19 – Processus d'activation OTAA 27
Figure 20 - Le logo de la plateforme ThingPark 29
Figure 21 – Network server LoRaWAN du groupe 3s 29
Figure 22 – Enregistrement d'un nouveau device 30
Figure 23 – Créer un profil de routage Cayenne 31
Figure 24 – Ajouter un End Device à Cayenne 31
Figure 25 – Carte de développement STM32 B-L072Z-LRWAN1 35
Figure 26 – Fonctionnalités de l' STM32 B-L072Z-LRWAN1 36
Figure 27 – Le module Shield Dragino LoRa/GPS 38
Figure 28 – Exemple d'une instance NMEA 39
Figure 29 – Analyseur logique Saleae 39
Figure 30 – Logo CubeMX 41
Figure 31 – Logo STM32 CubeIDE 41

Figure 32 – Logo PuTTy 42
Figure 33 – Package d'extension I-CUBE-LRWAN 42
Figure 34 – Configuration d'un dispositif LoRa 43
Figure 35 – Configuration des paramètres OTAA 44
Figure 36 – Organigramme de l'envoie des données 46
Figure 37 – Fonction permettant d'envoyer une trame LoRaWAN 46
Figure 38 – Configuration du PuTTY 47
Figure 39– Acceptation de la jointure OTAA et l'envoi d'une donnée au serveur LoRaWAN ... 47
Figure 40 – Message transmis au serveur 48
Figure 41 – Conversion du message 48
Figure 43 - Câblage du module GPS 49
Figure 44 – Example d'une séquence GGA 50
Figure 45– L'organigramme d'extraction des coordonnées GPS 50
Figure 46 – La fonction HAL_UART_Receive_IT 51
Figure 47 – La fonction HAL_UART_Receive_IT 51
Figure 48 – Visualisation des données GPS 52
Figure 49– Visualisation des données GPS dans le terminal PuTTY 52
Figure 50– Affichage de la localisation dans Google Maps 53
Figure 51 – Logo Eagle 54
Figure 52 – Schéma de câblage du microcontrôleur STM32L072CZ 56
Figure 53 – Schéma de câblage du RFM95 57
Figure 54 – Schéma de câblage du RT8059 57
Figure 55– Schéma de câblage du CN3052A 58
Figure 56– Schéma de câblage du Connecteur USB micro B 58
Figure 57 – Schéma de câblage du odule GPS L70 59
Figure 58 – Routage de la carte avec un plan de masse 60
Figure 59– Routage de la carte avec un plan de masse 61
Figure 60– La carte électronique finale après le soudage des composants 62
Figure 60 – La carte électronique finale après le soudage des composants 62

Liste des tableaux

Tableau 1 – étude comparative entre les protocoles Sigfox et LoRa WAN.....18

Tableau 2 – tableau comparative entre l'ABP ET l'OTAA 28

Tableau 3 – Tableau Comparatif des cartes de développement Arduino ESP32, Raspberry Pi et STM32 33

Tableau 4 – Caractéristiques du Shield Dragino LoRa/GPS.....38

Tableau 5 – Tableau Comparatif des environnements de développement pour STM32 40

Tableau 6 – Tableau de définition des clés d'activation 44

Tableau 7 – Configuration des paramètres du mode d'activation 45

Tableau 8 – Broches de connexion entre le GPS L70 la carte STM32.....49

Tableau 9 – Tableau récapitulatif des avantages et des inconvénients de l'UART 71

Liste des abréviations

3S : Standard Sharing Software
IoT : Internet of Things
LoRa : Long Range
GPS: Global Positioning System
GSM: Global System for Mobile communication
SMS : Short Message Service
LoRaWAN : Long Range Wide Area Network
SADT: Structured Analysis And Design Technique
FAST: Function Analysis System Technique
LPWAN: Low Power Wide Area Network
M2M: Machine to Machine
DevAddr: Device address
NwkSKey: Network Session Key
AppSKey: Application Session Key
DevEUI: Device Extended Unique Identifier
AppKey: Application Key
AppEUI: Application Extended Unique Identifier
ABP: Activation By Personalization
OTAA: Over The Air Activation
RISC: Reduced Instruction Set Computer
UART: Universal Asynchronous Receiver/Transmitter
LPUART: Low Power UART
USB: Universal Serial Bus
ADC : Analogue Digital Converter

Introduction générale

Depuis des milliers d'années, l'homme à toujours chercher de se localiser et développer des solutions pour atteindre cet objectif. L'évolution de l'homme a déclenché le concept de la géolocalisation qui est devenue l'un des enjeux majeurs des recherches et de développement des entreprises.

En effet, déterminer la localisation d'un équipement ou d'un objet, suivre le déplacement d'un véhicule ou identifier les coordonnées d'un outil permet de réaliser le partage des ressources dans les différents domaines comme le transport, la santé, le domaine militaire et d'autres domaines. C'est en fait, une méthode qui permet de localiser un objet ou une personne sur une carte grâce à ses coordonnés géographiques. Il existe plusieurs technologies développées et consacrés pour la localisation qui assurent une bonne précision, une couverture et une facilité de mise en place.

Dans notre vie quotidienne, l'apparition des nouveaux objectifs comme la localisation des objets perdus dans les grandes surfaces et le positionnement des voitures accidentés ou les équipements médicaux représentent l'objectif de notre projet.

Pour se faire, ce rapport est divisé en quatre chapitres :

Un premier chapitre où on présente l'entreprise d'accueil, l'étude et la critique de l'existant, la problématique et la solution proposée.

Le deuxième chapitre s'intéresse aux technologies utilisées.

Le troisième chapitre est consacré aux choix des matériels et aux tests qu'on a effectués pour la réalisation d'un prototype du projet.

Le dernier chapitre est consacré à la conception d'une carte électronique de géolocalisation dans lequel nous avons décrit les spécifications techniques de la carte et les choix effectués de sa conception.

Nous récapitulons à la fin par une conclusion générale et quelques perspectives.

Chapitre I. Cadre général du projet

I.1 Introduction

On s'intéresse dans ce chapitre à la représentation de la société d'accueil, lecadre général du projet et sa problématique en expliquant les différents objectifs à réaliser.

I.2 Présentation de l'organisme d'accueil

I.2.1 Présentation de la société (3S)

Standard Sharing Software (3S) est un groupe tunisien de hautes technologies fondé en 1988 (Logo figure 1) [1]. C'est une entreprise agile qui admet plus de 450 certifications positionnées comme l'intégrateur de réseaux n°1 en Tunisie et en Afrique du Nord. Ces axes d'activités principales sont l'intégration des systèmes, l'intégration des réseaux, l'édition de logiciels. Il offre des solutions innovantes dans les secteurs d'IT et Clean Tech. 3S crée une filiale appelée GlobalNet en tant que fournisseur d'accès Internet. En 2007, devient le premier partenaire de Cisco Systems en Tunisie, renforçant ainsi sa position sur le marché des réseaux informatiques.

En plus de GlobalNet, 3S possède également d'autres filiales spécialisées dans divers domaines des technologies de l'information, tels que la formation technique, la recherche de réseaux mobiles, le déploiement d'infrastructures GSM, le développement de logiciels de transaction, la technologie code-barres et RFID, les appels multicanaux, etc. Centrale et plateforme de réservation en ligne.

Figure 1 – Logo de l'entreprise 3S

I.2.2 Les filiales du Groupe 3s

Le groupe 3S a développé un réseau d'entreprises dans les secteurs des technologies informatiques et de communications, d'énergies renouvelables et de gestion d'énergie afin de répondre aux besoins de ses clients, d'offrir des solutions personnalisées et innovantes, ainsi que des solutions performantes avec l'optimisation desressources pour minimiser les coûts. Le groupe 3S est composé d'une douzaine de filiales telles que Treetronix, GlobalNet, Sky industries, ADVANCIA training,Smart Up, CYBERESA, Pro2c, Africa Solar, Candy Led, 3S SICAR, BIOPACK et BIOSEVE.

I.2.3 Présentation de la société Treetronix

Notre projet de fin d'année est réalisé au sein de la société Treetronix[2], qui est une filiale du groupe 3S (Logo figure 2). C'est une startup qui a été créée en 2018. Son siège social se situe à la Rue des Entrepreneurs Charguia 2 Tunis. Elle propose des solutionsde qualité dans le domaine de l'électronique embarquée, de l'internet des objets (IoT), de L'intelligence artificielle (AI), des systèmes d'information et des systèmes robotiques.

Figure 2 – Logo TREETRONIX

Treetronix développe des solutions connectées, innovantes et intelligentes qui répondent aux besoins des différents secteurs d'activités tels que : agricultures, automobiles, transports, biomédicales, gestion d'énergie, contrôles industriels et bâtiments. Elle propose des solutions ciblées à ces secteurs selon les besoins des clients, afin d'optimiser les ressources et améliorer la fonctionnalité de leurs produits. Ces solutions sont autour de l'industrie 4.0, l'agriculture intelligente, les bâtiments intelligents, les maisons intelligentes, les villes intelligentes (l'éclairage intelligent etle stationnement intelligent) et les automobiles.

La figure 3 regroupe les secteurs d'activités de Treetronix.

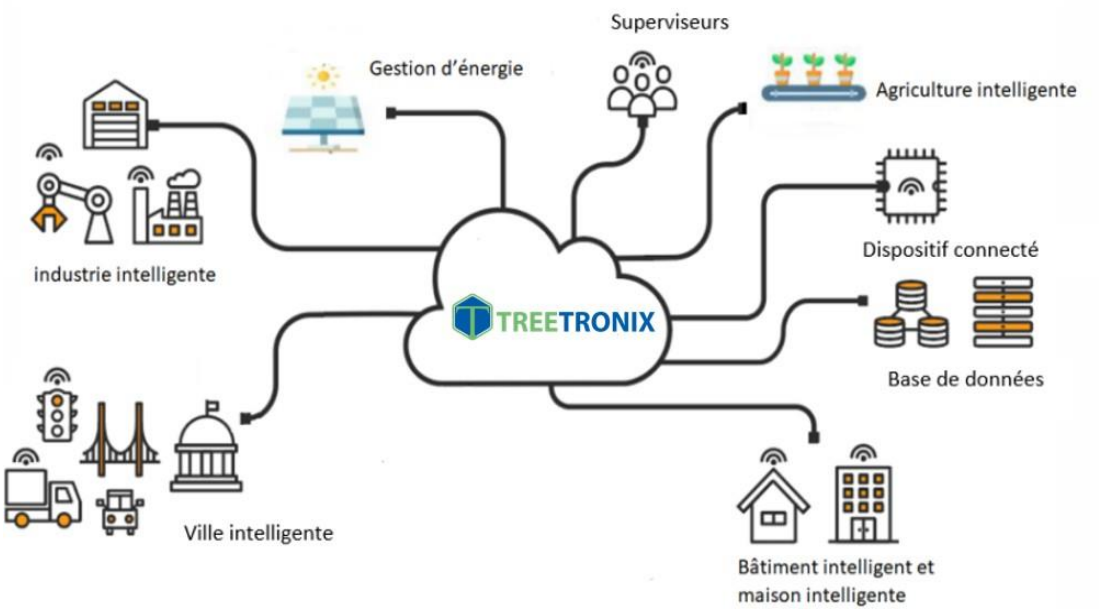


Figure 3 – Secteurs d'activités de Treetronix

I.3 Présentation du projet

Avec le développement rapide de la technologie Internet des objets (IoT) et la demande croissante de géolocalisation précise, le marché de la géolocalisation a connu une expansion significative. La géolocalisation IoT est unique dans sa capacité à connecter et à localiser des objets physiques via des appareils IoT, fournissant des informations précises et réelles. Positionnement du temps et ouvrant de nombreuses possibilités d'application dans divers domaines.

Ce projet consiste à réaliser une solution (IoT) de géolocalisation en exploitant la technologie LoRa. On va étudier en premier lieu, la géolocalisation d'objets connectés et les différentes technologies existantes pour la réaliser.

Puis, on va distinguer les matériels et les logiciels qu'on va utiliser pour développernotre solution.

Les caractéristiques de notre projet :

- C'est une solution conçue pour être utilisable dans n'importe quel endroit.
- La faible consommation d'énergie pour tous matériels électroniques utilisés.
- La longue portée de cette technologie a le potentiel d'envoyer toutes les coordonnées de localisation du site via le protocole LoRaWAN.

I.4 Problématique

Avec l'évolution du rythme de vie qui est devenue plus rapide et avec l'utilisation fréquente des appareils mobiles, les applications de la géolocalisation font désormais partie de la vie quotidienne, c'est pourquoi les clients cherchent toujours des solutions pour les localiser et surtout localiser leurs objets connectés en cas de perte ou de vol, de plus les applications et les objectifs de géo positionnement se diversifient, de l'aide à la navigation à la mise en relation des personnes en passant par la gestion en temps réel du personnel et des véhicules de l'entreprise. C'est pourquoi les clients cherchent une solution à utiliser partout dans l'industrie, les hôpitaux, les écoles, etc.

Une solution qui permet de gagner le temps, réduire les couts et hausser la productivité.
La géolocalisation est une technologie qui permet de déterminer la localisation d'un objet ou d'une personne avec une certaine précision, en identifiant ces coordonnées géographiques dans l'espace. Cette technologie repose généralement sur le système de positionnement global (GPS) ou bien sur les interfaces de communication cellulaire. C'est pourquoi qu'il faut bien choisir les techniques nécessaires qui conviennent avec le milieu de travail et l'objectif à atteindre.

I.5 Etude de l'existant

I.5.1 Exemples de solutions de géolocalisation

La technologie de suivi des objets est devenue de plus en plus courante dans notre vie quotidienne. Grâce à des capteurs et des systèmes de géolocalisation, il permet aux entreprises de suivre les déplacements de leurs marchandises, collaborateurs ou clients. Dans le but d'obtenir la meilleure précision, le cout minimal et la longue portée et également la possibilité de suivre les animaux, les personnes, les véhiculeset même certains objets en temps réel pour un simple utilisateur. Il existe plusieurs solutions de géolocalisation disponibles sur le marché, chacune avec ses propres avantages et inconvénients. Voici quelques exemples :

WhatsApp Live Location

WhatsApp propose une fonctionnalité de partage de position en temps réel appelée "Live Location". Ces fonctionnalités utilisent la géolocalisation GSM pour partager la position actuelle de l'utilisateur avec ses contacts pendant une période définie.

Les inconvénients potentiels de la géolocalisation GSM sont :

• **Précision variable** : la précision de la géolocalisation GSM peut varier en raison de facteurs tels que la densité des antennes relais, les obstacles physiques, les interférences électromagnétiques et la puissance du signal téléphonique. La précision peut être acceptable dans les zones urbaines densément peuplées, mais moins dans les zones rurales ou montagneuses.

• **Latence** : la géolocalisation GSM peut entraîner des retards dans la transmission des données de localisation. Cela peut être un problème lorsque la localisation en temps réel est essentielle, comme dans les situations d'urgence.

• **Consommation d'énergie** : L'utilisation continue de la géolocalisation GSM peut entraîner une augmentation de la consommation de la batterie de votre appareil mobile. Cela peut raccourcir la durée de vie de la batterie et nécessiter des recharges plus fréquentes.

Un relais GPS Tracker de suivi de voiture :

Ce dispositif utilise des signaux GPS pour déterminer l'emplacement d'un véhicule (figure 4). Il peut utiliser le réseau GSM pour communiquer ces informations à un utilisateur distant. Le dispositif est généralement installé dans une voiture et il peutêtre configuré pour envoyer des mises à jour de position à intervalles réguliers ou àla demande. Les informations sur la position du véhicule peuvent être consultées via des SMS.

Figure 4 – Relais GPS Tracker De Suivi De Voiture

L'inconvénient potentiel de l'utilisation d'un relais GPS Tracker de suivi de voiture est le coût d'abonnement, il peut être assez coûteux à l'achat, en plus descoûts d'installation et d'abonnement à un service de suivi. Ces coûts peuvent être un obstacle pour certaines personnes ou entreprises.

En outre, Certains dispositifs de suivi GPS peuvent consommer une quantité importante d'énergie, ce qui peut entraîner une réduction de la durée de vie de la batterie de la voiture.

Il est important de choisir un dispositif de suivi qui consommeune quantité minimale d'énergie pour éviter ce problème.

AirTag d'Apple :

Ce petit dispositif de géolocalisation conçu et fabriqué par Apple est destiné à aider les utilisateurs à localiser des objets perdus tels que les clés, les portefeuilles, les animaux domestiques, les véhicules ou les sacs à dos (figure 5).

Par ailleurs, il peut être utilisé pour suivre les enfants lorsqu'ils sont à l'école ou à l'extérieur de la maison.

Bien que l'AirTag soit un produit utile pour aider à localiser des objets perdus,il y a certains inconvénients à prendre en compte :

- **Portée limitée** : L'AirTag utilise la technologie Bluetooth pour communiquer avec votre appareil Apple, ce qui signifie que sa portée est limitée. Si vous êtes loin de votre AirTag, vous ne pourrez pas le localiser avec précision.
- **Risque d'interférence** : si plusieurs AirTags sont utilisés dans un même espace,il peut y avoir des interférences entre les signaux Bluetooth, ce qui peut rendre difficile la localisation précise d'un AirTag particulier.
- **Nécessité d'un iPhone** : Pour utiliser pleinement l'AirTag, vous devez posséder uniPhone ou un iPad avec la dernière version du système d'exploitation. Cela signifie que si vous n'avez pas d'appareil Apple, l'AirTag n'est pas une option pour vous.
- **Consommation de la batterie** : l'AirTag utilise une pile bouton remplaçable, qui a une durée de vie d'environ un an. Cela signifie que les utilisateurs devront remplacerla pile régulièrement, ce qui peut être coûteux et contribue à la production de déchets électroniques. De plus, l'utilisation de l'AirTag peut également entraîner une consommation accrue de la batterie des appareils Apple utilisés pour la géolocalisation, ce qui peut affecter l'autonomie de la batterie de ces appareils.

Figure 5 – AirTag d'Apple

I.6 La solution proposée

Afin de remédier les problèmes qu'on a cités dans le paragraphe précédent, Ce projet consiste à développer une solution IoT de géolocalisation des objets connectés plus précise, faible consommation d'énergie, faible coût, facile à intégrer et à longueportée qui est une chose garantie par l'utilisation de la technologie LoRa.

Le concept de cette solution, c'est que l'objet connecté envoie l'information de son capteur au Gateway via le protocole de communication LoRaWAN qui à sontour transmet les trames ou bien les données au serveur LoRaWAN. Ces données sont ensuite décodées et transmises à l'application serveur ou à la plate-forme IoT Cayenne qui la décode et la convertit en données utilisables pour l'utilisateur final. Ces données peuvent inclure des mesures de géolocalisation telles que l'altitude, la longitude et la latitude, ainsi que des informations sur l'accélération, le gyroscope et d'autres capteurs intégrés au dispositif LoRaWAN. Ces mesures peuvent ensuite être affichées sur une carte interactive ou dans un tableau de bord personnalisé.
La figure 6 ci-dessous représente l'architecture générale de la chaine IOT de cettesolution.

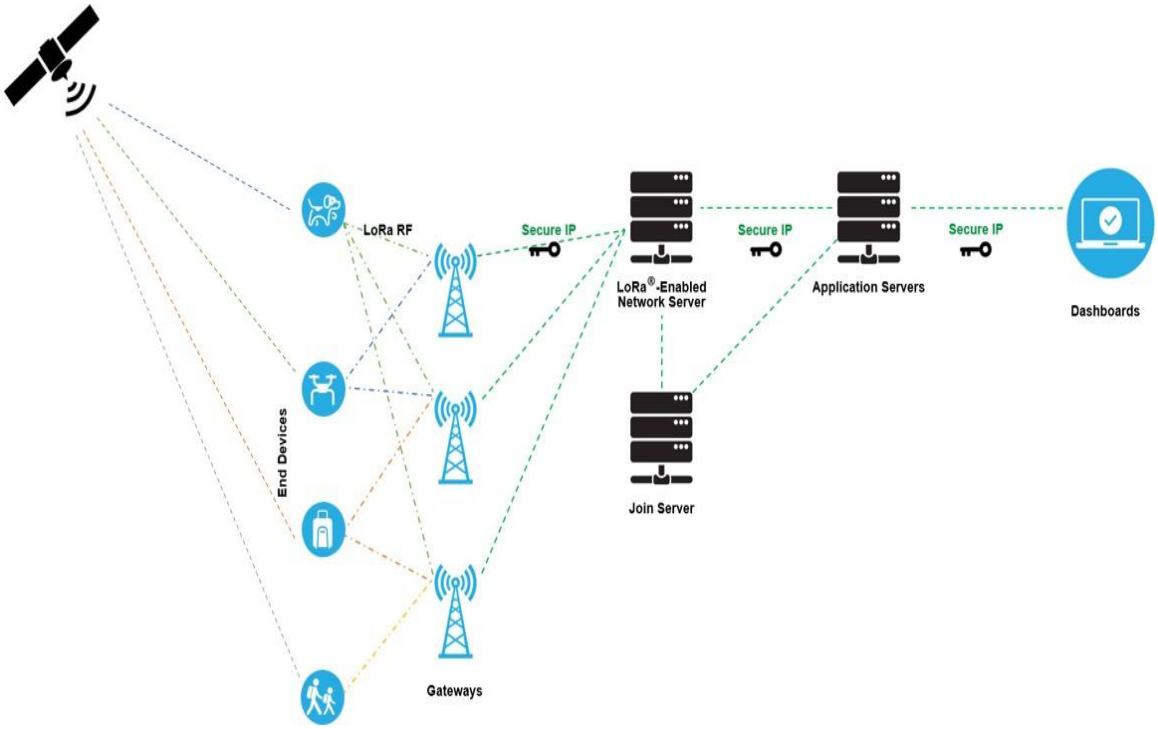


Figure 6 – Architecture de la solution proposée

I.7 Analyse fonctionnelle

I.7.1 Diagramme FAST

Une analyse fonctionnelle technique permet une transition entre une analyse des besoins fonctionnels et une planification détaillée incluse dans les considérations techniques

Nous développons le diagramme FAST illustré dans la figure ci-dessous pour effectuer l'analyse des capacités techniques. Cette analyse permet de déterminer les solutions technologiques utilisées pour réaliser les fonctions du système.

