

UFR de mathématique et d'informatique

Université de Strasbourg

CURSUS MASTER EN INGÉNIERIE (CMI) - INFORMATIQUE, SYSTÈMES ET RÉSEAUX

Mémoire de stage de L3 présenté par

Hassan EL SAHILY

hassan.el-sahily@unistra.fr

LOCALISATION VIA DES CARTES RADIO ULTRA-WIDE-BAND (UWB)

30 juillet 2025

Stage encadré par

Fabrice THEOLEYRE

theoleyre@unistra.fr

Au sein du

LABORATOIRE DES SCIENCES DE L'INGÉNIEUR, DE L'INFORMATIQUE ET DE
L'IMAGERIE (ICUBE)



Table des matières

Table des matières	2
1 Remerciements	3
2 Introduction	4
3 Présentation du laboratoire ICube	5
4 Objectifs du stage et missions confiées	6
4.0.1 Objectifs généraux	6
4.0.2 Missions réalisées	6
4.0.3 Perspectives	6
5 Réalisations	7
5.0.1 Outils utilisées	7
5.0.2 Méthodologie adopté	8
5.0.3 Résultats	10
5.0.4 Problèmes rencontrés	11
5.0.5 Bilan	12
6 Conclusion	13
7 Références	14
8 Annexes	15

1 Remerciements

Je tiens en premier lieu à exprimer ma gratitude envers le laboratoire ICube pour m'avoir offert l'opportunité de réaliser ce stage.

Mes remerciements s'adressent tout particulièrement à Monsieur Pierre David, dont le soutien et les conseils précieux m'accompagnent depuis plusieurs années. C'est grâce à sa recommandation que j'ai pu intégrer ce stage, et je lui suis reconnaissant pour sa disponibilité constante et sa bienveillance. Je remercie également l'ensemble de mes enseignants, ainsi que l'équipe pédagogique et administrative de l'UFR de Mathématique et Informatique de l'Université de Strasbourg, pour la qualité de leur encadrement et leur accompagnement tout au long de mon parcours.

Je tiens à exprimer également ma profonde gratitude envers les personnes qui ont contribué de manière significative et m'ont aidé durant mon stage, en particulier Monsieur Fabrice Theoleyre, mon maître de stage, pour son encadrement, soutien constant et sa disponibilité à discuter de mes idées.

Je tiens également à remercier chaleureusement l'ensemble de l'équipe d'ICube, ainsi que les autres stagiaires, pour leur accueil bienveillant et leur esprit collaboratif. Leurs encouragements et leur disponibilité ont grandement facilité mon intégration au sein du laboratoire. Ils m'ont accordé leur confiance en me confiant des tâches variées et enrichissantes, tout en prenant le temps de m'accompagner dans la découverte et la maîtrise des outils informatiques nécessaires à la réalisation de mon travail.

2 Introduction

ICube est un laboratoire de recherche français rattaché à l'Université de Strasbourg, actif dans les domaines des sciences de l'ingénieur, de l'informatique et de l'imagerie. Il regroupe plusieurs équipes pluridisciplinaires menant des travaux à la fois fondamentaux et appliqués, notamment dans les domaines de l'imagerie, des réseaux de communication et des sciences des données.

Depuis le début de mon parcours, j'ai toujours été attiré par la recherche, et je souhaitais compléter ma formation académique par une expérience concrète au sein d'un environnement scientifique stimulant. C'est dans cette optique que j'ai choisi d'effectuer mon stage au sein de l'équipe Réseaux du laboratoire ICube, dont l'expertise est largement reconnue. Ce stage représentait pour moi une opportunité idéale d'approfondir mes connaissances, de participer à un projet de recherche appliquée, et de travailler aux côtés de chercheurs expérimentés.

Le sujet du stage portait sur la mise en œuvre d'un système de localisation en environnement intérieur à l'aide de modules UWB (Ultra-Wideband) MaUWB et d'un affichage temps réel sur un microcontrôleur tactile ESP32S3 (MaTouch). L'objectif était de construire un démonstrateur interactif, capable d'afficher dynamiquement la position de plusieurs tags sur une carte 2D embarquée, en exploitant une infrastructure radio open source existante.

