

**Institut Universitaire de Technologie,
Aix-Marseille Université**

**RAPPORT DE STAGE de fin de deuxième année
Bachelor Universitaire de Technologie
Spécialité Réseaux et Télécommunications
parcours cybersécurité**

**Développement logiciel embarqué et ressources
pédagogiques pour la carte
STeaMi**

Mattéo CANADA

Laboratoire d'Aix-Perimentation et de Bidouille

Responsable entreprise : Sébastien Nedjar

Responsable académique : Arnaud Fevrier

2026

PAGE BLANCHE

Le document sera imprimé en recto-verso, donc il ne doit pas y avoir de texte sur cette feuille.
Et il faut supprimer ces 3 lignes !

Table des matières

1	Introduction.....	5
2	Présentation de la structure d'accueil.....	6
2.1	Le LAB : Laboratoire d'Aix-périmentation et de Bidouille	6
2.2	Activités principales.....	6
2.3	Objectifs pédagogiques et techniques	6
2.4	Public visé.....	6
2.5	L'environnement technique du stage	6
2.5.1	Présentation de la carte STEAMI.....	7
2.5.2	Les capteurs intégrés	7
2.5.3	Écran OLED.....	7
2.5.4	Communication BLE	8
2.5.5	Environnement logiciel	8
3	Développement des projets techniques	8
3.1	Mise en place et prise en main de la STeaMi	8
3.1.1	Installation de l'environnement de développement sur Debian.....	9
3.1.2	Premiers tests BLE et advertisements	9
3.2	Développement des communications Bluetooth Low Energy	10
3.2.1	Fonctionnement du BLE sur la STeaMi	10
3.2.2	Mise en place d'un réseau BLE multi-cartes	11
3.2.3	Système de localisation par trilatération BLE.....	12
3.3	Développement de projets ludiques et algorithmiques	16
3.3.1	Maze Game et algorithme de Dijkstra	16
3.4	Optimisation logicielle et migration vers le C++.....	17
3.4.1	Comparaison MicroPython / C++	18
3.4.2	Optimisation mémoire et performances	18
3.4.3	Portage du système de localisation en C++ : gains concrets	18
4	Participation aux activités pédagogiques	18
4.1	Ateliers scolaires CE2 / CM1.....	18
4.1.1	Scratch déconnecté.....	19
4.1.2	Bits de parité	19
4.1.3	Tours de Hanoï.....	19
4.2	Journée ITER Robots — Stand STeaMi	19
4.3	Création de ressources pédagogiques	19
5	Bilan du stage.....	20
5.1	Compétences techniques acquises	20
5.2	Difficultés rencontrées et solutions apportées	20
5.3	Apports professionnels et personnels.....	20
6	Conclusion	20
7	Remerciements.....	22
8	Glossaire.....	24
9	Bibliographie.....	27

PAGE BLANCHE

Le document sera imprimé en recto-verso, donc il ne doit pas y avoir de texte sur cette feuille.
Et il faut supprimer ces 3 lignes et les lignes en vert ci-dessous :

Forme :

Le texte doit être justifié à gauche ET à droite.

Nombre de pages :

Depuis l'introduction (sur cette page) jusqu'à la conclusion, il doit y avoir minimum 20 pages (22 pages maximum). Si on compte depuis la couverture jusqu'à la fin cela fait 30 à 32 pages maximum.
Le reste peut être mis en annexe.

Pieds de page :

Le contenu des pieds de page doit être identique à ceux de ce document.

Conseils généraux :

- Mettre en lien les cours théoriques reçus à l'IUT et la pratique professionnelle.
- Si votre sujet s'y prête, faire un petit comparatif de la solution technique sur laquelle vous travaillez et les technologies concurrentes et montrez pourquoi votre solution est choisie.
- Relatez toutes les tâches effectuées pendant le stage. Regroupez les petites tâches de même thématique ensemble.