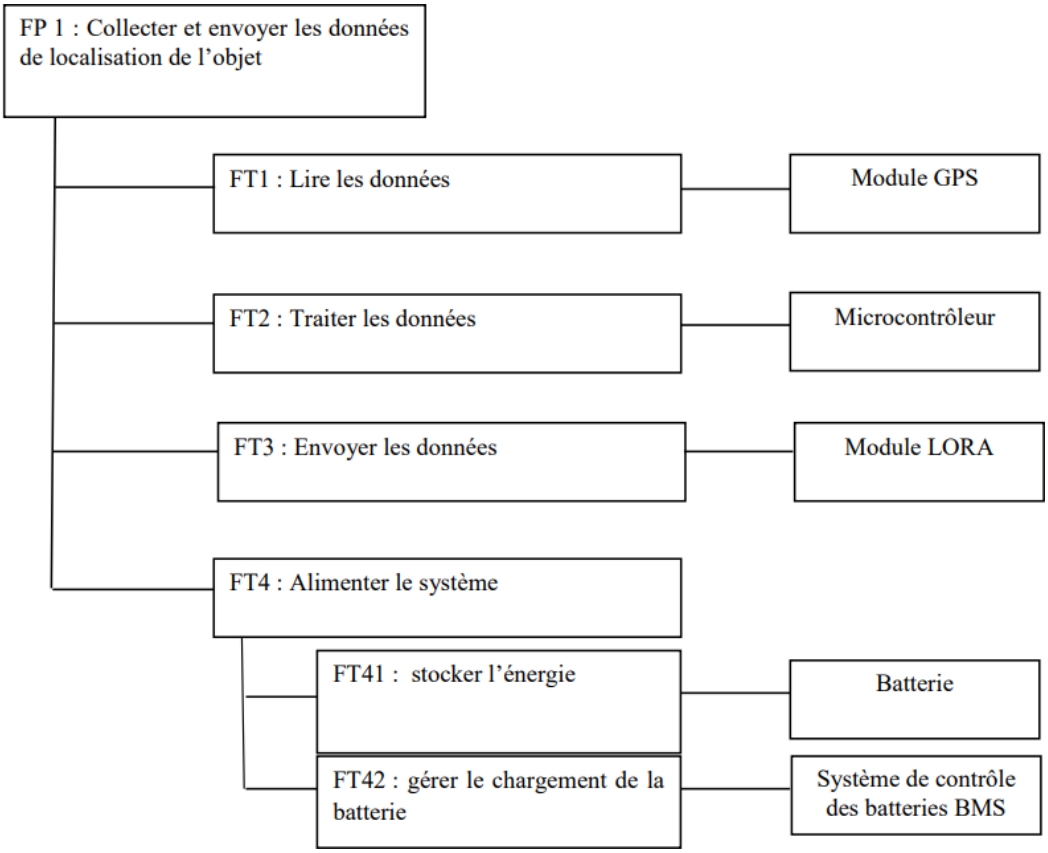


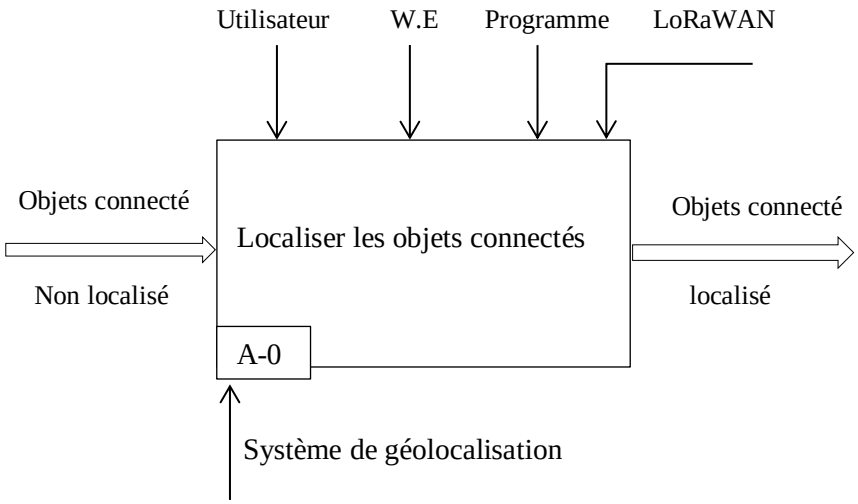
Figure 7– Le Diagramme FAST

I.7.2 Analyse S.A.D.T

Les techniques d'analyse et de modélisation S.A.D.T structurées sont une approche descendante et hiérarchique pour décrire la fonctionnalité globale d'un système. Pour ce faire, nous utilisons d'abord une boîte dans le diagramme pour montrer les capacités globales de notre système (la fonctionnalité globale du système), et pour montrer le trafic entrant et sortant et les données de contrôle ainsi que le support actif.

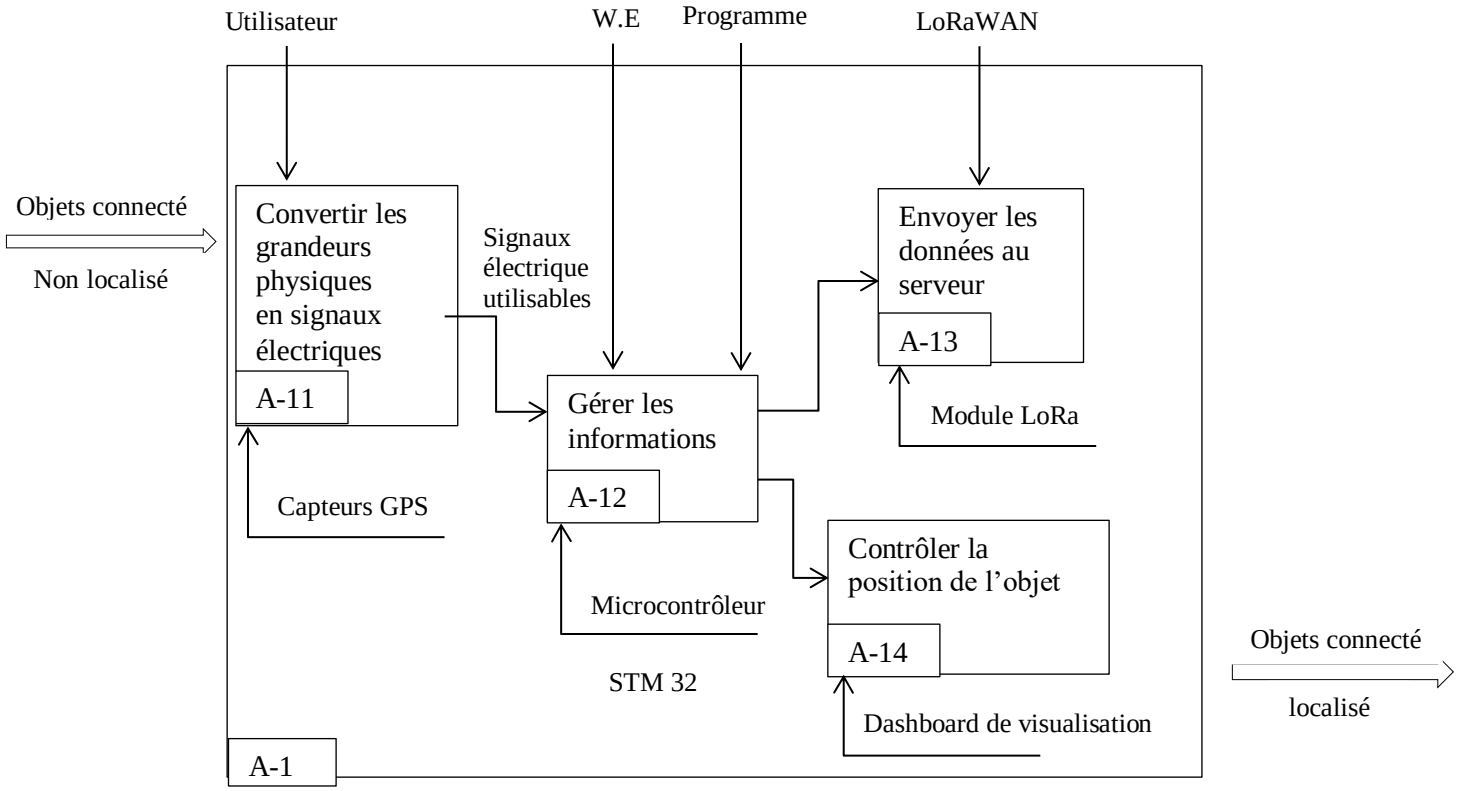
Ensuite, nous décrivons l'analyse descendante de notre système, comme le montre le diagramme A-1.

➢ **Niveau A-0 : La fonction globale du système**



La fonction globale peut être détaillée en des sous-fonctions comme illustre le diagramme suivante.

➢ **Niveau A-1**



I.8 Spécification des besoins

I.8.1. Spécification des besoins fonctionnels

Les exigences fonctionnelles de mon système de géolocalisation se concentrent sur les fonctions et opérations spécifiques que le système doit fournir pour répondre aux besoins des utilisateurs. Cela inclut la possibilité de fournir une localisation précise des objets connectés, de permettre le contrôle à distance de la géolocalisation, de traiter et d'analyser les données pour obtenir des informations utiles et de partager les données de géolocalisation avec les réseaux d'entreprise.

Besoin fonctionnel global : Géolocalisation.

- Sous-besoin 1 : La solution doit être capable de localiser avec précision les objets ou les personnes à l'extérieur d'un bâtiment, en utilisant des technologies de géolocalisation telles que le GPS.
- Sous-besoin 2 : La solution doit permettre de suivre en temps réel les objets ou les personnes localisées à l'extérieur, en affichant leur position sur une carte virtuelle.

I.8.2 Spécification des besoin non fonctionnels

Les exigences non fonctionnelles, également appelées exigences non fonctionnelles, décrivent les caractéristiques et les contraintes d'un système au-delà de sa fonction spécifique. Ils définissent les attentes en termes de qualité, de performance, de sécurité et d'expérience utilisateur. Dans la conception et la mise en œuvre de solutions de géolocalisation IoT, certaines exigences non fonctionnelles sont essentielles, telles que :

• **Performances en temps réel** : le système doit pouvoir fournir des informations de localisation en temps réel avec une latence minimale et une réactivité rapide.

• **Convivialité** : L'interface utilisateur doit être intuitive, conviviale et facile à naviguer.

• **Évolutivité** : Le système doit être conçu de manière à pouvoir supporter à l'avenir de nouveaux types d'objets connectés et de nouvelles fonctions.

• **Sécurité** : Les données de géolocalisation doivent être cryptées et protégées contre tout accès non autorisé.

I.1 Conclusion

Ce chapitre d'introduction donne un aperçu du projet proposé par l'organisme d'accueil. Nous avons présenté le projet d'une manière qui donne une image claire du travail à faire. Une fois les objectifs de la problématique sont identifiés, il faut justifier le choix des technologies utilisé dans la solution proposées. Dans le chapitresuivant nous allons préciser la technologie utilisée pour collecter les données.

Chapitre II. Généralités sur L'IOT et LoRaWAN

II.1 Introduction

Dans cette section nous présentons des notions théoriques en relation avec notre système à concevoir et que leur maitrise est primordiale pour réussir la mise en place d'un tel système. Nous commençons par l'IoT la définition et son architecture. Puis Nous étudions les différents protocoles de communication destiné à l'IoT afin de faire une étude comparative pour choisirle plus adéquat à notre système et nous allons présenter quelques notions autour de la technologie de modélisation LoRa et le protocole de communication loRaWAN.

II.2 L'internet des objets

II.2.1 Définition :

L'Internet des objets (IoT) :

L'Internet des objets désigne le réseau d'objets physiques connectés à Internet et capables d'échanger des données entre eux. Cette connectivité des objets apporte une dimension d'intelligence et d'automatisation dans de nombreux domaines.

Les objets IoT sont équipés de capteurs, d'actionneurs, de microprocesseurs et de connectivitésans fil ou filaire.

Grâce à ces éléments, ils sont en mesure de :

- ✓ Détecter des conditions dans le monde physique via leurs capteurs
- ✓ Traiter les données captées par les capteurs
- ✓ Communiquer avec d'autres objets et avec Internet
- ✓ Exécuter des actions via leurs actionneurs
- ✓ Automatiser certaines tâches sans intervention humaine

Les données générées par les objets IoT peuvent être utilisées pour optimiser des processus, prendre des décisions avisées et créer des services à valeur ajoutée.

L'IOT permet à de nombreux secteurs de devenir plus intelligents et efficaces (Figure8) : villes, bâtiments,logistique, Santé, ...

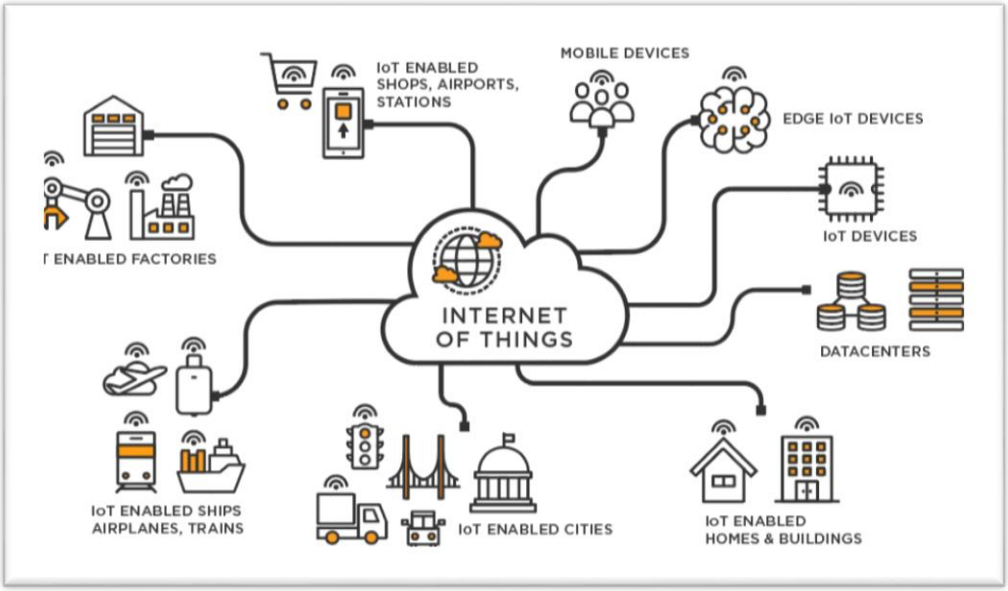


Figure 7 – l'IoT

II.2.2 Un Objet Connecté

Un objet connecté : peut-être tout objet embarqué qui peut transférer des données sur un réseau sans aucune interaction humaine comme les smartphones, les capteurs, les machines dans les usines, les appareils dans les maisons et les bâtiments, ou les composants d'un véhicule.

Un objet connecté peut être contrôlé à distance et assure généralement deux rôles :

- un rôle de capteur pour surveiller l'apparition d'un événement ou d'une mesure spécifique (capteur de présence, capteur thermique, ...).
- un rôle d'actionneur pour réaliser une action suite à un événement spécifique mesuré ou détecté (déclenchement d'une alarme en cas d'intrusion, ouverture d'une porte à distance...).

II.2.3 L'architecture IOT

L'architecture globale de l'Internet des Objets (IOT) se compose de plusieurs niveaux :

- Le niveau capteur : Il est constitué des différents capteurs et actionneurs qui collectent des données sur l'environnement physique et agissent dessus. C'est le niveau le plus bas.

 Le niveau connectivité : Il comprend les technologies réseau filaires ou sans fil qui permettent aux objets de communiquer entre eux et avec Internet. Il peut s'agir de réseaux cellulaires. Wi-Fi, Bluetooth, LoRa, etc.

 Le niveau traitement : Ilregroupe les passerelles, microcontrôleurs et autres objets capables de traiter les données collectées localement avant de les transférer au cloud.

 Le niveau cloud : C'est le cœur névralgique de l'IoT qui permet le stockage et l'analyse à grande échelle des données, l'exposition d'API et le pilotage des objets connectés.

 Le niveau application : Il comprend les différentes applications métier utilisant les données IoT pour améliorer un processus ou fournir un service.

 Le niveau interface utilisateur : Il contient les interfaces (tableaux de bord, assistants vocaux, applications mobiles) qui permettent aux utilisateurs finaux de superviser et contrôler l'IOT.

En résumé, cette architecture modulaire de l'IoT permet de séparer clairement les différentes fonctions : collecte des données, connectivité, traitement local, traitement dans le cloud, applications métiers et interfaces utilisateurs. Chaque couche interagit avec les couches adjacentes pour former un système IoT global et cohérent.

Cependant, cette architecture reste quelque peu théorique car dans la réalité, les frontières entre les différentes couches peuvent être floues. Mais elle aide à comprendre le fonctionnement d'ensemble de l'Internet des Objets.

II.2.4 Les protocoles de communication pour L'IOT

Une étude des différents protocoles de communication destinés à l'IoT est primordiale afin d'en choisir le plus adéquat càd qui répond aux différents besoins fonctionnels et non fonctionnels déjà soulevés dans le précédent chapitre.

II.2.4.1 Les principaux protocoles de communication pour des objets (IOT)

L'IoT fait appel à divers protocoles de communication, dont nous distinguons

Zigbee

- Protocole maillé, à faible consommation, dédié aux réseaux de capteurs. Permet de connecter plusieurs milliers de nœuds sur quelques dizaines à centaines de mètres

LoRaWAN

- C'est un protocole longue portée et à bas débit, dédié principalement aux applications IoT nécessitant une faible consommation énergétique. Il permet de connecter des capteurs et objets sur plusieurs kilomètres.

Bluetooth Low Energy (BLE)

- Norme présente dans de nombreux appareils. Elle permet des communications à courte portée, mais à un débit plus élevé. Utilisée dans les wearables et certains capteurs

WiFi

- Norme de communication sans fil grand public, débit élevé mais consommation plus importante. Peu utilisée nativement dans l'IoT.

NB-IoT

- Réseau basé sur la 4G, dédié à l'IoT. Il offre de bonnes performances en termes de capacité, de couverture et de consommation tout en restant abordable.

LTE-M

- Basé également sur la 4G, il vise les mêmes applications IoT que le NB-IoT, mais avec un meilleur débit.

6LowPAN

- Réseau maillé IP compatible IPv6, à très faible consommation. Il permet de connecter des milliers d'objets sur des distances de l'ordre du kilomètre.

Sigfox

- Autre protocole longue portée à très bas débit, propriétaire. Il vise les mêmes types d'applications que LoRaWAN.

II.2.4.2 Etude comparative des protocole IoT

La figure 2.2 présente une étude comparative entre les différents protocoles en fonction de la portée, débit, cout et la puissance consommée …

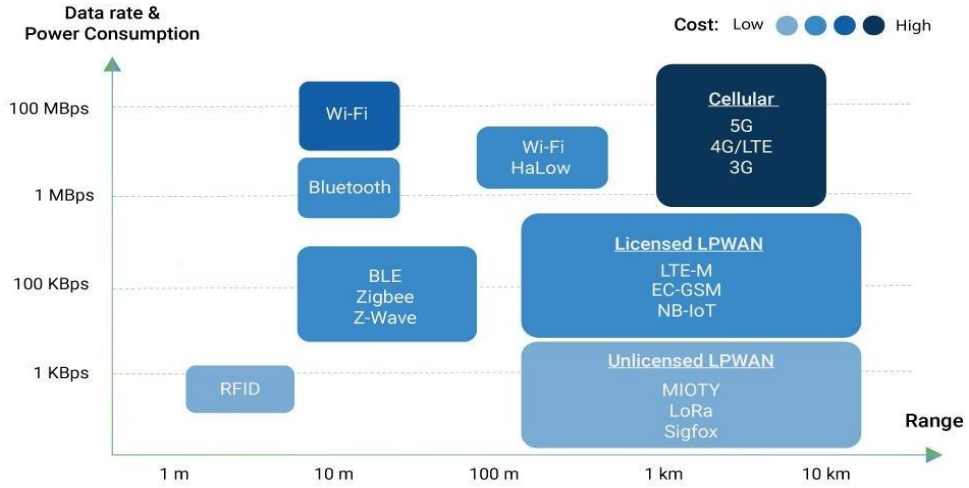


Figure 9 - Etude comparative des protocoles IoT

D'après la figure ci-dessous, les deux protocoles LoRaWAN et Sigfox répondent aux mieux à l'exigences de notre système à savoir : la longue portée, la faible consommation, le faible cout…

Nous allons à ce stade, faire une comparaison entre ces deux protocoles afin de trancher et choisir le meilleur.

Tout comme LoRaWAN, le protocole Sigfox vise avant tout les applications de l'Internet des Objets nécessitant :

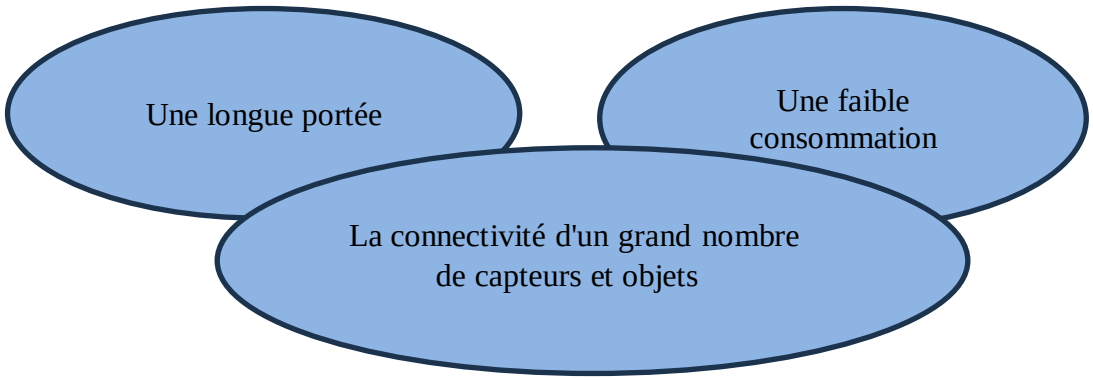


Figure 9- les principaux avantages communs

Le tableau 1 résume les différences entre les protocoles Sigfox et LoRaWAN :

Tableau 1- étude comparative entre les protocoles Sigfox et LoRa WAN

Comparaison	LoRaWAN	Sigfox
Portée	Jusqu'à 15 km en zone rurale	Jusqu'à 10 km
Débit	Faible à moyen (50 kbps)	Très faible (100 bps)
Densité du réseau	Forte, jusqu'à 1 million de nœuds/ km2	Moyenne,100000 nœuds/km2
Consommation d'énergie	Plus élevée à cause du débit plus élevé	Très faible
Coût	Coût modéré	Coût bas
Fiabilité	Moyenne	Faible
Fonctionnalités	Bidirectionnel, mises à jour à distance	Unidirectionnel, sans mise àjour

Au vu des performances supérieures de LoRaWAN mises en évidence par ce tableau comparatif, nous avons sélectionné ce protocole pour notre projet IoT.

LoRaWAN surclasse en effet Sigfox sur des critères essentiels pour nous tels que la portée, le débit, la densité du réseau et les fonctionnalités, même si cela se fait au prix d'une consommation d'énergie et de coûts matériels plus importants qui restent toutefois dans les limites de notre cahier des charges.

II.3 LoRa

La couche physique LoRa est développée par la société Semtech, vise à mettre en place un réseau étendu de faible puissance (Low-Power Wide-Area Network (LPWAN) basé sur une technologie[3] radio permettant des communications sansfil à longue portée et à faible débit (figure 11)[4].

Elle s'apparente à un protocole cellulaire (systèmes mobiles 2G/3G/4G) mais optimisée pour IoT/M2M.

Figure 10 – LoRa

La technologie LoRa ne nécessite pas de licence d'utilisation du spectre, car elle utilise des bandes de fréquences libres, par exemple, 863-870 MHz en Europe, 902-928 MHz aux États-Unis et 779-787 MHz en Chine.

II.4 LoRaWAN

LoRaWAN est un protocole de communication [5] conçu pour les réseaux IoT (Internet des objets) à faible consommation d'énergie et à longue portée, offrantune solution pour les applications nécessitant une communication à longue distanceavec des dispositifs connectés à faible consommation d'énergie (figure 12), il vise également à sécuriser la couche de contrôle d'accès au support d'un réseau LoRa .

Les cas d'utilisation auxquels LoRaWAN vise à répondre sont nombreux tels que :

Figure 12 – LoRaWAN

- **Surveillance environnementale** : LoRaWAN peut être utilisé pour surveiller lesparamètres environnementaux tels que la qualité de l'air, la température, l'humidité,la pression atmosphérique, etc.