Le défi principal consistait à intégrer plusieurs composants hétérogènes (cartes UWB, ESP32, écran tactile) dans une architecture fluide, réactive et portable. Une attention particulière a été portée à la gestion de la communication série entre les modules, au traitement des données de localisation, et à leur représentation graphique claire et intuitive. La problématique générale du projet était sur comment concevoir un système embarqué capable d'exploiter efficacement les données de localisation UWB et de les restituer en temps réel, sans configuration manuelle préalable.

Pour répondre à ces objectifs, ce rapport est structuré comme suit : d'abord est présenté le contexte du stage, les objectifs poursuivis et les missions confiées au sein du laboratoire. Ensuite sont décrites la plateforme matérielle et logicielle mise en place ainsi que les outils mobilisés pour le projet. Vient ensuite l'exposé de la méthodologie adoptée pour le développement du système, suivi de la présentation et de l'analyse des résultats obtenus. Enfin, le rapport se termine par un bilan du travail réalisé et les perspectives d'amélioration envisagées.

3 Présentation du laboratoire ICube

ICube est un laboratoire de recherche pluridisciplinaire rattaché à l'Université de Strasbourg. Créé en 2013, il regroupe les compétences en sciences de l'ingénieur et en informatique, avec une particularité : l'imagerie est au cœur de ses projets, que ce soit dans la santé, l'environnement ou le développement durable.

En d'autres termes, ICube conçoit et développe des technologies innovantes qui vont de la robotique chirurgicale aux réseaux de capteurs, en passant par l'intelligence artificielle, la bio-informatique ou encore les énergies renouvelables.

Son organisation est structurée autour de quatre grands départements qui travaillent en synergie :

- Informatique (D-IR) : réseaux, calcul haute performance, intelligence artificielle et bio-informatique.
- Électronique, Systèmes et Photonique (D-ESSP) : nanoélectronique, photonique, systèmes embarqués, instrumentation médicale, énergies propres.
- Imagerie, Robotique, Télédétection et Santé (D-IRTS) : imagerie médicale, robotique chirurgicale, neurosciences, observation de la Terre.
- Mécanique (D-M) : mécanique des fluides, biomécanique, génie civil, géothermie, matériaux innovants.

La mission principale d'ICube est de transformer la recherche en solutions concrètes pour relever les grands défis de notre société. Pour y parvenir, le laboratoire mène de nombreux projets collaboratifs avec des hôpitaux, des entreprises et des institutions, notamment dans le cadre des grands programmes nationaux tels que les "Investissements d'Avenir". Il abrite également deux Équipements d'Excellence (Robotex et FIT) et trois Laboratoires d'Excellence (IRMIA, CAMI, G-Eau-Thermie).

Avec plus de 450 chercheurs, ingénieurs, doctorants et techniciens, ICube joue un rôle clé en réunissant le monde académique, le secteur industriel et le domaine médical. Le laboratoire joue aussi un rôle majeur dans la formation des futurs experts, grâce à ses collaborations avec plusieurs écoles d'ingénieurs et à l'accueil de nombreux stagiaires et doctorants.

En résumé, ICube est un laboratoire d'expertise qui établit le lien entre la recherche fondamentale et ses applications concrètes, afin de favoriser l'innovation au service de la santé, de l'environnement et de la société.

4 Objectifs du stage et missions confiées

Ce stage a été réalisé au sein de l'équipe Réseaux du laboratoire ICube, sous la supervision du Dr Fabrice Théoleyre. Il s'inscrit dans le domaine de la localisation radio en utilisant la technologie Ultra-Wide-Band (UWB), avec pour objectif principal de développer une plateforme de démonstration complète de localisation en environnement indoor.

4.0.1 Objectifs généraux

La localisation par UWB consiste à déterminer la position d'objets ou de personnes dans un espace tridimensionnel, en mesurant les distances entre des balises fixes (ancres) et des balises mobiles (tags). Cette technologie, particulièrement robuste face au multichemin radio, est bien adaptée aux environnements complexes comme les bâtiments.

Le stage avait ainsi pour but de concevoir un système fonctionnel, interactif et facilement configurable, permettant non seulement de réaliser cette localisation, mais aussi de visualiser en temps réel la position des balises sur une interface dédiée.