Référence bibliographique :

Ici je cite une référence bibliographique (Empson, April 17, 2005).

Glossaire :

Tous les mots soulignés sont expliqués dans le glossaire, par exemple métadonnées¹. Je souligne seulement la première fois où le mot est rencontré.

Sigles :

Les sigles sont détaillés la première fois qu'ils sont rencontrés, par exemple : **BUT, Bachelor Universitaire de Technologie.**

¹ Les mots soulignés dans le texte sont expliqués dans le glossaire. Merci de laisser cette note de bas de page

1 Introduction

Le stage s'est déroulé au sein du Laboratoire d'Aix-périmentation et de Bidouille, une structure spécialisée dans les projets technologiques, pédagogiques et expérimentaux autour des systèmes embarqués et du numérique éducatif. Les travaux réalisés durant cette période se sont principalement appuyés sur la carte STeaMi, une plateforme embarquée intégrant différents capteurs, un écran OLED ainsi que des fonctionnalités de communication sans fil.

La mission principale du stage était le développement et l'optimisation des fonctionnalités Bluetooth Low Energy (BLE) de la STeaMi. L'objectif était de concevoir des systèmes de communication fiables entre plusieurs cartes, d'implémenter un système de localisation par trilatération et multilatération BLE, et d'optimiser les émissions et scans afin de réduire la consommation énergétique et d'améliorer les performances réseau.

Le stage a également permis de travailler sur le développement en MicroPython et en C++/Arduino, l'exploitation de capteurs embarqués, l'optimisation de drivers graphiques, la conception d'interfaces utilisateur et la rédaction de ressources pédagogiques destinées aux ateliers organisés par le LAB.

Dans un premier temps, ce rapport présentera la structure d'accueil ainsi que l'environnement technique dans lequel le stage s'est déroulé. Ensuite, les différents projets réalisés autour de la STeaMi et des communications BLE seront détaillés en mettant en avant les problématiques rencontrées, les choix techniques effectués et les solutions développées. Enfin, une analyse des compétences acquises et des apports professionnels de cette expérience viendra conclure ce rapport.

2 Présentation de la structure d'accueil

2.1 Le LAB : Laboratoire d'Aix-périmentation et de Bidouille

Le Laboratoire d'Aix-périmentation et de Bidouille, aussi appelé **L.A.B**, est une association loi 1901 basée à Aix-en-Provence. Il s'agit d'un **Fab Lab** (laboratoire de fabrication numérique) dont l'objectif est de promouvoir la culture maker, le bricolage numérique et la médiation scientifique. Le LAB met à disposition un espace collaboratif où les membres peuvent expérimenter, apprendre, partager leurs connaissances et réaliser des projets techniques ou créatifs. L'association respecte également la charte des Fab Labs du MIT.

Le LAB dispose de plusieurs équipements de fabrication numérique comme des imprimantes 3D, une découpe laser, une fraiseuse CNC ou encore du matériel électronique.

2.2 Activités principales

Les principales activités du LAB sont :

- La fabrication numérique et le prototypage ;
- La programmation et l'électronique ;
- La conception d'objets et de robots ;
- L'impression 3D et la découpe laser ;
- L'organisation d'ateliers et de permanences collaboratives ;
- La participation à des projets éducatifs et scientifiques.

Le LAB organise également des événements appelés **GeekBidouille**, permettant aux passionnés de se rencontrer, d'échanger et de présenter leurs projets.

2.3 Objectifs pédagogiques et techniques

Le LAB a pour objectif de démocratiser l'accès aux technologies numériques et scientifiques. Il encourage l'apprentissage par la pratique, le travail collaboratif et l'innovation. L'association souhaite développer les compétences techniques des participants dans des domaines comme la programmation, l'électronique, la robotique ou la fabrication numérique.

Le LAB participe aussi à plusieurs projets pédagogiques autour des sciences, du numérique et de la robotique dans un cadre éducatif.