- **Gestion de l'eau** : LoRaWAN peut être utilisé pour surveiller les niveaux d'eau dans les réservoirs, les rivières et les lacs, pour aider à la gestion de l'eau en temps réel et à la détection des fuites.

- **Agriculture** : LoRaWAN peut être utilisé pour mesurer les conditions environnementales qui influencent la croissance des plantes, et pour contrôler les systèmes d'irrigation.

- **Smart City** : LoRaWAN peut être utilisé pour surveiller les équipements urbains tels que les lampadaires, les poubelles, les places de parking, les panneaux de signalisation, etc.

- **Logistique et chaîne d'approvisionnement** : LoRaWAN peut être utilisé pour suivre les expéditions et les livraisons en temps réel, pour surveiller les conditions environnementales dans les entrepôts et les camions de transport, et pour optimiserla logistique.

Le réseau LoRaWAN présente plusieurs avantages, notamment une couverture étendue et une consommation d'énergie réduite.

Dans le cas d'un réseau LoRaWAN, le serveur IoT reçoit les données de la passerelle LoRa, puis les stocke. Ces données peuvent ensuite être accessibles à partir de n'importe quel terminal connecté (un ordinateur, un ordinateur portable, smartphone ou une tablette).

- **Les avantages clés du réseau LoRaWAN :**

a) Longue portée : Le réseau LoRaWAN se distingue sa capacité à couvrir des zones étendues atteignant les quinzaines de kilomètres, sans nécessiter l'utilisation de répéteurs. Par conséquent, cette option est un choix considérable pour les cas d'utilisation qui impliquent une communication à longue portée.

b) Faible consommation d'énergie : Les appareils connectés au réseau LoRaWAN peuvent fonctionner pendant de nombreuses années sans avoir besoin de changer les piles en raison de sa faible consommation d'énergie.

c) Coût réduit : Les frais de mise en place et de préservation des autres modes de réseaux de communication comme le Wi-Fi ou le GSM sont comparativement supérieurs à ceux du réseau LoRaWAN.

d) Grande capacité de connexion : Le réseau LoRaWAN possède une capacité impressionnante en termes de capacité de se connecter à des milliers d'appareils simultanément, ce qui en fait une solution hautement adaptée et efficace pour les applications de l'IoT.

e) Sécurité: Le réseau LoRaWAN déploie des mesures de sécurité assez sophistiquées pour garantir la sécurité des données transmises.

II.5 Architecture du réseaux LoRaWAN

La Figure 13 montre l'ensemble de l'architecture LoRaWAN.

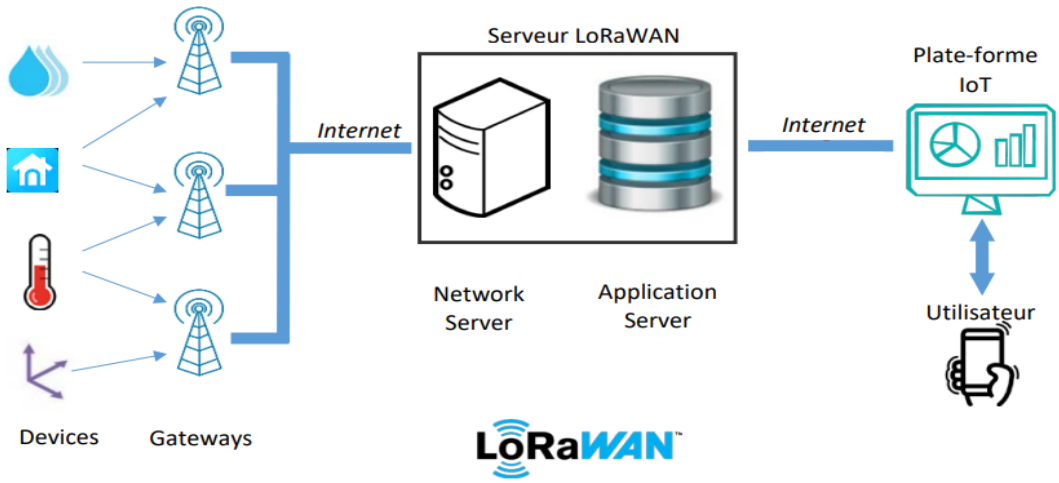


Figure 13 – Structure globale d'un réseau LoRaWAN

Les End Devices , les Gateways, le Network Server et l'Application Server constituent le cœur de l'architecture LoRaWAN. Mais Le Plat-forme IoT et la connexion de l'utilisateur (à droite sur la figure) n'ont rien à voir avec ce protocole, car il s'agit simplement d'un service web classique.

Sur le côté gauche,on trouve les Devices LoRaWAN qui transmettent les données. L'utilisateur se trouve de l'autre côté et réceptionne les données qui ont transité sur le réseau.

II.5.1 Les devices (dispositifs) LoRaWAN

Les devices LoRaWAN sont des systèmes électroniques embarqués faisant partie de l'univers de l'IoT, caractérisés par leur faible consommation d'énergie, leur petite taille, leur faible puissance et leur coût abordable. Ils sont équipés d'un émetteur-récepteur radio LoRa qui leur permet de communiquer avec les Gateways.

Contrairement à d'autres technologies, un device LoRaWAN n'est pas lié à un Gateway spécifique, car tous les devices situés dans la zone de couverture peuvent transmettre des données à toutes les Gateways présents et disponibles.

II.5.2 Les Gateway (Passerelles) LoRaWAN

Les Gateways LoRaWAN sont équipées pour écouter sur tous les canaux et tous les facteurs d'étalements simultanément.

Lorsqu'elles reçoivent une trame LoRa, elles transmettent son contenu au serveur LoRaWAN préalablement configuré.

Chaque Gateway LoRaWAN a un identifiant unique qui lui permet de s'activer ou s'enregistrer sur un network server.

II.5.3 Le Network Server (serveur web)

Les messages envoyés par les Gateways sont reçus par le Network Server, qui se charge d'éliminer les duplicatas. Plusieurs Gateways peuvent recevoir le même message et le transmettre au Network Server, d'où la nécessité de cette étape de suppression des doublons. Ensuite, le Network Server authentifie le message en utilisant une clé de chiffrement AES 128 bits appelée NwkSKey.

II.5.4 L'Application Serveur (serveur d'application)

Des messages chiffrés provenant d'un Network Server sont reçus par l'Application Server. Le chiffrement et les déchiffrements se font à l'aide d'une clé AES 128 bits appelée AppSKey.

II.5.5 Serveur LoRaWAN

Un serveur LoRaWAN est un système combinant un Network Server et un Application Server pour gérer les messages transmis par les Gateways. Le Network Server authentifie le message à l'aide de la clé d`authentification et l'Application Server chiffre les données avant de les envoyer à la plateforme IoT ou à l'utilisateur final. Pour assurer une sécurité de bout en bout, il est recommandé d'intégrer son propre Application Server dans l'architecture.

II.5.6 La Plateforme IOT

L'application utilisateur se situe à cet endroit. Les services principaux comprennent :

- Un connecteur pour récupérer les données de l'Application Server via des protocoles sécurisés tels que HTTP(S) ou MQTT(S).
- Une base de données où sont stockées les données collectées.
- Un tableau de bord est disponible pour l'utilisateur via une page web ou une application mobile.

II.6 Les Classes du réseau LoRaWAN

II.6.1 Classe A

Dans la classe A, les dispositifs LoRaWAN peuvent émettre des données vers une passerelle (uplink) sans vérifier au préalable si la passerelle est disponible. Cela signifie que le dispositif peut envoyer ses données à tout moment, sans se soucier de la disponibilité de la passerelle pour la transmission. Après l'émission des données, le dispositif a deux fenêtres de réception très courtes RX1 et RX2. Ce sont des moments prédéfinis où le dispositif est à l'écoute des éventuels messages de la passerelle.

Aussi elles permettent au réseau serveur (via la passerelle) de transmettre un message de retour (downlink) au dispositif.

Cependant, il est important de noter que le réseau serveur peut transmettre le message de retour pendant soit le slot RX1, soit le slot RX2, mais pas les deux à la fois.

La figure 14 illustre le fonctionnement des dispositifs LoRaWAN de classe A.

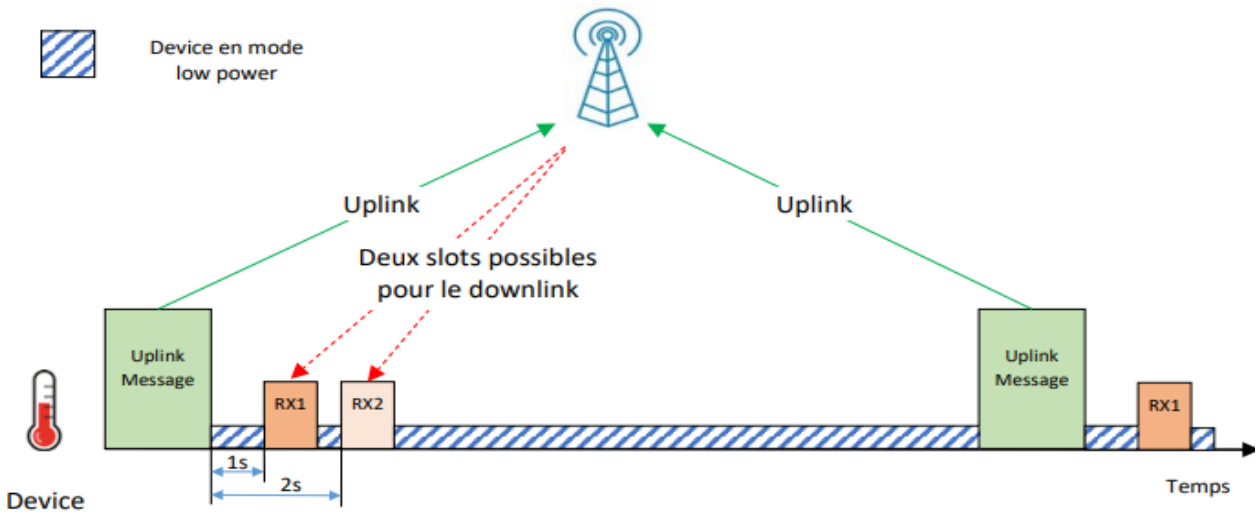


Figure 14 – Fenêtres de réception RX1 et RX2 pour un dispositif de classe A

II.6.2 Classe B

La classe B des dispositifs LoRaWAN se comporte de manière similaire aux dispositifs de classe A, mais avec des fenêtres de réception supplémentaires programmées à des moments précis.

Pour synchroniser ces fenêtres de réception, les passerelles doivent émettre un signal régulier appelé balise.

La balise permet aux dispositifs de synchroniser leurs fenêtres de réception supplémentaires avec la passerelle, afin de maximiser les chances de recevoir des messages de la passerelle. En plus de cela, la balise peut également fournir des informations sur le réseau et sur la passerelle aux dispositifs de classe B, tels que le temps et la fréquence à utiliser pour la communication.

La figure 15 illustre le fonctionnement des dispositifs LoRaWAN de cette classe B.

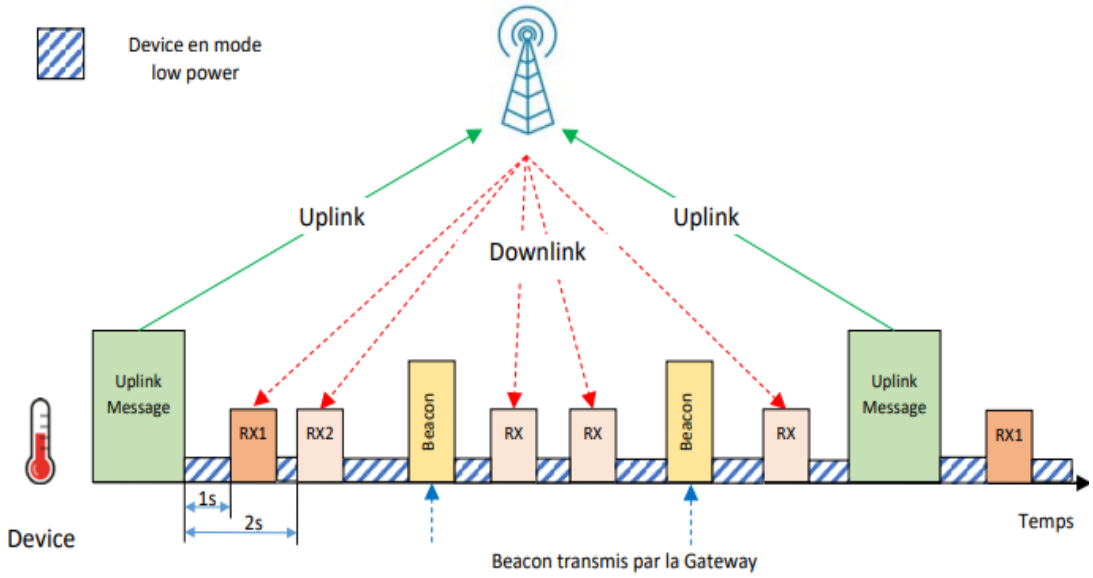


Figure 15 – Fenêtres de réception pour un dispositif de classe B

II.6.3 Classe C : Écoute en permanence

Les dispositifs de classe C ont des fenêtres de réception qui sont constamment ouvertes entre deux transmissions. Ces dispositifs consomment beaucoup plus d'énergie.

Le dispositif LoRaWAN est en écoute permanente entre deux messages uplink. Les paramètres de réception sont fixes, à l'exception de RX1 qui a toujours le même comportement que dans les classes A et B.

La Figure 16 présente plus de détails sur le fonctionnement des dispositifs LoRaWAN de classe C.

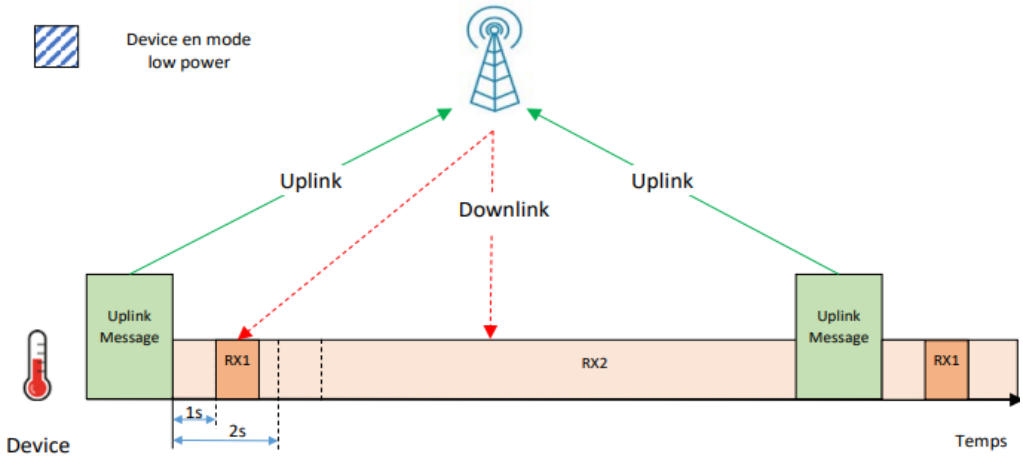


Figure 16 – Fenêtres de réception pour un dispositif de classe C

D'après les figures précédentes, nous pouvons remarquer que :

✓ Un dispositif de classe B est également un dispositif de classe A (les emplacements RX1 et RX2 sont toujours présents).

✓ Un dispositif de classe C est également un dispositif de classe A (les emplacements RX1 et RX2 sont toujours présents).

Ainsi, la classe B et la classe C sont donc une extension de la classe A. Concernant leur consommation, nous pouvons les représenter dans la figure 17 suivante.

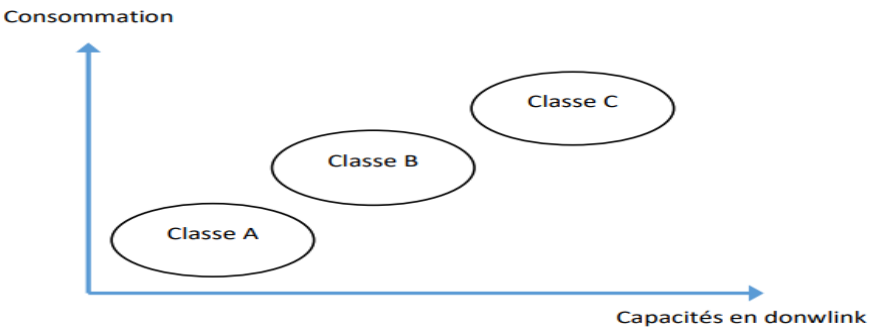


Figure 17 – Consommation d'énergie et capacités en downlink des dispositifs LoRaWAN

II.7 Activation d'un End-device

Afin de rejoindre le réseau, le device doit être d'abord activé sur ce réseau. Troiséléments sont indispensables pour l'activation des devices :

- **DevAddr** pour l'identification.
- **NwkSkey** pour l'authentification.
- **AppSkey** pour le chiffrement des données.

Voici la liste des éléments qui concerne l'activation d'un device au sein d'un réseauLoRaWAN.

- DevAddr : est un mot de 32 bits qui identifie l'end-devices dans le réseau loRaWAN.
- NwkSKey : Utilisé pour l'authentification avec le Network Server.
- AppSKey : Utilisé pour le chiffrement des données avec l'Application Server.
- DevEUI : Identifiant unique pour le Device LoRaWAN. C'est l'équivalent d'uneadresse MAC sur Ethernet.
- AppKey : Cette clé utilisée pour authentifier le Join-Request, pour crypter le Join-Accept et pour générer les "Sessions Keys" (Network Session Key et ApplicationSession Key). Cette clé doit être gardée secrète et ne jamais être partagée avec qui que ce soit.
- AppEUI/JoinEUI : identifie le provider de l'application.

Deux méthodes sont possibles pour fournir ces informations à la fois au nœud LoRa et au serveur :

- **A**ctivation **B**y **P**ersonalization : **ABP**
- **O**ver **T**he **A**ir **A**ctivation : **OTAA**

II.7.1 La méthode d'activation ABP

La méthode ABP représentée par la figure 18, est très simple, elle nécessite le stockage des même paramètres DevAddr, NwkSKey et AppSKey dans le device LoRaWAN et le serveur LoRaWAN pour assurer la communication entre eux.

Dès que le Device démarre, il peut envoyer et recevoir des messages LoRaWAN.

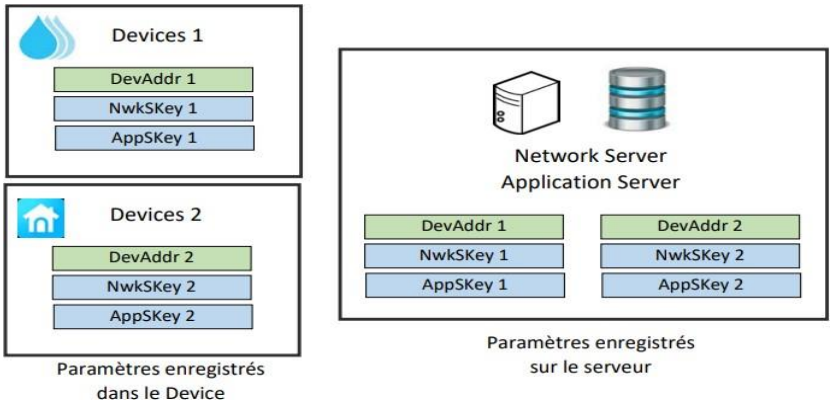


Figure 18 – DevAddr, NwkSkey et AppSkey en ABP

II.7.2 La méthode d'activation OTAA

Cette méthode d'activation nécessite l'enregistrement des trois éléments DevEUI, AppKey et AppEUI dans l'appareil LoRa et le serveur LoRaWAN, après une procédure de jointure au cours de laquelle l'appareil transmet une requête (joinRequest) pour joindre le réseau. Une fois l'appareil est autorisé par le serveur, ilreçoit une confirmation (join accept), ensuite les paramètres DevAddr, NwkSKey et AppSKey sont générés comme indique la figure 19.

Ce processus d'activation est effectué uniquement pendant la première connexion entre le dispositif LoRa et le serveur LoRaWAN.

La figure 19 montre le processus d'activation OTAA.

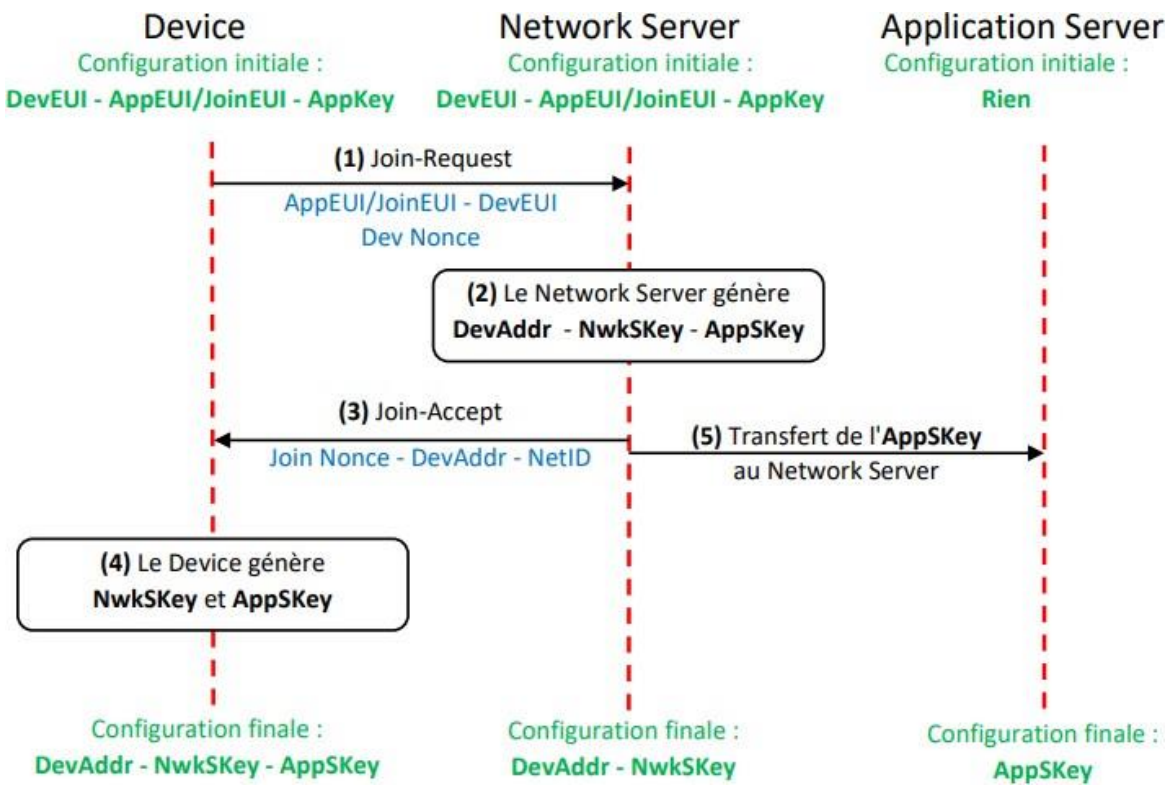


Figure 19 – Processus d'activation OTAA

Le tableau 2 situé ci-dessous résume les différents points entre les deux modes d'activation :

Tableau 2– tableau comparative entre l'ABP ET l'OTAA

Caractéristique	OTAA	ABP
Activation	Dynamique, en Over-The-Air	Par configuration
Installation	Ne nécessite aucune configuration préalable	Besoin d'une configuration préalable
Indépendance	Le dispositif est indépendant	Le dispositif dépend d'une configuration
Connections	Connectivité requise lors de l'activation	Pas besoin de connectivité lors del'activation
Mobilité	Le dispositif peut être remplacé facilement	Le dispositif est statique
Complexité	Plus complexe à mettre en œuvre	Plus simple
Sécurité	Meilleure, utilise des keys dynamiques	Moins bonne, utilise des keysstatiques
Mises à jour	Poussées par le réseau	Tirées par le dispositif

Le choix entre les modes ABP et OTAA se fait donc en fonction des priorités du déploiement : latence, débit, facilité de déploiement vs sécurité et facilité de gestion.