4.0.2 Missions réalisées

Pour atteindre ces objectifs, plusieurs missions ont été confiées :

- Exploiter l'implémentation open source existante pour la localisation des tags, en s'appuyant sur la plateforme MaUWB développée par Makerfabs.
- Automatiser les scénarios de test et la configuration du système, incluant notamment la localisation des ancres et la mise en place d'un tableau de bord de contrôle.
- Étendre l'affichage et les fonctionnalités des dispositifs (ancres et tags), notamment grâce à l'intégration d'écrans LCD configurables.
- Développer une interface de visualisation en temps réel sur un écran tactile ESP32 (Ma-Touch), permettant de représenter dynamiquement la position des balises dans l'espace.

4.0.3 Perspectives

Une fois ces tâches accomplies, plusieurs pistes d'amélioration ont été identifiées, notamment :

- Étendre le tableau de bord avec de nouvelles fonctionnalités interactives.
- Collecter et analyser les données de localisation afin d'évaluer les performances du système.
- Explorer une représentation des positions en coordonnées relatives plutôt qu'absolues, afin de rendre le système plus autonome et adaptable.

En résumé, ce stage visait à concevoir une démonstration complète et interactive de localisation UWB, combinant traitement des données, automatisation et visualisation, pour offrir un système à la fois robuste et évolutif.

5 Réalisations

5.0.1 Outils utilisés

Pour mener à bien ce projet, nous avons utilisé plusieurs outils matériels et logiciels, chacun jouant un rôle essentiel dans son développement.

Sur le plan matériel, nous avons principalement utilisé :

- **Des modules MaUWB** basés sur la puce DW3000, pouvant être configurés soit comme ancrs (points fixes servant de référence) soit comme tags (éléments mobiles à localiser). Ces modules prennent en charge jusqu'à 8 ancrs et 64 tags simultanément. Chaque module intègre un microcontrôleur STM32 pré-programmé par Makerfabs, qui prend en charge les tâches essentielles liées au fonctionnement UWB, telles que la gestion des créneaux temporels et la coordination des communications entre plusieurs ancrs et tags. Les résultats de ces traitements sont ensuite transmis via UART, ce qui simplifie considérablement l'intégration et l'utilisation des modules.
- **Une carte MaUWB ESP32-S3 MaTouch** de MakerFabs, intégrant un microcontrôleur ESP32-S3, un écran tactile IPS de 7 pouces et un transceiver UWB basé sur la puce DW3000.
D'un côté, cette carte assure le positionnement en temps réel grâce à la technologie UWB, en communiquant avec plusieurs ancrs et tags ; de l'autre, elle permet d'afficher directement sur son écran tactile les informations collectées, offrant ainsi un suivi visuel immédiat.
La connectivité via le protocole ESP-NOW, un protocole de communication sans fil développé par Espressif utilisant le matériel Wi-Fi intégré des ESP32 mais fonctionnant

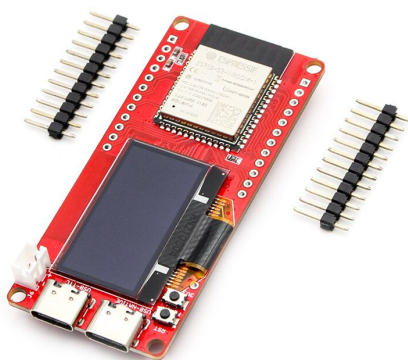


FIGURE 5.1 – Module MaUWB (ancres ou tag) [4].

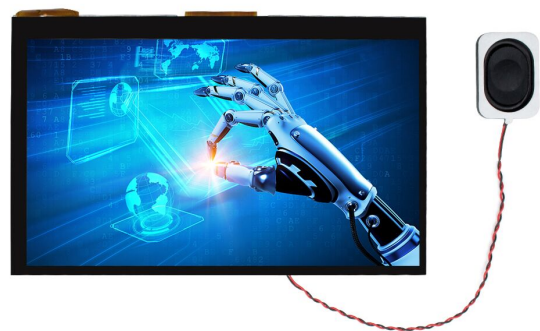


FIGURE 5.2 – Carte MaUWB ESP32-S3 MaTouch [2]

en mode *peer-to-peer* (sans point d'accès ni routeur), a permis des communications rapides et fiables entre les différents modules du système. Par ailleurs, l'ajout d'une carte MicroSD a permis le stockage local des données expérimentales, garantissant ainsi une traçabilité complète.