2.4 Public visé

Le LAB s'adresse à un public très varié :

- Étudiants ;
- Enseignants ;
- Chercheurs ;
- Entreprises et start-ups ;
- Artistes ;
- Passionnés de technologie et de DIY ;
- Scolaires et jeunes participant à des ateliers pédagogiques ;
- Toute personne curieuse de découvrir la fabrication numérique.

L'objectif est de créer une communauté ouverte où chacun peut apprendre, expérimenter et partager ses connaissances.

2.5 L'environnement technique du stage

Durant mon stage au sein du Laboratoire d'Aix-périmentation et de Bidouille (L.A.B), j'ai travaillé dans un environnement orienté développement embarqué, électronique et collaboration open source. Le travail réalisé reposait principalement sur l'utilisation de la carte **STEAMI**, utilisée dans différents projets pédagogiques et techniques du LAB.

Le développement s'effectuait dans un environnement collaboratif. L'ensemble des projets et du code source était hébergé sur les dépôts GitHub gérés par le LAB. Mes travaux étaient déposés directement sur ces dépôts via des *issues* permettant de suivre les tâches et via des *Pull Requests (PR)* afin de proposer et valider les modifications apportées au projet.

2.5.1 Présentation de la carte STEAMI

La carte STeaMi est une carte électronique programmable utilisée dans un cadre pédagogique et expérimental. Elle permet de développer des projets autour de la programmation, des objets connectés et de l'électronique embarquée.

Elle intègre plusieurs composants facilitant l'apprentissage et le prototypage, notamment des capteurs, un écran OLED ainsi qu'un module de communication Bluetooth Low Energy (BLE).



Figure 1 : STEaMi de face



Figure 2 : STEaMi de dos

2.5.2 Les capteurs intégrés

La carte STeaMi intègre plusieurs capteurs permettant de réaliser des projets autour de l'électronique, de la robotique et des objets connectés. Ces composants permettent de mesurer différentes informations provenant de l'environnement ou des mouvements de l'utilisateur.

Parmi les principaux capteurs présents sur la carte, on retrouve notamment :

- Un **accéléromètre** permettant de détecter les mouvements et les inclinaisons ;
- Un **gyroscope** utilisé pour mesurer les rotations ;
- Des capteurs environnementaux permettant d'obtenir certaines données physiques comme la température ou la luminosité ;
- Des boutons tactiles et entrées utilisateur.

Ces capteurs rendent la carte adaptée à des projets pédagogiques variés comme la création d'objets interactifs, la collecte de données ou encore le développement de systèmes embarqués.

2.5.3 Écran OLED

La carte STTeaMi possède un écran **OLED circulaire 128×128 pixels** intégré directement sur la carte. La technologie OLED (*Organic Light-Emitting Diode*) permet un affichage lumineux, contrasté et peu énergivore, adapté aux systèmes embarqués.

Cet écran peut afficher :

- Des valeurs issues des capteurs ;
- Des menus de navigation ;
- Des informations de débogage ;
- Des éléments graphiques simples.

Il facilite les phases de test et de développement en offrant un retour visuel immédiat sans nécessiter de connexion permanente à un ordinateur.

2.5.4 Communication BLE

La carte STEAMI intègre une connectivité **Bluetooth Low Energy (BLE) 5.2** via le microcontrôleur STM32WB55RG.

Le BLE est une technologie de communication sans fil conçue pour consommer très peu d'énergie tout en permettant l'échange de données entre plusieurs appareils.

Cette technologie permet des échanges avec un ordinateur, un smartphone, une tablette ou d'autres dispositifs compatibles Bluetooth.

Elle est particulièrement utilisée dans les objets connectés et les systèmes embarqués nécessitant une communication légère et économe en énergie.

2.5.5 Environnement logiciel

Le développement réalisé durant le stage s'appuyait sur plusieurs outils et technologies.

MicroPython

MicroPython est une version allégée du langage Python conçue pour les microcontrôleurs.