II.8 Plateforme utilisée au niveau du serveur LoRaWAN

ThingPark [6] est une plateforme IoT fournie par Actility, une société spécialisé dans le domaine de l'IoT.

ThingPark fournit une plate-forme complète pour créer et gérer des applications à grande échelle de l'IoT, Il offre des services et des fonctionnalités pour les développeurs tel que la gestion des données et des appareils, le déploiement, la connectivité sans fil, la création d'applications personnalisées d'une manière sécurisée et efficace.

Cette plateforme est formée de ces éléments :

- **ThingPark Wireless** : c'est un réseau basé sur la LoRaWAN permettant la communication des objets IoT sur de longue distance.

- **ThingPark Activation** : C'est un système de gestion et d'activation des objets connectés sur le réseau.
- **ThingPark Market** : C'est un marché en ligne qui fournit aux utilisateurs des applications, des solutions, des outils …
- **ThingPark X** : C'est un ensemble d'ApI au service des développeurs qui leur permettent d'accéder aux fonctionnalités de la plateforme.

Figure 20 - Le logo de la plateforme ThingPark

II.8.1 Exporter nos données du serveur LoRaWAN

Le groupe 3S a développé son propre network serveur LoRaWAN qui offre une meilleure connectivité LoRaWAN permettant de répondre aux besoins spécifiques de ses utilisateurs.il garantit une grande flexibilité et une disponibilité élevée, grâceà des fonctionnalités avancées telles que la gestion des clés de sécurité (DevAddr, NwKSkey, DevEUI...) et les différentes méthodes d'accès au réseau (ABP ou OTAA),ce qui permet d'ajouter rapidement et facilement des nouveaux end devices et de suivre les données en temps réel.

la figure au-dessous représente la page d'accueil de network serveur LoRaWAN du groupe 3s.

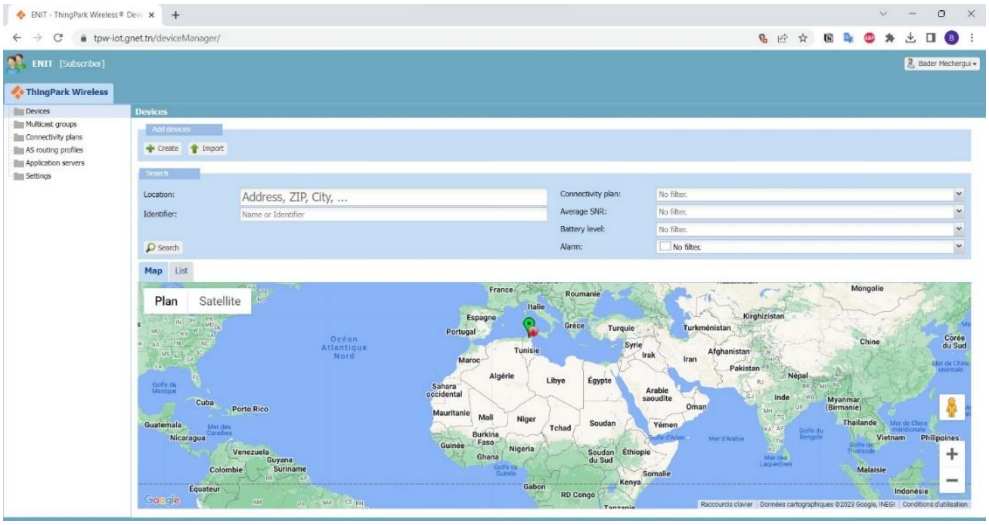


Figure 21 – Network server LoRaWAN du groupe 3s

II.8.2 Enregistrement du Device

Avant de pouvoir utiliser et expoilter notre end device LoRa sur le Networkserver du groupe 3s, il doit être enregistré sur le réseau.

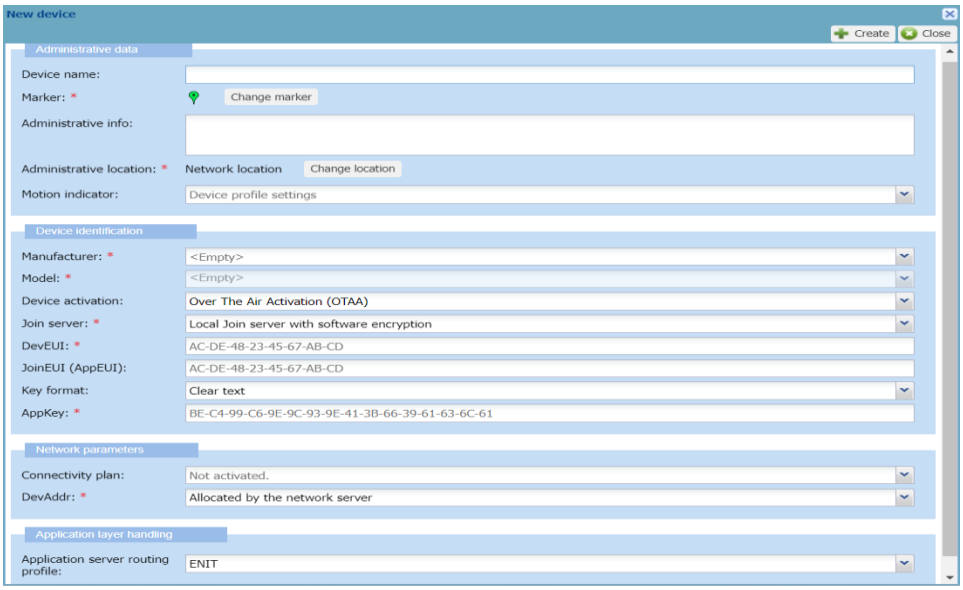


Figure 22 – Enregistrement d'un nouveau device

Le processus d'enregistrement d'un end Device nécessite plusieurs informations :

- le nom : Un identifiant défini par l'administrateur .
- Le Mode d'activation : ABP ou OTAA
- les clés de configuration : Appkey, DevEUI,...
- La classe du Device : A, B ou C.
- La séléction d'un profil de routage.

II.8.3 Créer un profil de routage Cayenne

Pour que les données de notre end device parviennent à Cayenne, nous avons besoins de créer un serveur d'application qui pointe vers Cayenne dans le network server [9]. Il faut également ajouter un profil de routage qui utilise ce serveur.

Lorsque nous effectuons cette étape qu'une seule fois, tous les end devices que nous avons souhaité ajouter à Cayenne devront alors être configurés pour utiliser le profil de routage, sinon Cayenne ne recevra pas leurs données.

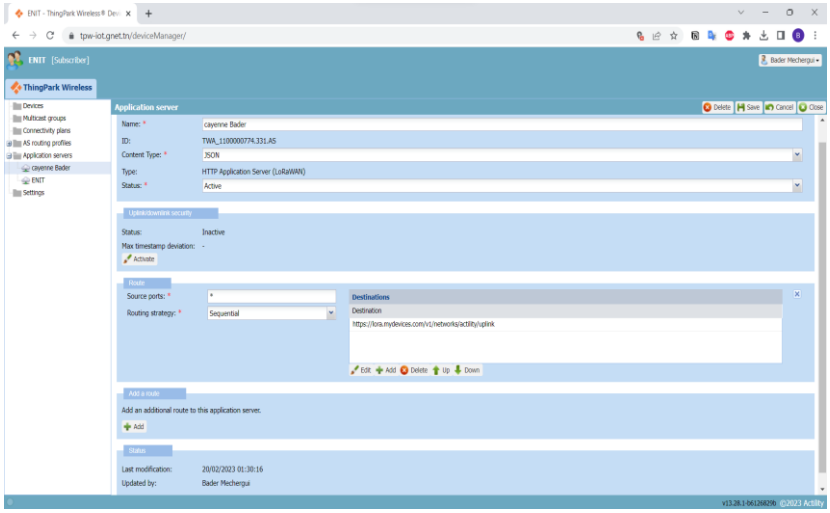


Figure 23 – Créer un profil de routage Cayenne

II.8.4 Ajouter un End Device à Cayenne

Une fois notre End Device enregistré, programmé et configuré pour transmettre des données à Cayenne, nous pouvons procéder à l'ajout de l'appareil sur Cayenne afin qu'il apparaisse dans un Dashboard.

Afin d'ajouter notre end device, Cayenne a besoin de connaître certaines informations sur l'appareil et comment il sera affiché sur le dashboard :

- Un nom à notre End Device.
- Un DevEUI.
- Un ThingPark Server
- Un email ThingPark et un mot de passe ThingPark

la figure au-dessous représente comment Ajouter un End Device à Cayenne

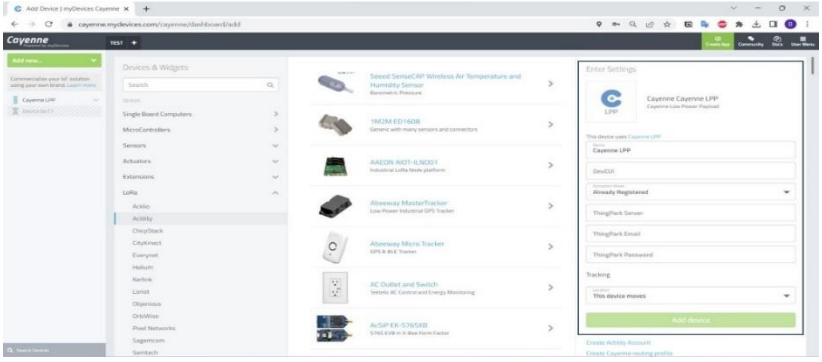


Figure 24 – Ajouter un End Device à Cayenne

II.1 Conclusion

Dans ce chapitre, nous avons expliqué le choix de la technologie LoRa dans le monde IoT et avons détaillé son architecture ainsi que la manière de récupérer et de visualiser les données de géolocalisation en utilisant la méthode d'activations OTTA en combinant le protocole LoRaWAN avec la plateforme Cayenne.

Chapitre III. Tests et réalisation un prototype de géolocalisation

III.1 Introduction

Après une présentation approfondie de la technologie LoRa et du protocole LoRaWAN, nous pouvons continuer à présenter les différentes étapes de réalisation d'un prototype de géolocalisation.

Dans ce chapitre, nous allons étudier le choix de matériels pour la réalisation denotre solution. Ensuite, nous allons expliquer les tests, le principe de suivi et lesmesures assurant la localisation de notre objet connecté.

III.2 Matériels nécessaires

III.2.1 Choix de la carte de développement

Nous proposons sur le tableau 3 les caractéristiques techniques de modèles de cartes programmables disponibles sur le marché et en adéquation avec les exigences de notre projet.

Tableau 3 – Tableau Comparatif des cartes de développement Arduino ESP32, Raspberry Pi et STM32 :

Caractéristique	Arduino ESP32	Raspberry Pi	STM32
Processeur	Xtensa dual core	ARM Cortex	ARM Cortex
Fréquence	160 à 240 MHz	1,5 GHz	168 MHz
Mémoire flash	4/16/32 Mo	Jusqu'à 8 Go	512/1024/2048 Ko
RAM	512 Ko	1 à 8 Go	192 Ko
Prix	Abordable	Coûteux	Peu coûteux

Langages de programmation	C/C++, Micropython, Arduino IDE	Le système prend en charge plusieurs langages, notamment Python, C, C++, et bien d'autres.	C/C++, STM32CubeIDE,
Avantages **Avantages**	• Wifi et Bluetooth intégrés • Mémoire et rapidité suffisantes pour les tâches embarquées Abordable	• Beaucoup de RAM et mémoire flas • Système d'exploitation complet (Linux) Grande connectivité	• Très bas coût • Gamme étendue de microcontrôleurs Facile à utiliser pour les tâches embarquées

En se basant sur ce tableau comparatif (Tableau 3), nous optons pour la carte STM32 pour les raisons suivantes :

- **Performance élevée** : Les microcontrôleurs STM32 offrent de hautes performances avec des vitesses d'horloge allant jusqu'à 216 MHz. Ils ont une grande puissance de calcul par rapport aux Arduinos par exemple.
- **Faible consommation d'énergie:** Les STM32 consomment moins d'énergie que des cartes comme le Raspberry Pi. Ils sont donc idéaux pour les applications nécessitant une longue autonomie.
- **Grande mémoire :** Les STM32 disposent de quantités importantes de mémoire flash et RAM selon les modèles, allant jusqu'à 2 Mo de chaque. Cela permet de supporter des applications complexes.
- **Périphériques intégrés :** Les STM32 intègrent de nombreux périphériques tout-en-un comme des modules CAN, USB, ADC, etc. ce qui réduit le besoin de composants externes.
- **Faible coût :** Les cartes STM32 sont généralement moins chères que des cartes comme l'ESP32 ou l'Arduino. Leur rapport performance/prix est donc intéressant.

- **Flexibilité :** Les STM32 offrent une grande flexibilité dans la configuration matérielle des périphériques internes du microcontrôleur.
- **Large gamme :** ST propose une vaste gamme de cartes STM32 avec des caractéristiques variées répondant à la plupart des besoins.
- **Fiabilité :** Les microcontrôleurs STM32 sont réputés pour leur robustesse et leur fiabilité, ce qui est essentiel pour les applications critiques.

III.2.2 Carte de développement STM32 B-L072Z-LRWAN1

La carte B-L072Z-LRWAN1[7] présentée par la figure 25 est une carte électronique conçue par STMicroelectronics pour aider les ingénieurs et les développeurs à créer des applications basées sur la technologie LoRaWAN. La carte comporte un composant hybride SOC MURATA intégrant un microcontrôleur STM32l072CZY6TR avec le cœur RISC Arm Cortex-M0+ (32 bits) et un émetteur récepteur LoRa SX1276 qui offrent une connectivité longue portée et à faible consommation d'énergie pour les objets connectés (IoT).

Le microcontrôleur sur la carte fonctionne à une fréquence de 32 MHz et dispose de mémoires à haute vitesse tels que RAM de 20 Ko, EEPROM de 6 Ko et une mémoire programmable Flash 192 Ko. Elle supporte également des interfaces de communications telles qu'I2C, SPI, UART, LPUART, USB et intègre des périphériques GPIO, ADC et Timers.

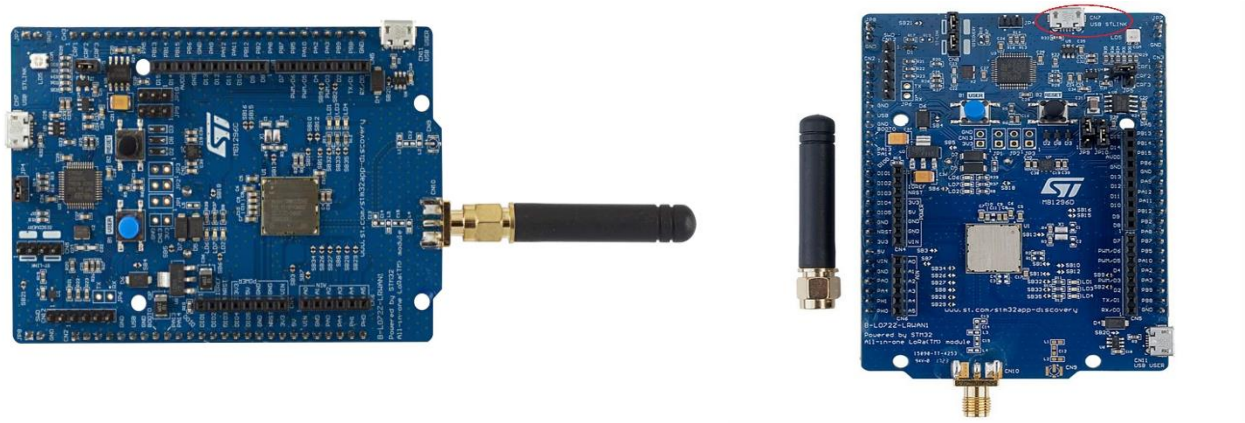


Figure 11 – Carte de développement STM32 B-L072Z-LRWAN1

La figure 26 suivante présente la carte de développement STM32 B-L072Z-LRWAN1, mettant en évidence ses différents composants et leurs fonctionnalités. Cette carte offre une alimentation flexible via une connexion USB ou une source externe. Elle intègre un module ST-Link/V2-1 facilitant la programmation et le débogage. Deux boutons-poussoirs et deux LED de couleurs différentes permettent une interaction utilisateur et l'indication de l'état du système. Des cavaliers de configuration sont disponibles pour une configuration aisée. Les connecteurs d'extension Arduino offrent un accès facile aux modules complémentaires. Enfin, la carte est équipée d'un connecteur d'antenne SMA pour le module LoRa Murata, assurant une communication sans fil fiable sur une bande de fréquence configurable (86 MHz à 930 MHz). Cet ensemble complet de fonctionnalités fait de la carte de développement STM32 B-L072Z-LRWAN1 une plateforme polyvalente et conviviale pour le développement et le prototypage de diverses applications.

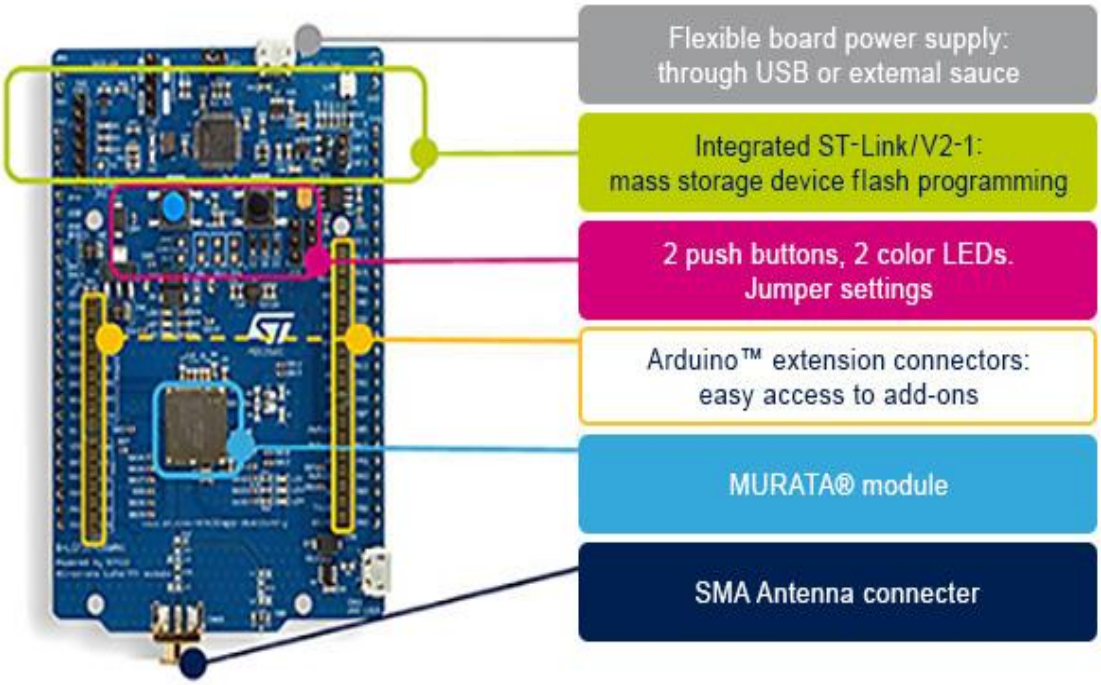


Figure 26 – Fonctionnalités de l' STM32 B-L072Z-LRWAN1

III.2.3 Shield Dargino LoRa/GPS

Les modules GPS offrent une solution technologique efficace pour suivre en temps réel la position des personnes atteintes de démence et les localiser en cas de besoin.

Pour notre projet, nous avons opté pour le module Shield Dragino LoRa/GPS (figure 27) qui est un choix de la société et qui présente certains avantages tels que :

- Une connectivité LoRa : Le module est équipé également d'une puce LoRa SX1278 – 868 MHz qui offre une communication sans fil à longue portée avec une consommation d'énergie réduite.
- Un GPS intégré : Le module intègre également un système de positionnement global (GPS) basé sur le 3339 SOCMT QUECTEL L80 permet d'obtenir des informations précises sur la localisation géographique avec une faible consommation d'énergie.

- **Le principe de fonctionnement du module GPS :**

Le module Shield Dragino LoRa/GPS est un module électronique développé par la société Dragino, spécialisée dans les solutions de connectivité IoT. Il combine deux technologies clés : LoRa pour la communication sans fil à longue portée et le GPS pour la localisation précise. Nous nous intéresserons seulement à la fonctionnalité GPS de ce module.

Le principe de fonctionnement du module est le suivant :

1. Le module Dragino LoRa/GPS se connecte à un microcontrôleur principal (comme un Arduino ou STM32) via une interface de communication série UART avec un débit de 9600 bauds.
2. Le module Dragino LoRa/GPS utilise les signaux GPS pour obtenir les coordonnées géographiques précises de l'endroit où il se trouve.
3. Le module GPS peut calculer et prédire automatiquement les orbites en utilisant les données d'éphémérides (jusqu'à 3 jours) stockées dans la mémoire flash interne. Ainsi, il peut fixer rapidement la position même à des niveaux de signal intérieurs.
4. Le module est capable de signaler jusqu'à 10 positions par seconde (10 Hz). Les protocoles pris en charge sont NMEA et MediaTek PMTK.

5. Les coordonnées GPS ainsi que d'autres informations telles que l'altitude, la vitesse et l'heure sont ensuite transmises au microcontrôleur principal.
6. Le microcontrôleur principal peut ensuite utiliser ces informations pour des applications telles que le suivi et la localisation en temps réel, la surveillance environnementale, la gestion des actifs, etc.

Le module doit être utilisé dans une zone accessible au public, offrant une visibilité directe des satellites pour garantir des résultats précis et corrects.

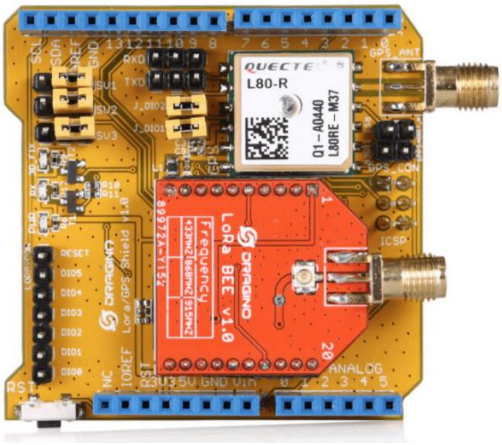


Figure 27 – Le module Shield Dragino LoRa/GPS

Le tableau 4 montre les caractéristiques dumodule GPS qui est soudé sur le Shield Dragino LoRa/GPS.