Côté logiciel, le développement du firmware a été réalisé en langage C/C++. Une première phase a été menée sous Arduino IDE avec un fichier principal `.ino`, puis le projet a été migré sous PlatformIO, où la logique a été réorganisée en plusieurs modules `.cpp/.h` pour une meilleure lisibilité et maintenabilité. Le tout s'appuie sur des bibliothèques spécialisées telles que `LVGL` pour la création d'interfaces graphiques interactives, ainsi que sur les bibliothèques UWB dédiées à la gestion des ancres et des tags.

Grâce à cette combinaison d'outils, nous avons pu intégrer de manière cohérente la partie embarquée et le traitement des données, obtenant ainsi un système fonctionnel et fiable répondant aux objectifs du projet.

5.0.2 Méthodologie adopté

Pour débiter cette exploration, je présente ci-dessous la méthodologie adoptée.

Dans un premier temps, il a été nécessaire d'analyser et d'exploiter l'implémentation open-source fournie par Makerfabs pour les modules MaUWB, afin de comprendre leur fonctionnement, d'identifier leurs limites et de poser les bases de la communication entre ancres et tags.

Une seconde étape a consisté à adapter le système aux besoins du projet, en automatisant certains paramètres pour garantir un fonctionnement stable, quel que soit le nombre de tags déployés. Enfin, la carte MaTouch a été enrichie d'un affichage en temps réel des positions sur un plan de la salle, d'un suivi des trajectoires et d'un enregistrement des données sur carte MicroSD. Cette approche a permis de disposer d'une plateforme complète de démonstration, combinant traitement embarqué, visualisation et traçabilité, tout en préparant le terrain pour de futures expérimentations.

Exploitation de l'existant

Makerfabs propose déjà une implémentation open-source de base pour ses modules MaUWB, offrant des fonctionnalités minimales permettant d'assurer la communication entre ancres et tags. Cette implémentation a servi de point de départ à notre projet, principalement pour comprendre l'organisation du code, la configuration des ports et l'utilisation des GPIO.

Pour les modules MaUWB, deux codes distincts étaient disponibles : l'un dédié aux ancres et l'autre aux tags.

- **Pour les ancres** : Au début du projet, aucun développement supplémentaire n'a été nécessaire, car leur rôle principal se limitait à assurer la communication UWB avec les tags. Toutefois, en raison de l'absence de documentation complète, des expérimentations ont été menées par la suite afin d'optimiser cette communication. Ces ajustements ont été motivés par la nécessité d'accélérer les échanges avec les tags, notamment dans le cadre d'un projet parallèle mené par une collègue en stage de M2, qui développait un robot nécessitant un calcul de position plus rapide que celui fourni par l'implémentation initiale.
- **Pour les tags** : Un travail plus approfondi a été nécessaire afin d'adapter le code fourni par Makerfabs aux besoins du projet. Dans un premier temps, la configuration matérielle

a été précisée (définition des GPIO, initialisation des interfaces série et I²C), et les bibliothèques nécessaires ont été intégrées (`Wire`, `Adafruit_GFX`, `Adafruit_SSD1306`, `ESP-NOW`). Un affichage local a été ajouté via un écran OLED, permettant de visualiser directement les distances mesurées avec chaque ancre. Enfin, les données collectées (identifiant du tag et distances par rapport aux ancres) sont encapsulées dans une structure spécifique et transmises en temps réel à la carte MaTouch via le protocole ESP-NOW, assurant ainsi une communication rapide et fiable pour l’affichage et le traitement centralisé.

Adaptation et automatisation

En fonction du nombre total de tags, des créneaux temporels (*time slots*) ont été réservés pour chacun d’eux, afin d’organiser leur communication avec les ancres et d’éviter toute émission simultanée. Finalement, le cycle complet a été adapté pour maintenir une durée fixe de 500 ms, quel que soit le nombre de tags (jusqu’à un maximum de 64), tout en garantissant que la durée minimale allouée à chaque tag reste supérieure à 10 ms. Ce paramétrage dynamique a permis d’automatiser le fonctionnement du système et de le rendre flexible, sans nécessiter de reconfiguration manuelle en fonction du nombre de modules déployés.

Parallèlement, une étape importante a consisté à centraliser l’ensemble des paramètres du projet dans un fichier de configuration global (`config.h`). Ce fichier définit notamment :

- la résolution de l’écran (`SCREEN_W`, `SCREEN_H`) et les paramètres nécessaires à l’initialisation du bus graphique ;
- le plan de la salle, incluant les positions prédéfinies des ancres (en mètres) et les facteurs d’échelle permettant de convertir ces coordonnées en pixels pour l’affichage ;
- les constantes liées au stockage sur carte SD (pins utilisés, taille des buffers) et aux limites du système (nombre maximal d’ancres et de tags) ;
- des macros facilitant la conversion des coordonnées physiques en coordonnées graphiques (`PX_X`, `PX_Y`).

Grâce à cette structuration, l’ensemble du projet est devenu plus modulaire : il est possible d’adapter rapidement la configuration (dimensions de la salle, disposition des ancres, nombre de tags et ancres, etc) sans modifier le cœur du code. Cette approche a simplifié le développement et réduit le risque d’erreurs lors de l’ajout de nouvelles fonctionnalités.

Extension sur MaTouch et visualisation

Pour la carte MaTouch, la situation a été plus complexe en raison de la présence d’une implémentation très générale (initialement prévue pour des projets variés tels que des jeux ou des horloges), qui sortait du cadre de notre projet, ainsi que de l’absence de documentation dédiée pour l’utilisation des fonctionnalités spécifiques et l’accès aux ports et GPIO. Il a donc été nécessaire d’examiner le code de projets existants afin de comprendre la manière dont l’écran, la carte SD et les autres périphériques étaient initialisés et configurés, et d’identifier la configuration adéquate à appliquer pour le projet.

Un développement spécifique a ensuite été réalisé de zéro pour intégrer l’ensemble des fonctionnalités nécessaires. L’affichage graphique a été confié à la bibliothèque `LVGL`, en combinaison avec `Arduino_GFX`, afin de piloter l’écran tactile 7 pouces via un bus RGB, et un *canvas* dynamique a été implémenté pour afficher en temps réel la carte de la pièce ainsi que les trajectoires des tags.

La communication entre la carte MaTouch et les modules UWB s'appuie sur le protocole `ESP-NOW`, utilisant le matériel Wi-Fi intégré en mode *peer-to-peer*, ce qui permet des transmissions rapides et fiables des mesures de distance. Un *callback* dédié (`onDataRecv`) traite les données reçues, met à jour les positions calculées des tags, déclenche leur affichage et consigne simultanément les résultats dans un fichier `CSV` stocké sur carte `MicroSD`.

Afin de fournir une représentation visuelle intuitive, la carte a été programmée pour afficher sur l'écran tactile un plan schématique de la salle dans laquelle évolue le système. Les positions des tags y sont représentées dynamiquement, et un tracé continu du déplacement est dessiné, permettant de suivre en temps réel la trajectoire, notamment celle du robot utilisé dans les expérimentations. Cette logique a été structurée autour d'une classe `UWB`, conçue pour encapsuler les fonctionnalités liées à chaque module (ancres ou tag), telles que la gestion des distances mesurées, le calcul de position, l'actualisation graphique et l'export des données.

Ainsi, la carte MaTouch remplit un rôle central : elle collecte, traite, visualise et archive les données de localisation en temps réel, tout en offrant une interface graphique claire et adaptée aux besoins du projet.

5.0.3 Résultats

Grâce à l'architecture mise en place, le système a permis de localiser jusqu'à 64 tags simultanément, avec une fréquence de mise à jour des positions fixée à 500 ms. L'affichage temps réel sur l'écran MaTouch a facilité l'analyse visuelle des trajectoires, et le stockage des mesures dans des fichiers `CSV` a assuré une traçabilité complète des expérimentations.