Tableau 4 – Caractéristiques du Shield Dragino LoRa/GPS :

Nom	Lora / GPS Shield
Alimentation	2.7-3.6 V (nominale 3 V)
Consommation d'énergie	Mode normal : 37 mA- 3V
Précision du signal	10ns
Précision de la vitesse	0.1 ms
Temps de ré acquisition	<1s
Communication	UART
dimension	10 x 5 x 2 cm

les trames de données envoyées par le GPS sont de la forme de phrases NMEA standardisées qui contiennent des informations sur la position, la vitesse, l'heure et d'autres données utiles (Figure 28).

Le format de trame NMEA standard est constitué d'un en-tête de ligne commençantpar le caractère "$", suivi d'un code de type de message à trois lettres, suivi desdonnées du message et d'une séquence de contrôle de checksum.

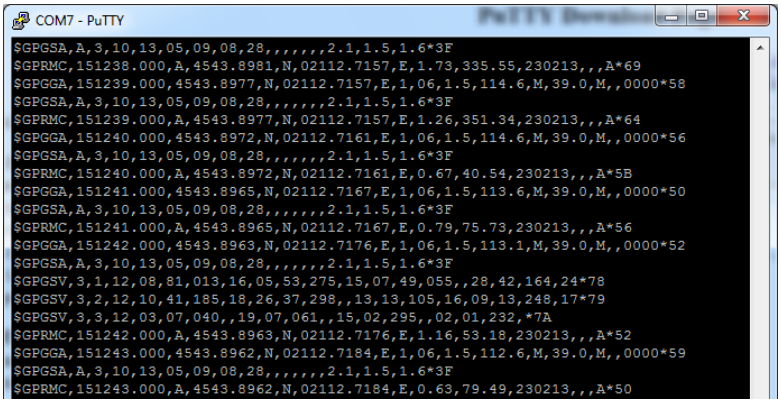


Figure 28 – Exemple d'une instance NMEA

III.2.4 Un analyseur logique

Un analyseur logique est un instrument de mesure électronique permettant de visualiser les signaux logiques sur des circuits numériques au cours du temps (Figure 29). Il possède également des interpréteurs de protocoles pour analyser plusieurs bus série tels que UART, SPI, I2C et CAN.

Figure 29 – Analyseur logique Saleae

III.3 Logiciels utilisés

III.3.1 Choix de l'environnement de développement

Le choix de l'environnement de développement est une étape importante dans le développement de tout projet électronique.

Pour le STM32 B-L072Z-LRWAN1, plusieurs logiciels sont disponibles pour faciliter le développement. Nous présentons un tableau comparatif des différents environnements de développement adaptés au STM32 B-L072Z-LRWAN1 (tableau 5).

Tableau 5 – Tableau Comparatif des environnements de développement pour STM32

Environnement de développement	Plateformes (OS)	Avantages	Inconvénients
STM32CubeIDE	Windows, Linux, macOS	• Interface graphique intuitive, • Prise en charge de plusieurs langages de programmation, • Bibliothèques complètes, • Générateur de code • Débogage avancé	• Peut être lourd pour les systèmes plus anciens
Keil µVision	Windows	• Environnement de développement intégré complet • Support multiplateforme	• Coût de la licence élevé
TrueSTUDIO	Windows, Linux, macOS	• Gratuit • Support multiplateforme • Intégration avec Git	• Pas autant de fonctionnalités que d'autres logiciels

Bien que tous ces logiciels offrent des fonctionnalités de développement pour le STM32 B-L072Z-LRWAN1, le STM32CubeIDE se distingue par son interface graphique facile à comprendre, son support de plusieurs langages de programmation, ses bibliothèques complètes et son générateur de code.

De plus, il offre un débogage avancé pour faciliter le développement et la résolution des problèmes de l'application.

Pour cette raison nous avons opté dans le développement de notre application pour cet outil.

III.3.2 STM32CubeMX

STM32CubeMX est un outil graphique, qui permet une configuration simpleet facile des microcontrôleurs, microprocesseur STM32. Son logo est présenté par la figure 30 . À travers des initialisations correspondantes, il génère un projet en langage C valable pour plusieurs IDE tels que kiel µVision, IAR, Atollic, STM32Cube IDE.

Figure 30 – Logo CubeMX

III.3.3 STM32CubeIDE

STM32CubeIDE est une plateforme de développement C/C++ avancée, elledispose d'un ensemble de fonctionnalités de configuration des périphériques, de génération de code, de compilation et de débogage de codes pour les microcontrôleurs et les microprocesseurs STM32. Elle est totalement gratuite puisqu'il s'agissait d'une plateforme développée par ST, contrairement à Keil µVision qui nécessitel'achat d'une licence pour pouvoir communiquer des données de grande taille. De plus, elle offre des vues avancées pour contrôler et modifier le contenu des adressesmémoire. La figure 31 représente son logo.

Figure 31 – Logo STM32 CubeIDE

III.3.4 PuTTY

PuTTy est un programme de communication open source et gratuit.Il intègre aussi une émulation terminale qui supporte les connexions sur les ports séries et les connexions en TCP/IP, généralement utilisées pour connecter les systèmes embarqués au PC pour collecter des données. Son logo est présenté par la figure 32 .

Figure 32 – Logo PuTTy

III.4 Installation de la bibliothèque I-CUBE-LRWAN

Pour transmettre des données d'un dispositif LoRa au serveur LoRaWAN dela société Treetronix, la première étape effectuée est l'installation de la biblio- thèque I-CUBE-LRWAN développée par STMicroelectronics, disponible sur leur site comme indique la figure 33 . Elle est compatible avec les microcontrôleursdes séries STM32L0, STM32L1 et STM32L4 et elle prend en charge les cartes d'extension radio Semtech SX1276MB1MAS, SX1276MB1LAS, SX1272MB2DASet SX1262DVK1DAS.

Part Number	General Description	Supplier	Latest version	Download
I-CUBE-LRWAN	LoRaWAN software expansion for STM32Cube (UM2073)	ST	2.1.0	Get latest

Figure 33 – Package d'extension I-CUBE-LRWAN

III.5 Configuration de la bibliothèque LoRaWAN

L'étape de configuration est une phase préliminaire dans le développement de notre application, nous cherchons toujours à minimiser la consommation d'énergie,à répondre aux besoins de l'application et à augmenter la sécurité des informationstransmises,

pour cela, nous allons définir le mode d'activation et la classe de notre dispositif.

La configuration de la classe et de la méthode d'activation se fait dans le fichier Lora_app.h . Pour définir la classe de l'équipement nous utilisant la variable LO-RAWAN_DEFFAULT_CLASS, puis la classe choisie est CLASS_A, CLASS_B ou CLASS_C. Et pour définir la méthode d'activation nous utilisant la variable LORAWAN_DEFAUT_ACTIVATION_TYPE avec la méthode d'activation choisie ACTIVATION_TYPE_ABP ou ACTIVATION_TYPE_OTAA.

Dans notre application, nous avons choisi la classe A et la méthode d'activationOTAA. La figure 34 présente la configuration de notre dispositif.

```
lora_app.h
70 #define LORAWAN_DEFAULT_CLASS                       CLASS_A
71
72 /*!
73  * LoRaWAN default confirm state
74  */
75 #define LORAWAN_DEFAULT_CONFIRMED_MSG_STATE         LORAMAC_HANDLER_UNCONFIRMED_MSG
76
77 /*!
78  * LoRaWAN Adaptive Data Rate
79  * @note Please note that when ADR is enabled the end-device should be static
80  */
81 #define LORAWAN_ADR_STATE                           LORAMAC_HANDLER_ADR_ON
82
83 /*!
84  * LoRaWAN Default data Rate Data Rate
85  * @note Please note that LORAWAN_DEFAULT_DATA_RATE is used only when LORAWAN_ADR_STATE is disabled
86  */
87 #define LORAWAN_DEFAULT_DATA_RATE                   DR_0
88
89 /*!
90  * LoRaWAN default activation type
91  */
92 #define LORAWAN_DEFAULT_ACTIVATION_TYPE             ACTIVATION_TYPE_OTAA
93
94 /*!
95  * User application data buffer size
96  */
97 #define LORAWAN_APP_DATA_BUFFER_MAX_SIZE            242
98
99 /*!
100  * Default Unicast ping slots periodicity
```

Figure 34 – Configuration d'un dispositif LoRa

La configuration des paramètres de méthode d'activation s'effectue dans le fichier se-identitity.h. Puisque nous avons choisi la méthode d'activation OTAA, il fautdéfinir les clés DevEUI, AppEUI et AppKey qui sont toujours confidentielles.

Chaque paramètre admet une signification DevEUI qui est l'unique identifiée pour le diapositif LoRa. AppEUI est l'unique identifiée pour l'application serveur et AppKey sert à générer le MIC lors le Joint-Request. Les paramètres correspondantsà ses clés dans le fichier se-identity.h sont DevEUI : LORAWAN_DEVICE_EUI, AppEUI : LORAWAN_JOIN_EUI et AppKey : LORAWAN_APP_KEY.

Ils sont identifiés dans la figure 35.

```
se-identity.h
93  #define STATIC_DEVICE_EUI                                1
94
95  /*!
96   * end-device IEEE EUI (big endian)
97   */
98  #define LORAWAN_DEVICE_EUI                               { 0x00, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0x00 }
99
100 /*!
101  * App/Join server IEEE EUI (big endian)
102  */
103 #define LORAWAN_JOIN_EUI                                 { 0x00, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0xBB, 0x00 }
104
105 /*!
106  * When set to 1 DevAddr is LORAWAN_DEVICE_ADDRESS
107  * When set to 0 DevAddr is automatically set with a value provided by a pseudo
108  *       random generator seeded with a value provided by the MCU platform
109  */
110 #define STATIC_DEVICE_ADDRESS                            1
111
112 /*!
113  * Device address on the network (big endian)
114  */
115 #define LORAWAN_DEVICE_ADDRESS                           ( uint32_t )0x88888888
116
117 /*!
118  * Application root key
119  */
120 #define LORAWAN_APP_KEY                                  00,BB,BB,BB,BB,BB,BB,BB,BB,BB,BB,BB,BB,BB,BB,00
121
122 /*!
123  * Network root key
```

Figure 35 – Configuration des paramètres OTAA

Tableau 6 -Tableau de définition des clés d'activation

Clé D'activation	Définition	Paramètres respectifs à ses clés
DevEUI	C'est un identifiant unique pour le module LoRa qu'on a utilisé. Il est de 64 bits.	LORAWAN_DEVICE_EUI
AppEUI	C'est un identifiant unique qui caractérise le serveur de jointure. Il est codé sur 64 bits.	LORAWAN_JOIN_EUI
AppKey	C'est la clé utilisée pour les messages à chaque activation. Elle est spécifique à l'appareil	LORAWAN_APP_KEY

Le tableau 6 résume les principaux paramètres du mode d'activation ainsi que les configurations que nous avons considérées.

Tableau 7 - Configuration des paramètres du mode d'activation

Paramètre	Valeur	Description
Mode débogage	1	Active le débogage
DevEUI, AppEUI, AppKEY	0xBB	Identifiants simplistes pour le test du prototype
OTAA	1	Active l'OTAA pour rejoindre dynamiquement le réseau
Région	EU868	Région Europe/Afrique du Sud (pour LoRaWan)
Fréquence	868000000	Fréquence d'émission dans la bande 868 MHz

Une fois que tous les paramètres nécessaires ont été configurés, nous pouvons établir en toute sécurité une connexion à notre passerelle et ainsi qu'au serveur du réseau LoRa 3S.

III.6 Transmission d'un simple message vers le serveur LoRaWAN

Avant de détailler nos étapes de travail et de développement, nous avons d'abord envoyé un simple message pour vérifier le bon fonctionnement et l'acquisition des données de notre carte STM32 B-L072Z-LRWAN1 afin qu'il puisse joindre le serveur LoRaWAN de la société correctement.

La figure 36 présente l'organigramme de l'envoi d'un message par le protocole LoRaWAN.

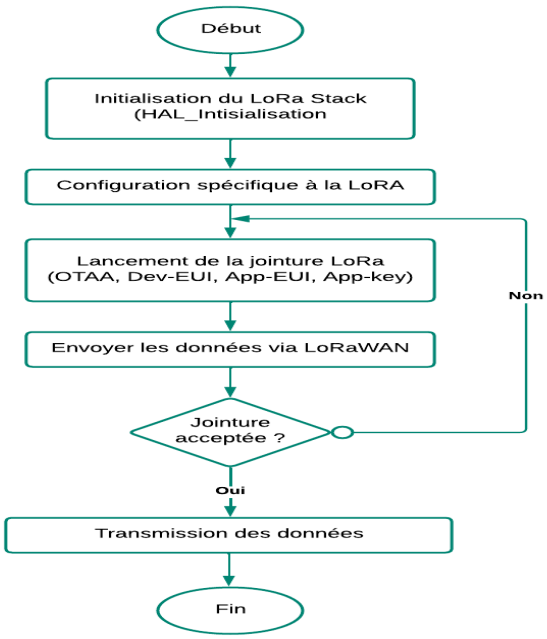


Figure 36 – Organigramme de l'envoie des données

Après la configuration du programme illustré par la figure 37 et le chargement du code dans notre carte stm32 à l'aide de ST-Link, le module LoRa SX1272 envoieune demande de jointure vers le network serveur passant par le Gateway LoRaWAN qui capte le message et l'envoie vers le serveur puis ce dernier vérifie l'identifiant AppEUI et s'il y a validation, on reçoit un message d'acceptation de la jointure.

```
se-identity.h   lora_app.h   lora_app.c   LmHandlerTypes.h   lora_app.c X
383 static void SendTxData(void)
384 {
385   /* USER CODE BEGIN SendTxData_1 */
386   UTIL_TIMER_Time_t nextTxIn = 0;
387   AppData.Port = LORAWAN_USER_APP_PORT;
388   char *msg= "hello LoRaWAN !";
389 //  uint32_t i = 0;
390   sprintf((char *)AppData.Buffer, msg);
391
392   AppData.BufferSize = strlen(msg);
393
394   if (LORAMAC_HANDLER_SUCCESS == LmHandlerSend(&AppData, LORAWAN_DEFAULT_CONFIRMED_MSG_STATE, &nextTxIn, false))
395   {
396     APP_LOG(TS_ON, VLEVEL_L, "SEND REQUEST\r\n");
397   }
398   else if (nextTxIn > 0)
399   {
400     APP_LOG(TS_ON, VLEVEL_L, "Next Tx in  : ~%d second(s)\r\n", (nextTxIn / 1000));
401   }
402
403   /* USER CODE END SendTxData_1 */
404 }
```

Figure 37 – Fonction permettant d'envoyer une trame LoRaWAN

À l'aide de la configuration du logiciel PuTTY comme le montre la figure 38 on peut visualiser l'acceptation de la jointure.

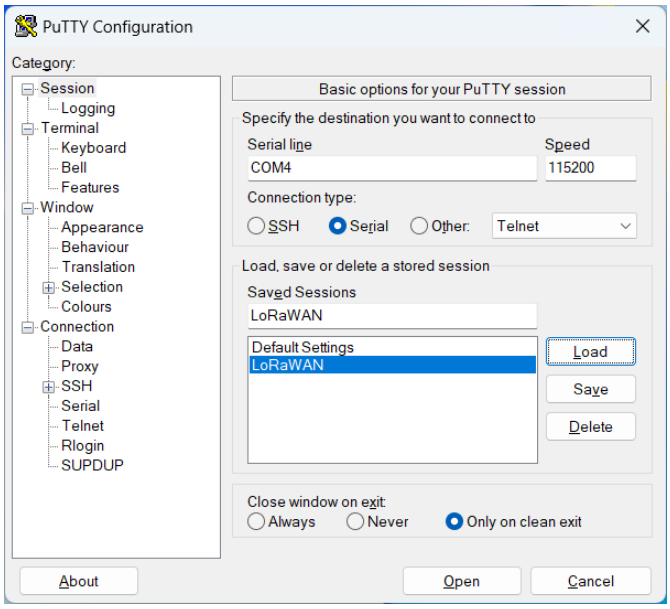


Figure 38 – Configuration du PuTTY

Pour visualiser l'acceptation de jointure nous utilisons le terminal PuTTY qui est un émulateur de terminal, une console série et une application de transfert de fichier réseau gratuite et open-source comme le montre la figure 39.

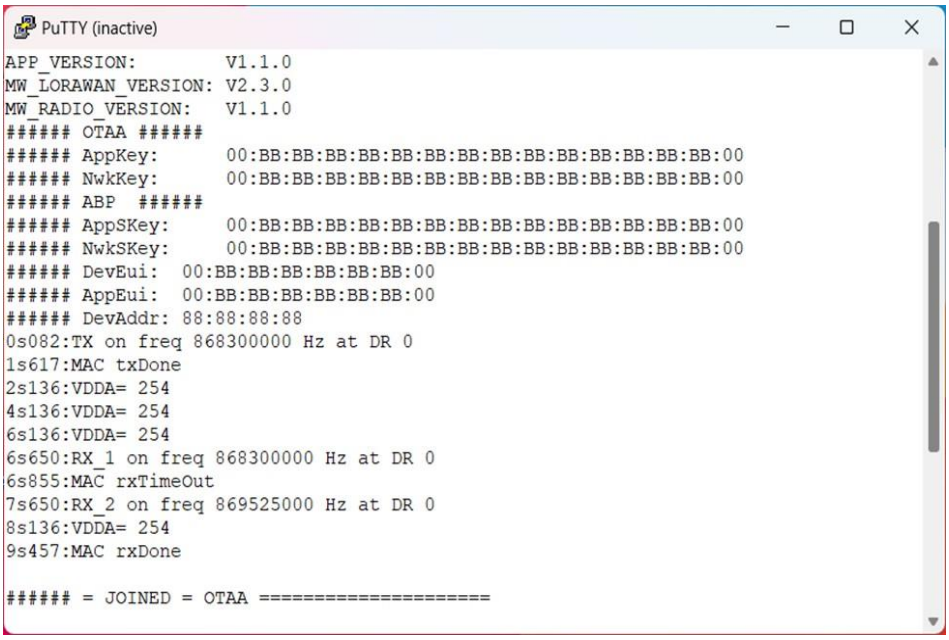


Figure 12– Acceptation de la jointure OTAA et l'envoi d'une donnée au serveur LoRaWAN

III.7 Visualisation de la sortie sur le serveur LoRaWAN

Nous avons réussi à envoyer un simple message au serveur LoRaWAN, le messageenvoyé arrive au serveur crypté en hexadécimale comme le montre la figure 40.

Figure 13 – Message transmis au serveur

Pour obtenir le message souhaité, nous devons convertir le message de l'hexadécimale au type déclaré dans le programme de notre application, le type déclaré est texte.

La figure 41 représente la conversion du message transmis au serveur.

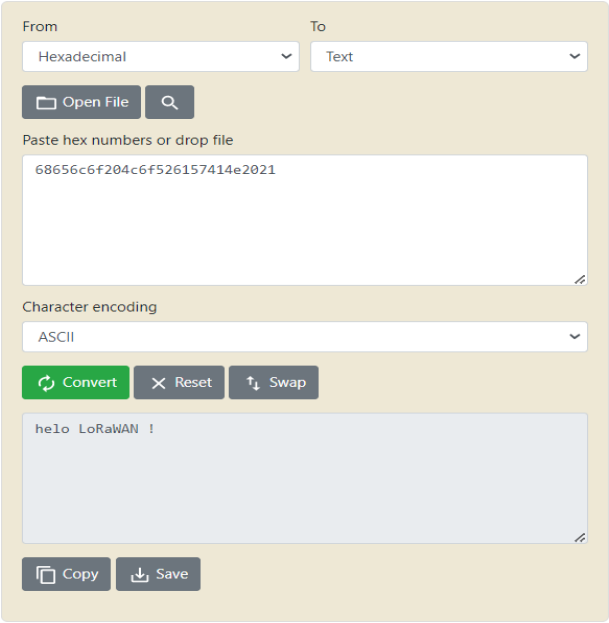


Figure 14 – Conversion du message

III.8 Programmation du module de géolocalisation

D'abord, nous avons connecté le module Shield Dragino LoRa/GPS à notre carte STM32 B-L072Z-LRWAN1 via une communication série UART (une description détaillée des protocoles de communication série est donnée sur l'Annexe 5). Le tableau 8 ci-dessous présente les broches de connexion utilisées.

Tableau 8 -Broches de connexion entre le GPS L70 la carte STM32

Shield Dragino LoRa/GPS	STM32B-L072RZ
1 (RX)	PB7 (USART1_TX)
2 (TX)	PB6 (USART1_RX)

La figure 43 montre une photo du câblage entre le module Shield Dragino LoRa/GPS et la carte STM32 B-L072Z-LRWAN1 :

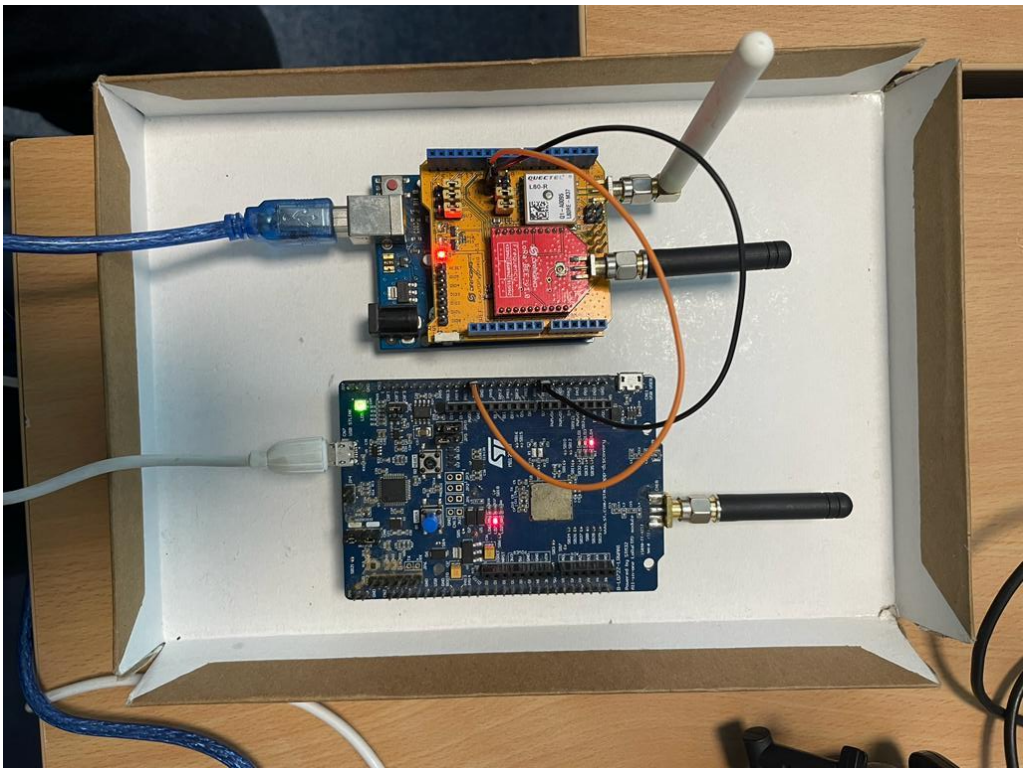


Figure 15 - Câblage du module GPS

La trame NMEA contient beaucoup de séquences de données comme GPRMC, GPGSA, GPGLL. Le test qu'on va faire sera sur GPGGA d'où on va extraire les coordonnées (latitude, longitude) du module GPS (Figure 44).