Pour le calcul des positions, deux approches ont été implémentées. Dans un premier temps, un algorithme simple basé sur la moyenne des positions calculées par paires d'ancres a été utilisé. Celui-ci, illustré dans la fonction `cal_avg()`, détermine pour chaque paire d'ancres une estimation de la position du tag en considérant les cercles d'intersection définis par les distances mesurées, puis fait la moyenne de toutes ces estimations. Bien que cette méthode offre une première approximation acceptable même avec un faible nombre de mesures valides, elle reste sensible aux valeurs bruitées et ne tient pas compte de la qualité relative des différentes distances.

Afin d'améliorer la précision, cet algorithme a été remplacé par une méthode de *Weighted Least Squares* (WLS). Cette approche résout un système surdéterminé en minimisant l'erreur quadratique pondérée entre les distances mesurées et celles calculées à partir de la position estimée, les poids reflétant la fiabilité des mesures. Ainsi, les lectures bruitées ont un impact réduit sur l'estimation finale. Ce choix a permis d'obtenir des positions plus stables et plus précises, en particulier dans les environnements à forte variabilité radio.

Les résultats obtenus sont illustrés en figure 5.3. On observe que les trajectoires estimées à l'aide de l'algorithme WLS (points verts) suivent globalement le tracé de référence (en rouge), avec des écarts locaux pouvant atteindre plusieurs dizaines de centimètres. Néanmoins, ces résultats s'avèrent nettement plus précis que ceux obtenus avec l'algorithme initial basé sur la moyenne des positions par paires d'ancres (en jaune), qui présentait des déviations plus importantes et une moindre stabilité. Ces écarts résiduels peuvent être attribués :

- au bruit de mesure inhérent à la technologie UWB, amplifié par les réflexions multipath dans l'environnement intérieur ;
- aux limites du WLS, qui fournit une estimation statique sans prendre en compte la cohérence temporelle des positions successives ;

Ainsi, bien que le WLS offre une solution rapide et adaptée aux contraintes de calcul de l'ESP32-S3,

ces résultats soulignent la nécessité d'intégrer à l'avenir des algorithmes de post-traitement ou de filtrage afin d'améliorer la précision et la stabilité des trajectoires.

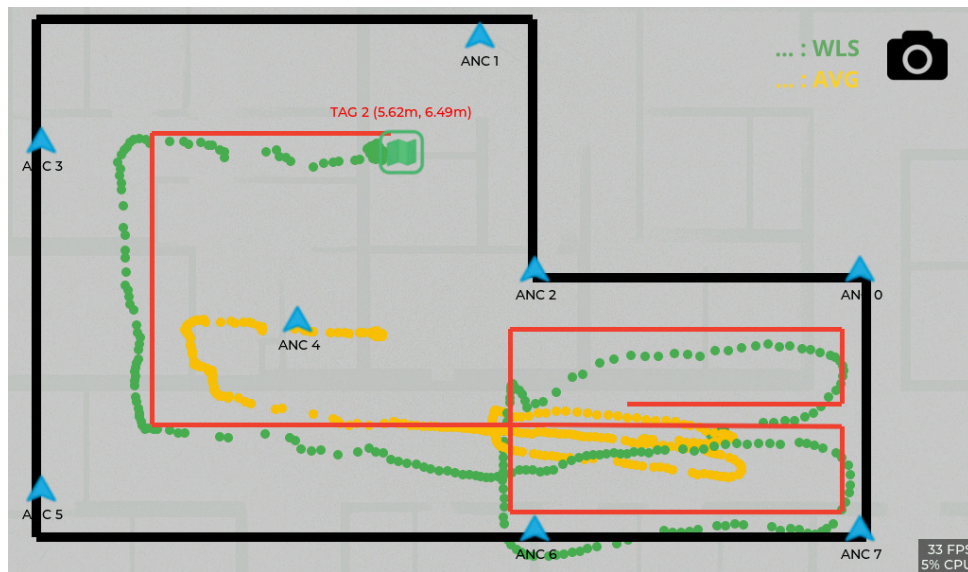


FIGURE 5.3 – Affichage en temps réel des tags et de leurs trajectoires sur la MaTouch.