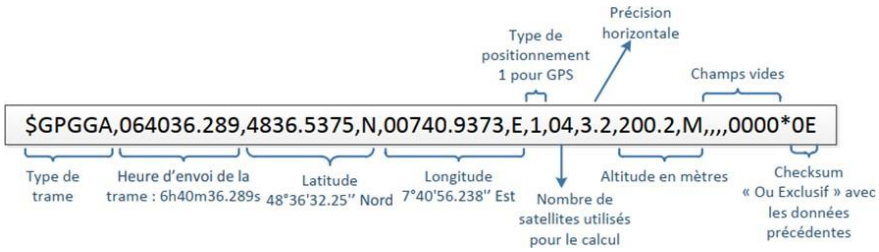


Figure 16 – Example d'une séquence GGA

L'organigramme présenté par la figure 45 montre l'obtention des coordonnéesGPS à partir de la trame reçue :

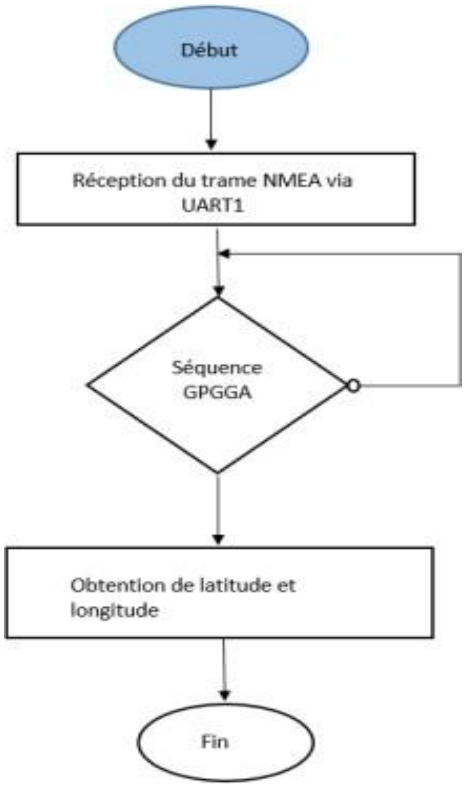


Figure 17– L'organigramme d'extraction des coordonnées GPS

Nous avons reçu une trame de données, à l'aide d'une interruption sur la connexion UART1 qui est configurée dans notre programme pour stocker les caractères de la trame dans un Buffer de données :

La fonction présenté par la figure 46 :

```
HAL UART Receive IT(UART HandleTypeDef *huart, uint8 t *pData, uint16 t Size)
```

Figure 18 – La fonction HAL_UART_Receive_IT

On a :

- *huart : l'adresse de l'uart utilisé ;
- *pData : l'adresse ou an va stocker le caractère reçu ;
- Size : égal à 1 implique un seul caractère ;

La figure 47 illustre le diagramme d'interruption de réception des données duGPS :

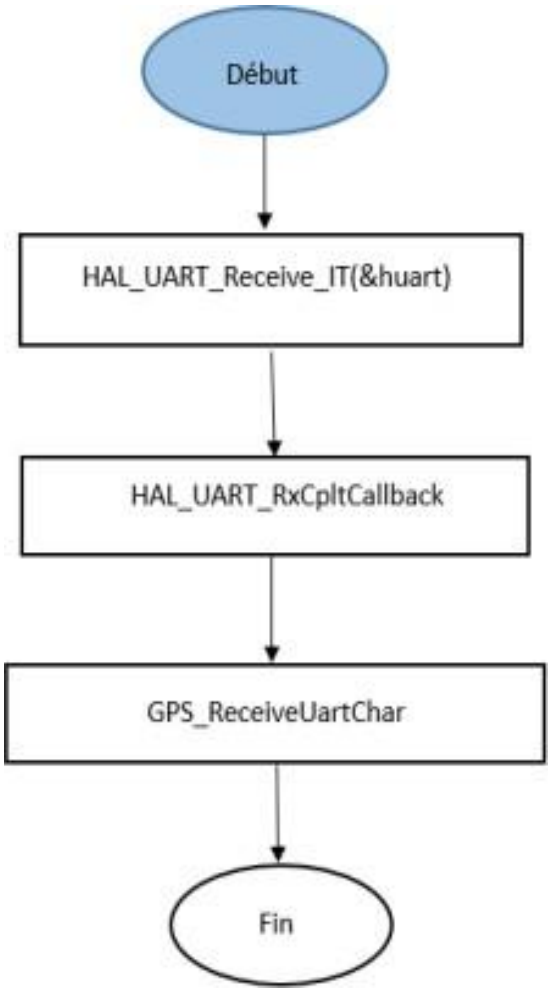


Figure 47 – La fonction HAL_UART_Receive_IT

Avant d'envoyer les données de localisation au serveur LoRaWAN, on peut avec le Debug mode du CubeIDE les visualiser comme ils sont montrés par la figure 48 .

Variables | Breakpoints | Expressions | Registers | Live Expressions | SFRs

Expression	Type	Value
GPS	GPS_t	{...}
dec_longitude	float	10.1354113
dec_latitude	float	36.8377228
altitude_ft	float	0
nmea_longitude	float	1008.12469
nmea_latitude	float	3650.26343
utc_time	float	103541
ns	char	78 'N'
ew	char	69 'E'
lock	int	1
satelites	int	4
hdop	float	1.82000005
msl_altitude	float	90.6999969
msl_units	char	77 'M'
rmc_status	char	0 '\000'
speed_k	float	0
course_d	float	0
date	int	0
gll_status	char	65 'A'
course_t	float	132.660004
course_t_unit	char	84 'T'
course_m	float	0
course_m_unit	char	0 '\000'
speed_k_unit	char	0 '\000'
speed_km	float	0

Figure 48 – Visualisation des données GPS

Ensuite, nous envoyons les données de localisation au serveur LoRaWAN tout les en visualisant sur le terminal PuTTY (Figure 49)

COM4 - PuTTY

```
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854458 Longitude: 10.209654
Latitude: 36.854465 Longitude: 10.209652
Latitude: 36.854469 Longitude: 10.209650
Latitude: 36.854469 Longitude: 10.209650
Latitude: 36.854469 Longitude: 10.209650
Latitude: 36.854469 Longitude: 10.209650
Latitude: 36.854469 Longitude: 10.209650
Latitude: 36.854469 Longitude: 10.209646
Latitude: 36.854469 Longitude: 10.209646
Latitude: 36.854469 Longitude: 10.209646
Latitude: 36.854469 Longitude: 10.209646
Latitude: 36.854465 Longitude: 10.209643
Latitude: 36.854465 Longitude: 10.209643
Latitude: 36.854465 Longitude: 10.209643
Latitude: 36.854465 Longitude: 10.209643
```

Figure 49– Visualisation des données GPS dans le terminal PuTTY

En entrant les valeurs de la longitude et la latitude dans « Google Maps », nous obtenant notre adresse qui correspond à notre emplacement l'entreprise Treetronix comme l'indique la figure 50.

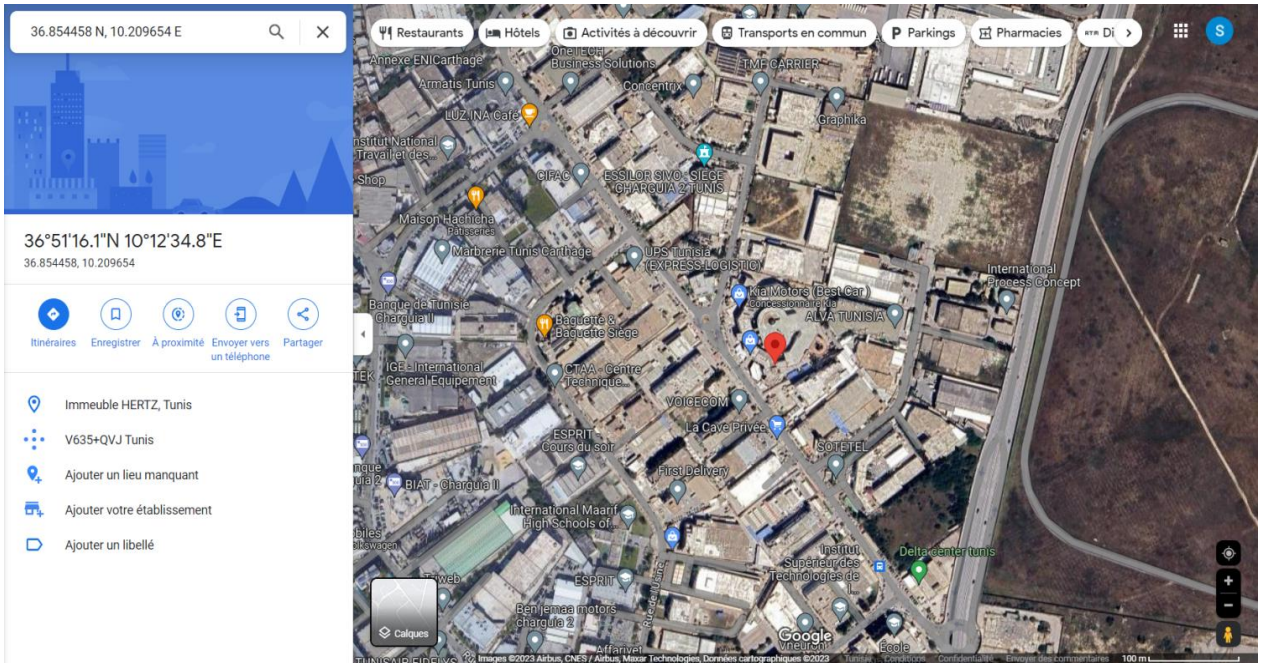


Figure 50– Affichage de la localisation dans Google Maps

Une fois les données transmises, nous pouvons utiliser la plateforme Cayennepour visualiser et analyser les informations collectées.

III.9 Conclusion

Dans ce chapitre, on a expliqué la configuration du programme et le câblage du matériel nécessaire retenus pour notre solution. Puis, nous avons présenté quelques résultats de localisation. ces résultats sont disponibles sur la plateforme Cayenne.

Chapitre IV. Conception d'une carte électronique de géolocalisation

IV.1 Introduction

Dans ce chapitre, nous allons décrire les différentes étapes de conception d'une carte de géolocalisation en utilisant les mêmes modules précédemment utilisés (Module GPS L70, Microcontrôleur STM32L072CZ) tout enajoutant des fonctionnalités supplémentaires. Nous allons explorer les différentes étapes de conception de la carte et de la création du schéma électrique. Enfin, nous terminerons ce chapitre par une description des tests effectués pour vérifier que le circuit imprimé fonctionne correctement et transmet les données de géolocalisation à la plateforme Cayenne.

IV.2 Conception sur ordinateur

Il existe de nombreux logiciels de CAO qui nous permettent de réaliser notreschéma électrique (Logo dans la figure 51). À cette étape, on a utilisé l'Eagle (Easily Applicable Graphical Layout Editor) qu'il comprend un éditeur de schémas, un outil de routage du circuitimprimé et un compilateur des erreurs. Après vérification, le schéma final se trouvedans l'ANNEXE 6,7,8 et 9.

Figure 19 – Logo Eagle

IV.3 Réalisation du schéma électrique

La première étape de la conception d'une carte électronique consiste à assemblertous les Composants dans des schémas qui incluent les différents blocs sur lesquels nous allons travailler afin d'assurer la fonctionnalité du système de géolocalisation d'objets connectés sur le réseau LoRaWAN qui sont mentionnés précédemment :

- Mesurer la température, les mouvements et les rotations d'un objet dans l'espace.
- Déterminer la position géographique en temps réel d'un objet.
- Transmettre des données de géolocalisation à distance via une liaison sans fil basée sur la technologie LoRaWAN.

D'autres blocs seront ajoutés au schéma électrique qui sont nécessaires dans notreprojet, tels que :

- Charger directement une batterie à partir d'un port USB.
- Fournir des informations visuelles sur l'état du circuit en utilisant 3 LED d'indi-cation.
- Changer la période du uplink vers le serveur LoRaWAN qui est par défaut 10 secondes en utilisant un bouton poussoir.

Dans les prochaines parties de ce chapitre, nous allons examiner chaque unité duschéma et expliquer la nécessité et l'importance de chaque partie.

IV.3.1 Microcontrôleur STM32L072CZ

Pour chaque système embarqué, il est nécessaire d'avoir une unité pour le traitement des données. Pour faire le bon le choix du microcontrôleur, il faut avoir des connaissances sur les performances techniques et la facilité de mise en œuvre. Ce choix a été motivé par les performances élevées et la faible consommation d'énergie de ce microcontrôleur qui correspondent aux exigences de notre projet.

Pour la réalisation du schéma électrique de notre circuit de commande, nous avons commencé par l'alimentation du microcontrôleur STM32L072 avec la tension 3.3V sur tous les pins d'alimentation des bus d'entrée/sortie (VDDA VSSA ; VDD1 VSS1 ; VDD2 VSS2 ; VDD3 VSS3).

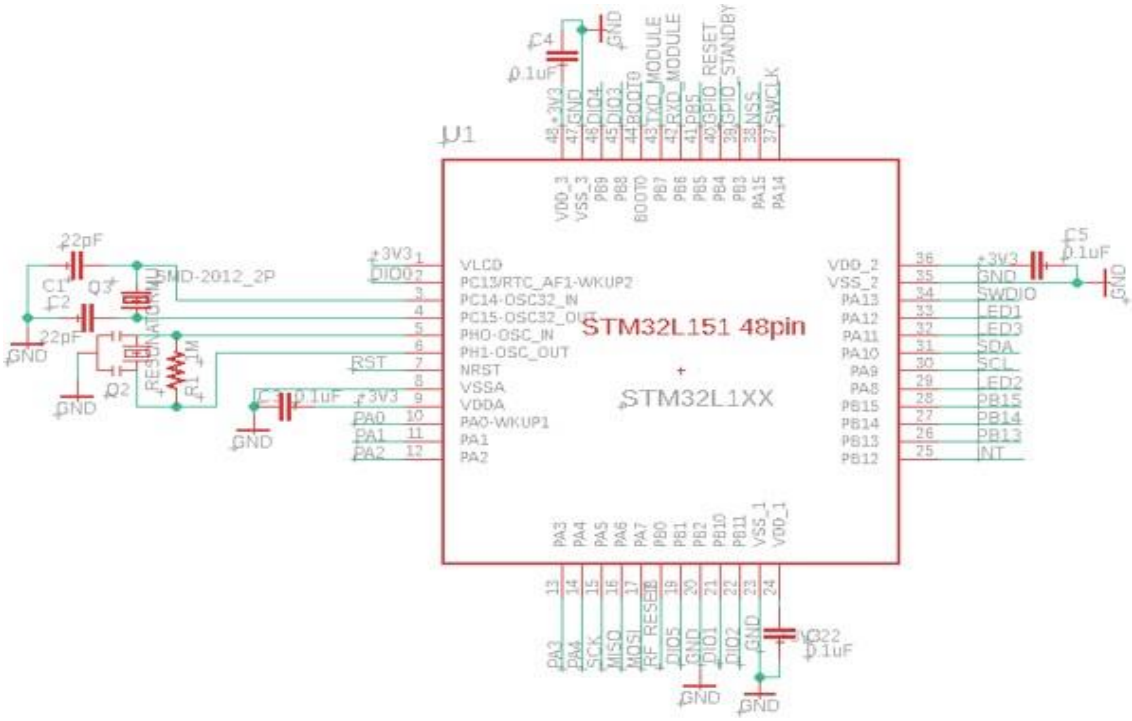


Figure 20 – Schéma de câblage du microcontrôleur STM32L072CZ

Afin de filtrer les ondulations de tension à haute fréquence, nous avons ajouté, entre chaque alimentation, une capacité céramique de 0.1 µF. Ces capacités sont placées très proche au microcontrôleur pour assurer une stabilité maximale de la tension d'alimentation du microcontrôleur.

Après, nous avons connecté le circuit d'oscillation (Q2 et Q3) qui sont utilisés afin de générer le signal d'horloge du microcontrôleur.

IV.3.2 Module LoRa RFM95

Le module LoRa RFM95 joue un rôle important dans notre projet, car il permet de transmettre les données de géolocalisation et de mouvement collectées par les modules GPS L70. Pour assurer cette transmission, le module RFM95 utilise la technologie de modulation de fréquence LoRa qui lui permet de communiquer avec les Gateways du réseau LoRaWAN.

La communication entre le module RFM95 et notre microcontrôleur STM32L072CZse fait via le bus SPI à l'aide de quatre lignes : MOSI, MISO, SCK et NSS.

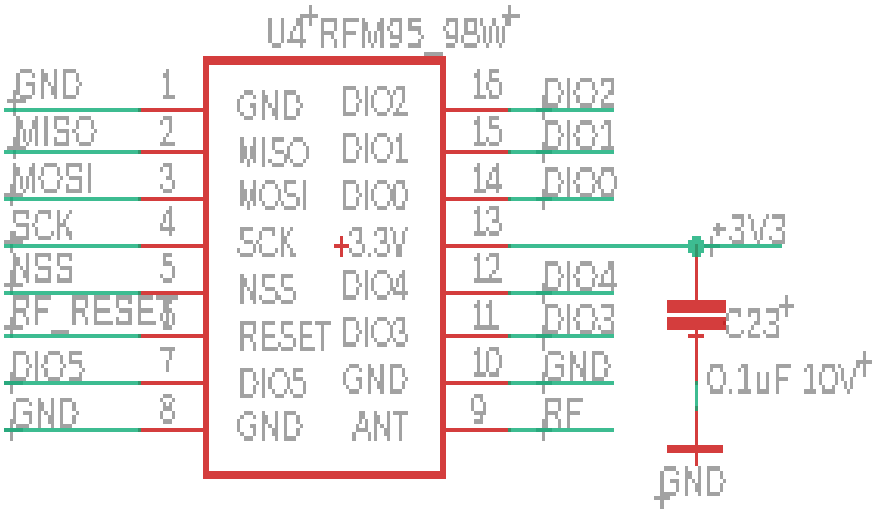


Figure 21 – Schéma de câblage du RFM95

IV.3.3 Le module RT8059

Le module RT8059 est un convertisseur de tension DC/DC. Dans notre cas, ilconvertit la tension de la batterie de 5 V en une tension de 3,3 V pour alimenter le microcontrôleur STM32L072CZ et les autres modules du système.

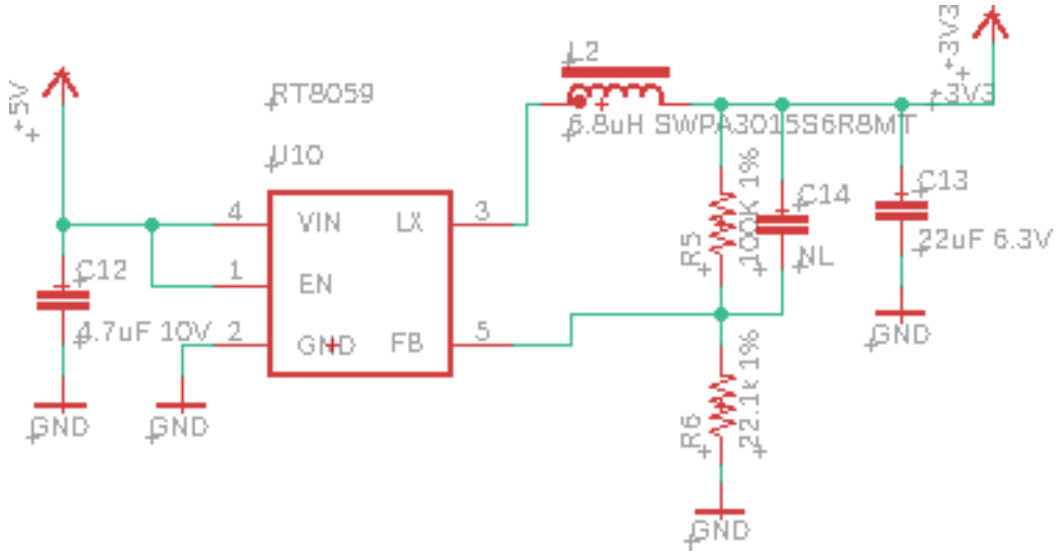


Figure 22 – Schéma de câblage du RT8059

IV.3.4 Le module CN3052A

Pour la charge de la batterie, nous avons utilisé un module de charge CN3052A qui est capable de charger des batteries lithium-ion avec une tension d'entrée allantjusqu'à 5 V à partir d'une source d'alimentation USB. Il possède également desfonctions de protection telles que la protection contre les surintensités, la protectioncontre les surtensions et la protection contre les courts-circuits.

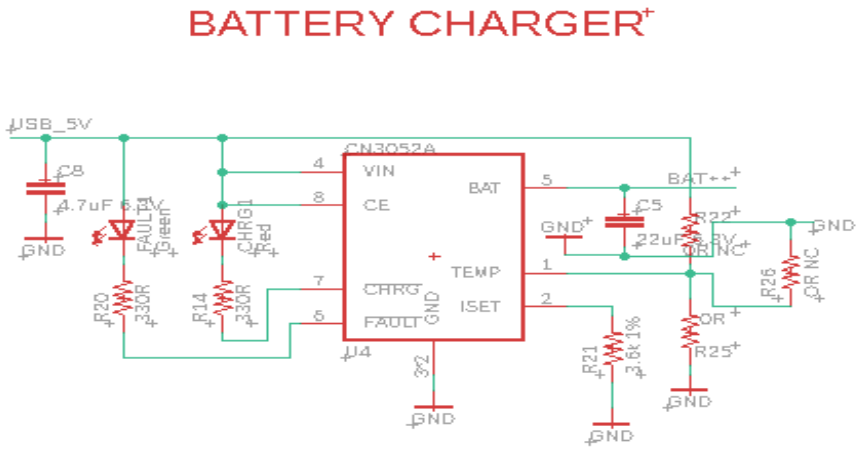


Figure 23– Schéma de câblage du CN3052A

IV.3.5 Connecteur USB micro B

Dans notre schéma électrique, nous avons utilisé un connecteur USB micro B pour deux fonctions principales. La première est la recharge de la batterie à travers un module de charge CN3052A. La deuxième fonction est le flashage dumicrocontrôleur STM32L072CZ à l'aide de l'interface SWDIO/SWCLK, permettantainsi le débogage de notre application embarquée.

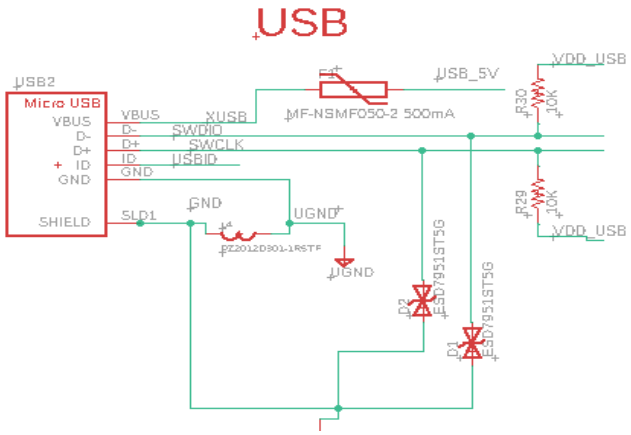


Figure 24– Schéma de câblage du Connecteur USB micro B

Deux diodes sont utilisées avec le connecteur USB micro B de type TVS (diodes de suppression de tension transitoire) . Leur rôle est de protéger le circuit contre les surtensions et les décharges électrostatiques qui peuvent se produirelors de la connexion ou de la déconnexion de l'USB. Ces diodes sont placées en parallèle avec les broches de données de l'USB pour détourner les surtensions et lesdécharges électrostatiques loin du circuit. Cela permet de protéger les composants électroniques sensibles tels que le microcontrôleur et les circuits de charge de la batterie contre les dommages.