5.0.4 Problèmes rencontrés

L'intégration au sein de l'équipe s'est révélée très positive : mon maître de stage était toujours à l'écoute et attentif à mes idées, et l'équipe, internationale et bienveillante, a favorisé une communication fluide, même avec les personnes ne parlant pas français.

Cependant, plusieurs défis ont marqué ce projet. Tout d'abord, je n'avais jamais programmé en C++ auparavant. Bien que l'apprentissage se soit avéré enrichissant, il a nécessité un temps d'adaptation, d'autant plus que je devais simultanément me familiariser avec l'environnement Arduino IDE. La migration vers PlatformIO sous VS Code a ensuite simplifié le travail, notamment en réduisant considérablement le temps de compilation, mais cette transition représentait également une nouveauté pour moi (y compris ma collègue en M2), ce qui m'a contraint à mener de nombreuses recherches personnelles pour résoudre les problèmes rencontrés.

Un autre problème majeur concernait la connexion réseau de la carte MaTouch. Malgré de longues sessions de débogage, il a été impossible de la connecter au réseau local de la salle au laboratoire, ni même à un point d'accès personnel, et les tentatives de transformation de la MaTouch en point d'accès ou d'utilisation d'un adaptateur Wi-Fi USB se sont révélées peu concluantes. Après avoir perdu un temps conséquent sur cette question, j'ai choisi d'abandonner cette approche et d'adopter le protocole ESP-NOW, qui s'est avéré plus adapté aux besoins du projet. Ce changement, bien que stressant dans un premier temps, a permis de repartir sur de nouvelles bases et de renforcer l'efficacité globale du système.

Un autre effet de ces difficultés a été la réduction du temps disponible pour explorer certaines pistes que j'avais initialement prévues, notamment l'intégration d'algorithmes d'apprentissage automatique visant à améliorer la localisation. L'idée était d'identifier, pour chaque mesure, le signal le plus faible ou celui le plus affecté par le multichemin, puis de l'exclure afin d'optimiser la précision des calculs de position. Faute de temps, cette approche n'a pas pu être développée comme je l'avais envisagé, ce qui constitue une piste d'amélioration importante pour la suite du projet.

5.0.5 Bilan

Les tâches que j'ai entreprises au cours de ce stage ont pleinement contribué à l'atteinte des objectifs fixés, tout en me confrontant à des défis techniques stimulants. L'apprentissage du C++ et la migration d'Arduino IDE vers PlatformIO m'ont permis de développer des compétences solides en programmation embarquée, d'améliorer ma compréhension des systèmes temps réel et d'adopter de meilleures pratiques de développement logiciel. Cette montée en compétence s'est accompagnée d'une meilleure gestion du temps, nécessaire pour respecter les délais et hiérarchiser efficacement les priorités dans un projet complexe.

Les problèmes liés à la connectivité réseau m'ont appris à ne pas m'enliser dans une solution unique et à envisager rapidement des alternatives techniques, comme l'adoption du protocole ESP-NOW, plus adapté au projet. Cultivant ainsi un esprit critique et une approche pragmatique de la résolution de problèmes.

En parallèle, la conception d'interfaces graphiques interactives avec LVGL et la mise en œuvre d'algorithmes de localisation, notamment par la méthode des moindres carrés pondérés, m'ont permis de revisiter et de mettre en pratique les notions abordées en cours d'Analyse numérique appliquée. Cette intégration des connaissances théoriques dans un contexte concret a renforcé mes compétences en mathématiques appliquées et approfondi ma compréhension des algorithmes utilisés dans les systèmes de localisation, tout en améliorant ma capacité à traiter efficacement des données embarquées.

Enfin, ce stage m'a offert une réelle immersion dans un environnement de recherche collaboratif et international, où la communication avec des interlocuteurs de profils variés a enrichi ma vision du travail en équipe. Cette expérience m'a non seulement permis de gagner en autonomie et en confiance dans mes compétences, mais elle a également renforcé mon intérêt pour la recherche appliquée.