IV.3.6 Module GPS L70

Le cœur de notre carte électronique est le module GPS L70 en collectant les données de géolocalisation, qui sont ensuite transmises via le module LoRa RFM95 aux passerelles LoRaWAN du réseau.

Cela nous permet de surveiller l'emplacementde l'appareil en temps réel.

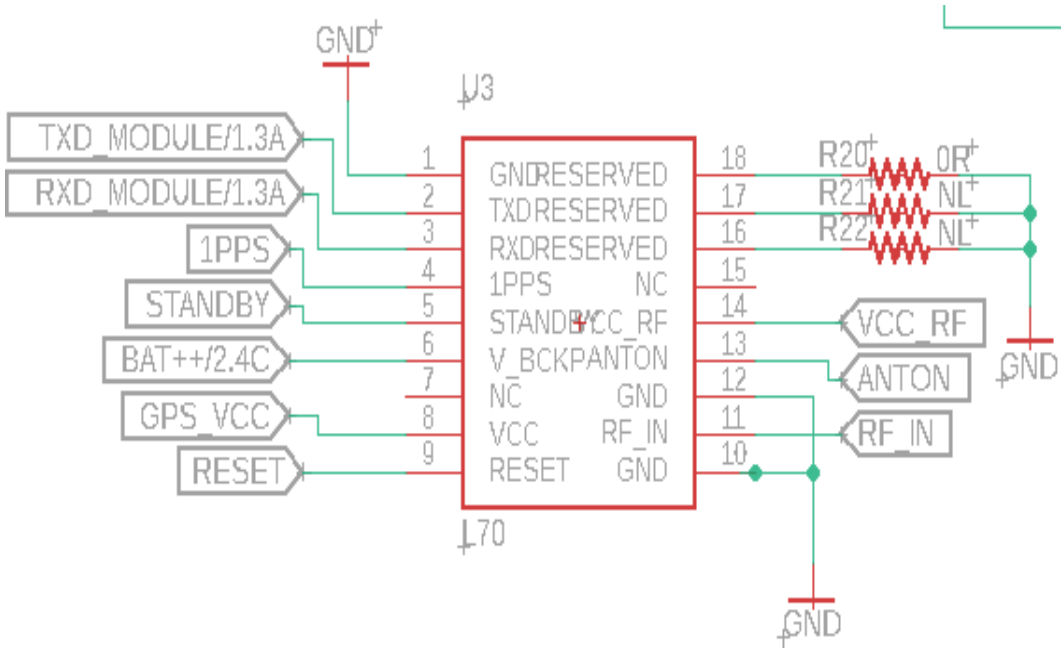


Figure 25 – Schéma de câblage du Module GPS L70

IV.4 Routage du schéma électrique

Après avoir terminé les schémas, nous sommes passés au routage du circuit (Figure 57) .C'était une tâche assez difficile et complexe, car le circuit n'était pas simple etla carte mesurait 7x4 cm avec deux couches : la couche inférieure et la couche supérieure. Pour faciliter le routage, nous avons ajouté un plan de masse. Cependant, nous avons rencontré un problème avec certaines bibliothèques de composants qui n'étaient pas disponibles sur Eagle. Comme solution, nous avons créé leur bibliothèque à partir de zéro en nous basant sur leurs dimensions.

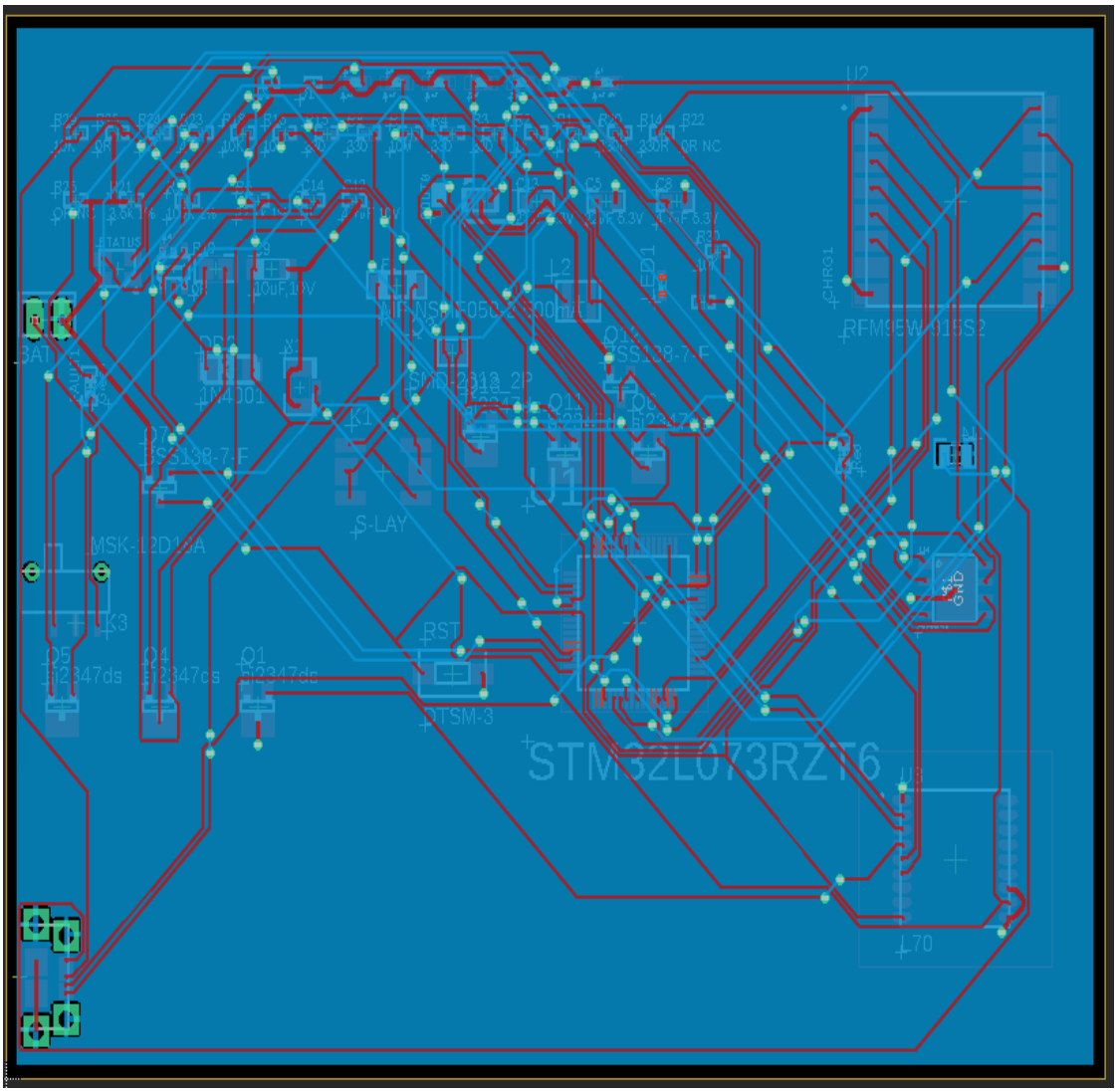


Figure 58 – Routage de la carte avec un plan de masse

La figure 59 montre la routage de la carte avec un plan de masse .

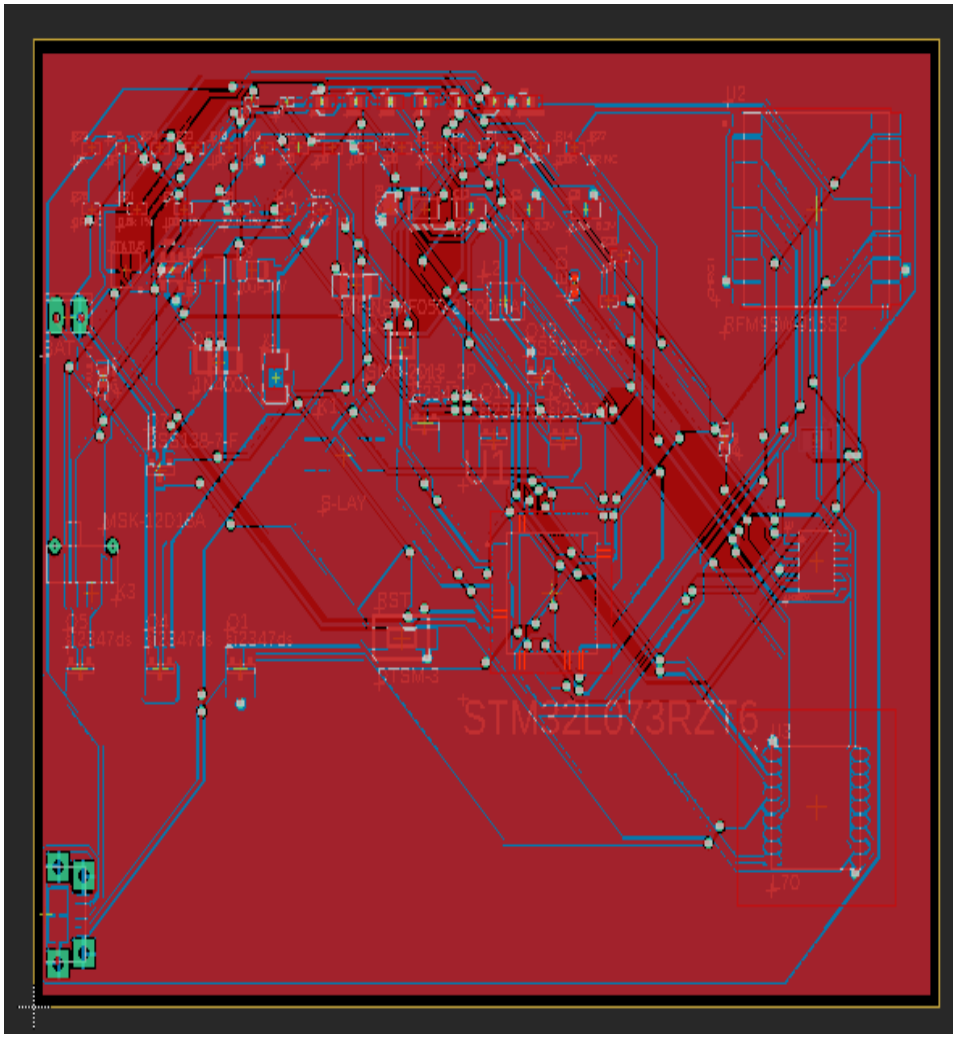


Figure 59– Routage de la carte avec un plan de masse

IV.5 Phase de test

Une fois que le routage a été terminé, nous avons envoyé à l'entreprise les fichiers de production au format Gerber. Ces fichiers comprennent les informationsnécessaires pour la fabrication de la carte électronique, telles que la disposition descomposants, les pistes de cuivre, les plans de masse et les trous de fixation .

La figure 60 montre la carte électronique finale.

Figure 60– La carte électronique finale après le soudage des composants

Pour flasher le microcontrôleur STM32L072CZ présent sur notre carte électronique finale, nous devons établir une connexion entre le connecteur SWD (CN12) de la carte B-L072Z-LRWAN1 et notre carte via un câble USB. Pour cela, nous retirons les jumpers du connecteur 8 (CN8) de la carte B-L072Z-LRWAN1 et nous relions les broches SWDIO, SWCLK et GND de la carte B-L072Z-LRWAN1 aux broches correspondantes de notre carte électronique. Une fois cette connexion établie, nous pouvons utiliser l'interface de programmation STLink pour charger le code de programme dans le microcontrôleur STM32L072CZ

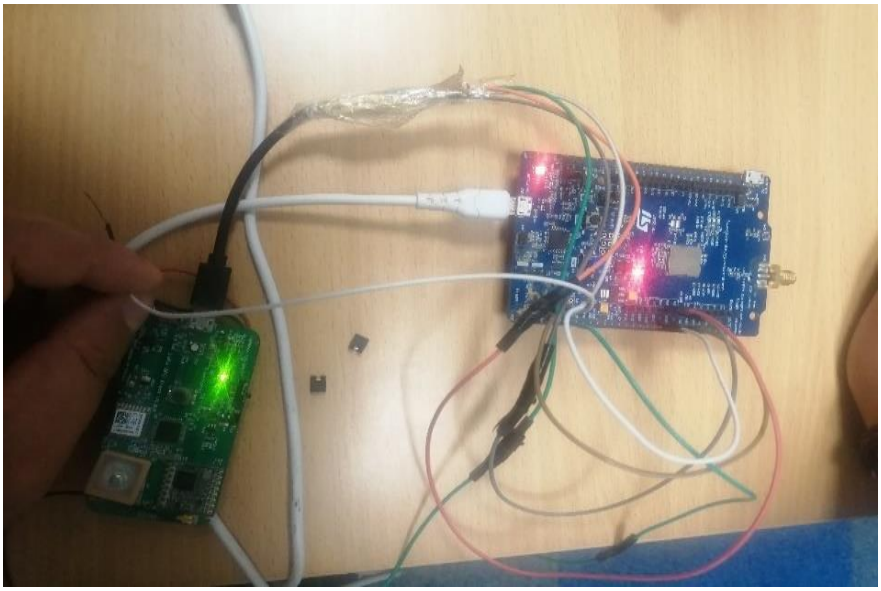


Figure 26 – La carte électronique finale après le soudage des composants

IV.6 Conclusion

Dans ce chapitre, nous avons décrit les étapes de la conception de notre carte électronique à deux couches en commençant par la réalisation du schéma électriquejusqu'au routage. En outre, nous avons expliqué chaque composant et différentes parties nécessaires afin de garantir les différentes fonctionnalités de notre système géolocalisation.

Conclusion générale

Au terme de ce projet de fin d'étude, nous avons présenté la conception d'une solution IoT de géolocalisation basée sur la technologie LoRaWAN pour l'entreprise Treetronix. Tout au long de ce projet, nous avons mis les différents aspects et étapes de la conception de cette solution. Nous avons étudié en détail les technologies que nous avons utilisées en expliquant le fonctionnement et les avantages de la technologie LoRaWAN.

Nous avons également évoqué les différents choix de matériels et logiciels que nous avons fait ainsi que l'utilisation de la plateforme Cayenne pour visualiser les données du GPS.
Nous avons détaillé les différentes étapes de conception d'une carte électronique de géolocalisation, dans lequel nous avons décrit les spécifications techniques de la carte et les choix de conception que nous avons faits.

Ce projet nous a permis de découvrir les méthodes d'intégration de l'IoT dans un objet classique et de nous initier à l'un des protocoles de communication LPWAN, qui est LoRaWAN.
De plus, nous a permis d'acquérir des compétences techniques dans le domaine de la réalisation et la conception d'une solution IoT.

Notre équipe a également été intégrée avec succès dans l'entreprise, ce qui a grandement contribué à la réussite globale du projet.

Enfin, ce projet nous a permis de mettre en pratique les connaissances théoriques acquises tout au long de notre formation et de mieux comprendre les exigences des entreprises en matière de développement de solutions innovantes.

Bibliographie

[1] Présentation du groupe 3S.

Disponible sur :https://www.3s.com.tn/en/.

[2] Présentation de la société Treetronix.

Disponible sur : https://treetronix.com/.

[3] La technologie LoRa.

Disponible sur : https://www.mokosmart.com/fr/lora-technology/.

[4] Le protocole LoRa.

Disponible sur : https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1197635-lora-comment-fonctionne-le-reseau-iot-20221010/.

[5] Protocole LoraWAN.

Disponible sur : https://www.univ-smb.fr/lorawan/en/livre-lora-lorawan-et-linternet-des-objets-2/

[6] Thinkparck.

Disponible sur : https://www.actility.com/ -

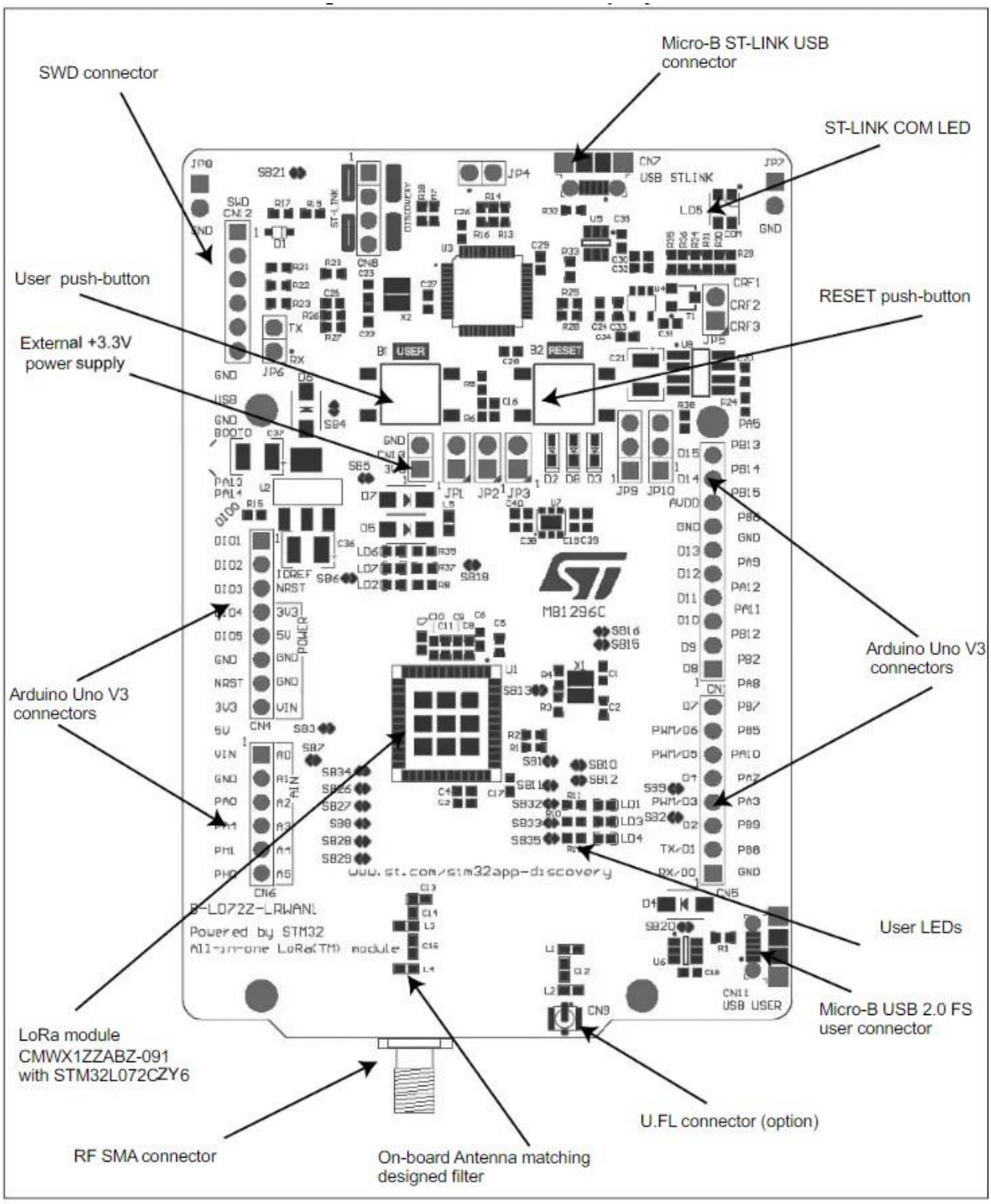
[7] B-L072Z-LRWAN1.

Disponible sur : https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html#documentation.

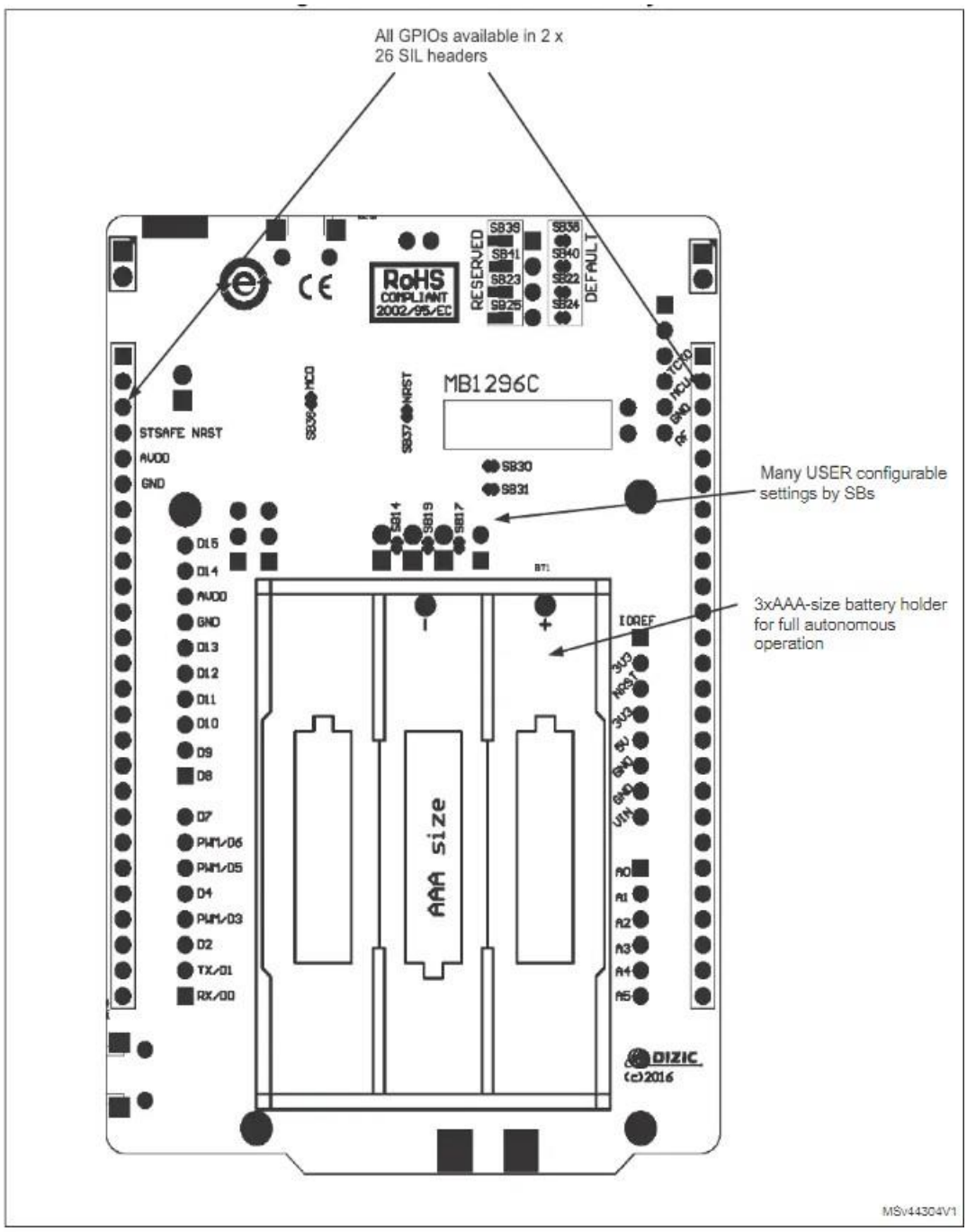
[8] stm32l0xx hardware development.