6 Conclusion

Mon stage a constitué une opportunité précieuse, me permettant d'accomplir des tâches en parfaite concordance avec mes objectifs personnels tout en apportant une réelle valeur ajoutée au laboratoire. Le développement de ce projet s'inscrit d'ailleurs dans une démarche collaborative, en complément d'un sujet de recherche de niveau Master 2, visant à concevoir un système de suivi en temps réel des déplacements. Dans ce cadre, la solution développée sur les modules UWB (tags) et l'affichage MaTouch sera directement exploitée, avec des perspectives d'application dans un contexte hospitalier.

Ce stage a eu un impact significatif sur mon développement, tant professionnel que personnel. Il m'a offert la chance de côtoyer mes enseignants dans un cadre différent de l'université, ce qui m'a permis de mieux apprécier leur engagement et leur expertise. J'ai ainsi acquis une vision plus globale du monde de la recherche et de sa dynamique, découvrant que les projets informatiques dépassent largement le cadre de la simple programmation pour inclure des dimensions organisationnelles, collaboratives et scientifiques.

Mon rôle dans ce stage m'a conduit à travailler de manière autonome, à comprendre des systèmes logiciels complexes et à explorer des solutions adaptées aux contraintes du projet. Cette autonomie dans la méthodologie et la résolution de problèmes m'a permis de confirmer mon intérêt pour les projets où je peux prendre des initiatives et explorer différentes approches. Ce constat, associé aux compétences acquises au sein du laboratoire, a renforcé mon intérêt pour une poursuite en thèse et m'a ouvert à la perspective d'une carrière dans la recherche, un domaine que je n'avais jusqu'alors ni envisagé ni considéré.

Dans la continuité de cette expérience, je souhaite approfondir mes compétences en programmation tout en maintenant mon orientation vers les systèmes et réseaux, qui continuent de constituer le cœur de mes intérêts. La formation dispensée par la Faculté de mathématiques et informatique de l'Université de Strasbourg m'a offert bien plus que des bases techniques solides : elle m'a permis de développer un réseau professionnel, d'élargir mes horizons et de repenser ma vision de mon avenir.

7 Références

- [1] MAKERFABS. *ESP32-S3 AT UWB Pro with Display v1.0 - Hardware Documentation*. Consulté le 3 juin 2025. 2025. URL : https://github.com/Makerfabs/MaUWB_ESP32S3-with-STM32-AT-Command/tree/main.
- [2] MAKERFABS. *MaTouch ESP32-S3 Parallel TFT with Touch 7 Inch*. Consulté le 2 juin 2025. URL : <https://www.makerfabs.cc/product/matouch-esp32-s3-parallel-tft-with-touch-7-inch.html>.
- [3] MAKERFABS. *MaTouch ESP32-S3 Rotary IPS Display with Touch 7 - PlatformIO Project*. Consulté le 10 juin 2025. 2025. URL : <https://github.com/Makerfabs/ESP32-S3-Parallel-TFT-with-Touch-7inch/tree/main/example>.
- [4] MAKERFABS. *MaUWB ESP32-S3 UWB Module*. Consulté le 2 juin 2025. 2025. URL : <https://www.makerfabs.com/mauwb-esp32s3-uw-module.html>.
- [5] MAKERFABS. *MaUWB New Firmware Upgrade V1.1.2*. Consulté le 2 juin 2025. 2025. URL : https://www.makerfabs.com/blog/post/mauwb-new-firmware-upgrade_v112.
- [6] MAKERFABS. *New Features of MaUWB - Data Transmission*. Consulté le 2 juin 2025. 2024. URL : <https://www.makerfabs.com/blog/post/new-features-of-mauwb-data-transmission>.
- [7] MAKERFABS. *Wiki - MaTouch ESP32-S3 7" Parallel TFT with Touch*. Consulté le 2 juin 2025. URL : https://wiki.makerfabs.com/MaTouch_ESP32_S3_7_Paralle_%20TFT_with_Touch.html.
- [8] MAKERFABS. *Wiki - MaUWB ESP32-S3 UWB Module*. Consulté le 3 juin 2025. 2025. URL : https://wiki.makerfabs.com/MaUWB_ESP32S3%20UWB%20module.html.

8 Annexes

Le code source complet développé dans le cadre de ce projet est disponible publiquement sur GitLab à l'adresse suivante :

https://gitlab.com/hassansahili29/uwb_demo.git