Disponible sur :http//www.st.com/resource/en/application_note/an4467-getting-started-with-stm32l0xx-hardware-development-stmicroelectronics.pdf

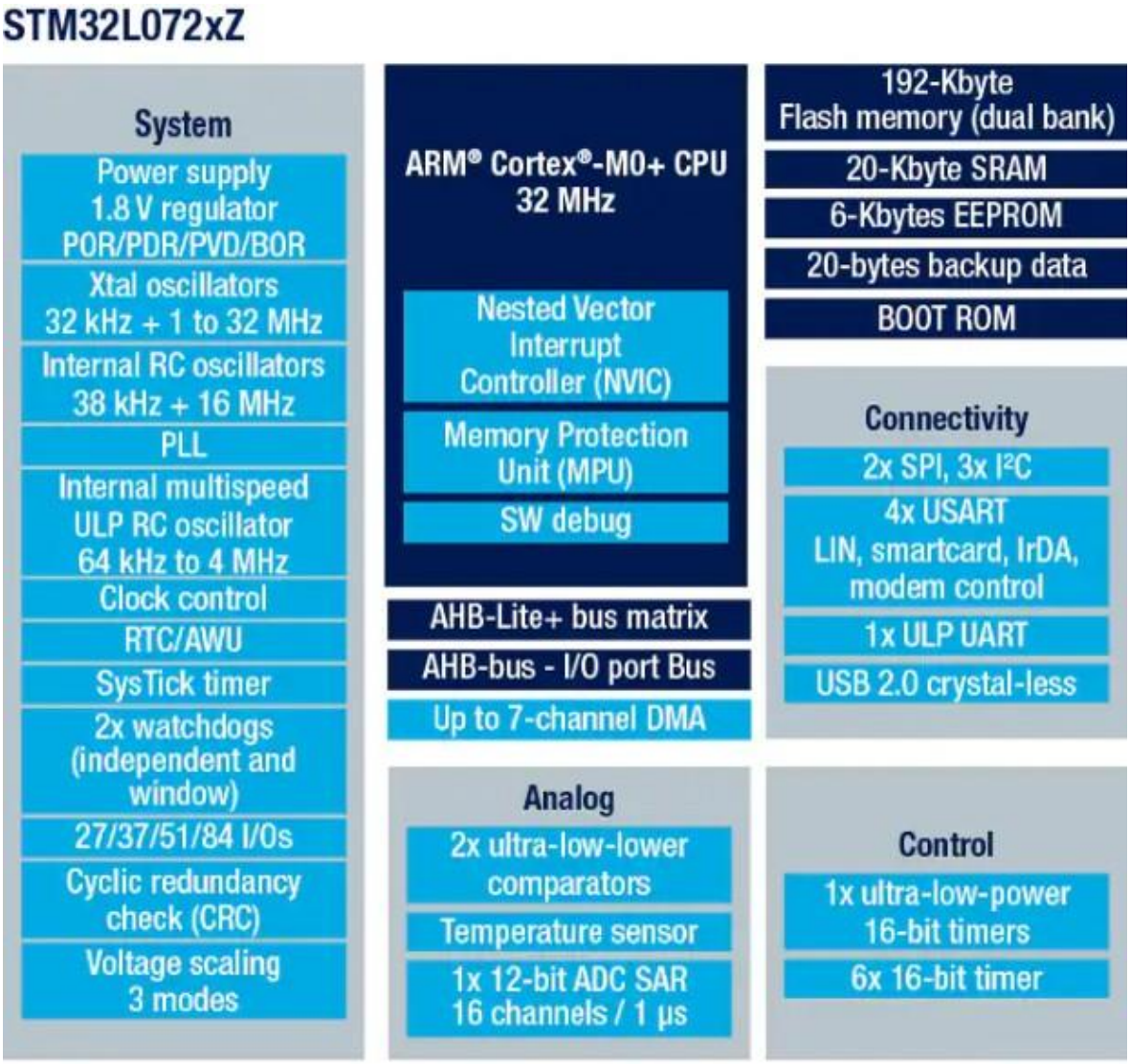
Annexe 1 : Disposition supérieure deB-L072Z-LRWAN1



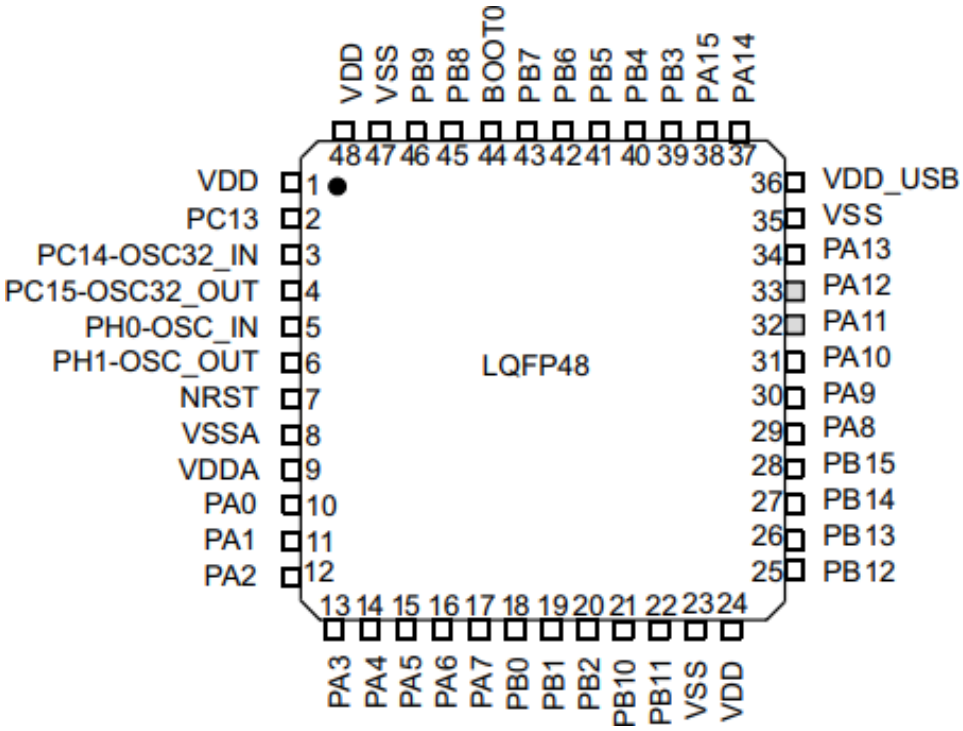
Annexe 2 : Disposition inférieure deB-L072Z-LRWAN1



Annexe 3 : Caractéristiques de STM32L072CZ



Annexe 4 : Les pins de STM32L072CZ



Annexe 5 : Les protocoles de communication pour les systèmes embarqués

1) Le protocole UART :

L'UART (Universal Asynchronous Receiver Transmitter) est un protocole série asynchrone utilisé fréquemment dans les systèmes électroniques embarqués.

Il définit les paramètres suivants :

• **Vitesse de transmission :** Elle définit le nombre de bits par seconde transmis via le port série UART. Les vitesses typiques vont de 9600 bps à 115200 bps.

• **Format des données :** nombre de bits, parité et bits d'arrêt.

• **Mode de fonctionnement :** le protocole UART fonctionne en mode asynchrone full-duplex, c'est-à-dire que l'envoi et la réception peuvent être effectués simultanément dans les deux sens.

• **Gestion des erreurs :** Les mécanismes de détection des erreurs reposent principalement sur les bits de parité et éventuellement sur les retransmissions.

Le bus UART se compose de 4 fils :

TX : Transmission des données.

RX : Réception des données.

GND : Masse commune pour deux dispositifs de communication.

VCC : Tension d'alimentation commune.

Le protocole UART est largement utilisé car il est simple, fiable, à faible coût et présent dans la plupart des microcontrôleurs, Il permet d'assurer la communication série entre des composants électroniques et se base uniquement sur 4 lignes simples et peu coûteuses, ce qui explique sa popularité.

Le protocole UART définit les paramètres de base pour une communication série asynchrone entre deux périphériques.

Les données sont transmises bit par bit, avec un bit de départ,

un certain nombre de bits de données, éventuellement un bit de parité pour la vérification d'erreur et un ou plusieurs bits d'arrêt. Le débit de transmission est déterminé par une horloge interne ou externe. Les données sont envoyées et reçues en utilisant deux lignes : la ligne de données RX et TX.

Ci-dessous un figure (Figure II.14).expliquant se qui précède :

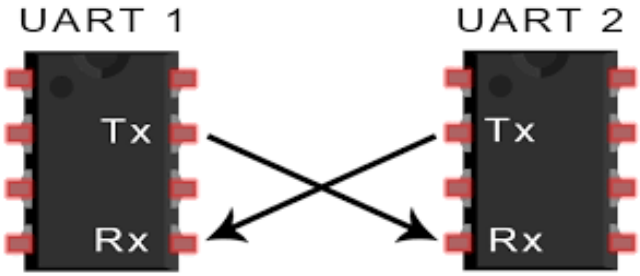


Figure A.1 : Communication de la broche Tx de l'UART émetteur vers la broche Rx de l'UART récepteur [36]

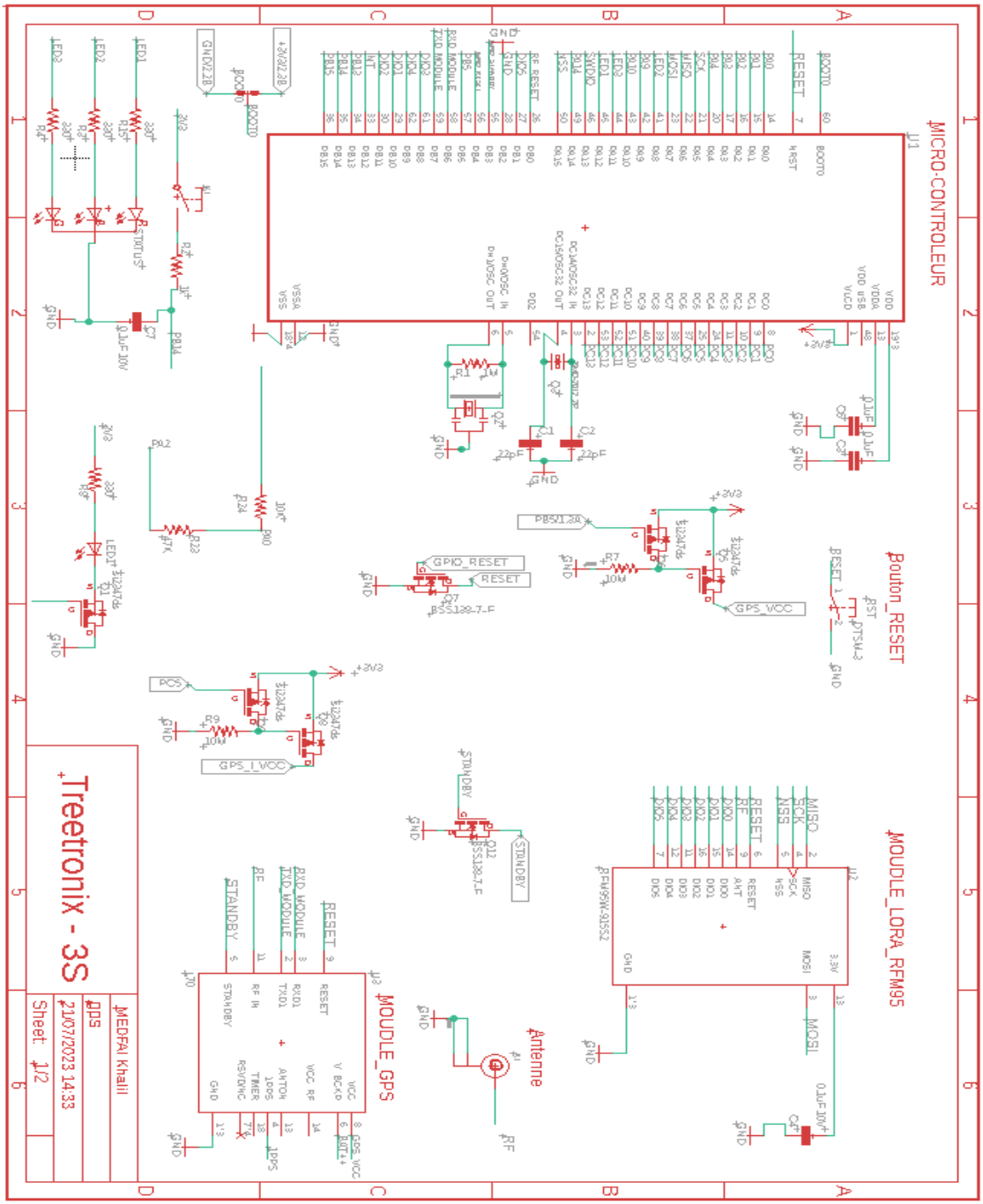
Tableau 9 - Tableau récapitulatif des avantages et des inconvénients de l'UART

Avantages	**Inconvénients**
Simplicité de mise en œuvre (nécessitant que 4 fils)	Fonctionnement asynchrone
Faible coût (grâce à sa simplicité)	Débits de transmission limités
Standardisation (Largement implémenté dans les microcontrôleurs)	Sensible aux perturbations

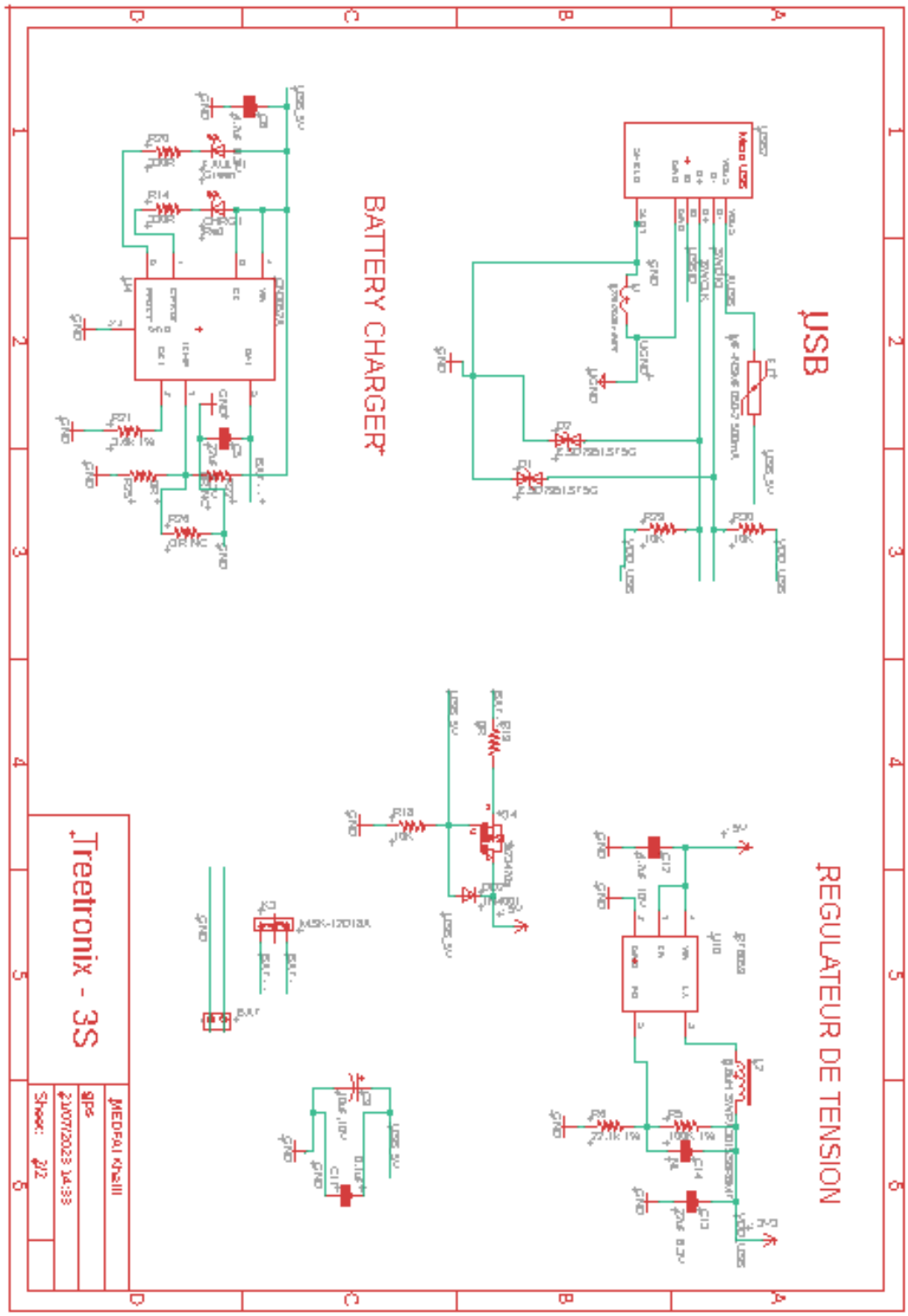
Avantages	**Inconvénients**
Détection d'erreur (Parité, détection de framing error)	Besoin d'un circuit UART sur chaque périphérique
Gamme de vitesses (De quelques bps à plusieurs Mbps)	Fiabilité limitée à haut débit

⇒ En somme, L'UART est pratique pour sa simplicité, son coût réduit et sa standardisation. Cependant, il présente des débits, une fiabilité et une synchronisation limités.

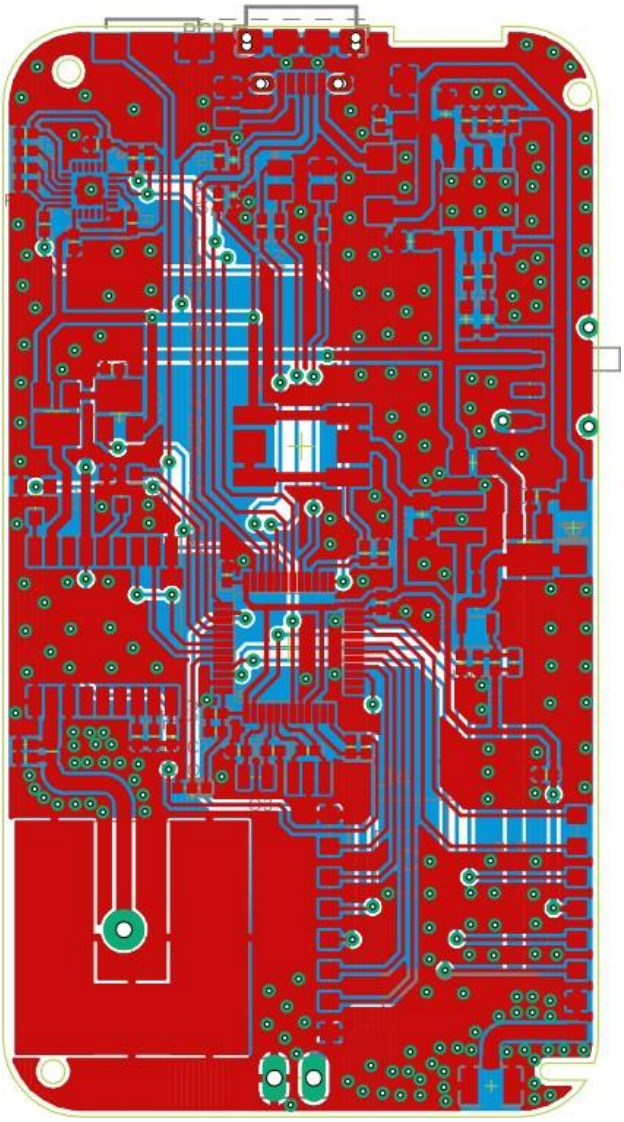
Annexe 6 : Schématique de la carte 1/2



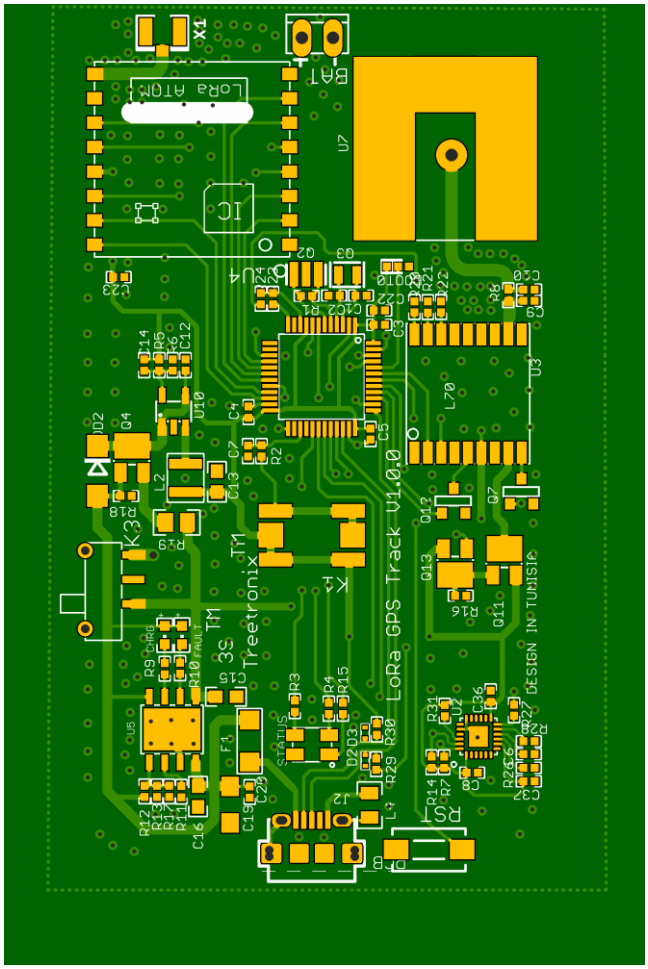
Annexe 7 : Schématique de la carte 2/2



Annexe 8 : Routage de la carte



Annexe 9 : Le schéma d'implantation de la carte



Développement d’un système de géolocalisation d’objets connectés sur le réseau LoRaWAN

Rapport de Stage PFE

RESUME

Dans le cadre de faire des solutions plus développées pour la géolocalisation, Nous avons réalisé notre projet de fin d’étude à Université Libre de Tunis au sein de la société Treetronix intitulé « Développement d’un système de géolocalisation d’objets connectés sur le réseau LoRaWAN ».

L’objectif principal de ce projet était de concevoir un système intelligent permettantde localiser et de suivre des objets à l’aide de la technologie LoRaWAN, en reposantsur l’utilisation d’un microcontrôleur STM32 et sur des compétences en conception électronique. L’idée était de développer un système qui puisse collecter et envoyer les données de localisation des objets vers un serveur. Dans cette optique, nousavons également intégré la plateforme Cayenne pour visualiser les données de localisation de manière conviviale et pratique.

Mots clés : LoRaWAN, STM32, IOT, Géolocalisation, Conception électronique, Cayenne.

SUMMARY

In order to develop more advanced solutions for geolocation, we carried out ourend of studies' project at Tunisia Private University, in collaborationwith Treetronix company. The project, entitled "Development of a geolocation system for connected objects on the LoRaWAN network server" aimed to design an intelligent system capable of locating and tracking objects using LoRaWAN technology, based on the STM32microcontroller and electronic design skills.

The main objective was to develop a system that could collect and send object location data to a server. To achieve this, we integrated the Cayenne platform, which provided a user-friendly and practical visualization of the location data.

Keywords: LoRaWAN, STM32, IoT, Geolocation, Electronic Design, Cayenne.