Il permet de programmer la carte STTeaMi de manière simple tout en ayant accès aux composants matériels (capteurs, écran, communication BLE).

Il est particulièrement adapté au prototypage rapide et à l'apprentissage de la programmation embarquée.

Arduino / C++

L'environnement Arduino en C++ a également été utilisé pour certains développements.

Il permet un contrôle plus bas niveau du matériel et une meilleure optimisation des performances.

Cette approche est couramment utilisée pour des applications nécessitant une gestion plus fine des ressources et des interactions matérielles.

3 Développement des projets techniques

3.1 Mise en place et prise en main de la STTeaMi

La prise en main de la carte STTeaMi a nécessité la mise en place d'un environnement de développement adapté aux systèmes embarqués.

L'environnement reposait sur plusieurs outils complémentaires :

- **Thonny**, utilisé principalement pour l'exécution et le débogage en MicroPython ;
- **mpremote (REPL)**, permettant l'interaction directe avec la carte via une console Python ;
- Un **environnement virtuel Python (venv)** afin d'isoler les dépendances et garantir un environnement de travail propre et reproductible.

La carte était connectée en USB, ce qui permettait le transfert de code, l'exécution de scripts et l'accès au REPL MicroPython pour des tests rapides.

Cette configuration a permis de disposer d'un environnement flexible, adapté à la fois au développement, au test rapide et au débogage embarqué.

3.1.1 Installation de l'environnement de développement sur Debian

L'environnement de développement a été mis en place sur une machine sous **Debian**, afin de permettre la programmation de la carte STeaMi, les tests MicroPython ainsi que la communication via USB.

Installation de Python et création de l'environnement virtuel

La première étape consiste à installer **Python 3** ainsi que les outils nécessaires au développement et à la gestion des environnements virtuels.

La seconde étape a pour but de créer un environnement virtuel afin d'isoler les dépendances et garantir un développement propre et reproductible.

```
sudo apt update  
sudo apt install python3 python3-pip python3-venv -y
```

Création et activation de l'environnement virtuel :

```
python3 -m venv steami-env  
source steami-env/bin/activate
```

Installation de mpremote

L'outil **mpremote** permet de communiquer avec la carte STeaMi via USB et d'accéder au REPL MicroPython.

```
pip install mpremote
```

Installation de Thonny

Thonny a été utilisé comme environnement léger pour l'exécution et le débogage de code MicroPython.

```
sudo apt install thonny -y
```

Il permet notamment de :

- Téléverser des scripts sur la carte ;
- Exécuter du code MicroPython ;
- Observer les sorties du programme en temps réel.

3.1.2 Premiers tests BLE et advertisements

Les premiers tests réalisés sur la carte STTeaMi ont porté sur le module Bluetooth Low Energy (BLE) afin de vérifier son bon fonctionnement et sa capacité à diffuser des données sans fil.

La mise en place s'est appuyée sur la bibliothèque native bluetooth disponible en MicroPython, qui permet un accès direct au contrôleur BLE du STM32WB55.

Le premier objectif était de tester le mode advertising, permettant à la carte d'émettre des trames BLE détectables par des appareils externes (smartphone, ordinateur ou scanner BLE).

```
import bluetooth

ble = bluetooth.BLE()
ble.active(True)

name = b"STTeaMi"
adv = bytes([2, 0x01, 0x06, len(name) + 1, 0x09]) + name

ble.gap_advertise(100_000, adv)
```

Ce test a permis de vérifier :

- La visibilité de la carte via un scanner BLE ;
- La diffusion correcte du nom et des trames publicitaires ;
- Le bon fonctionnement du module BLE intégré.

3.2 Développement des communications Bluetooth Low Energy

Cette partie présente la mise en place et l'exploitation des communications Bluetooth Low Energy (BLE) sur la carte STTeaMi.

L'objectif principal était de comprendre le fonctionnement du BLE dans un contexte embarqué, puis de l'utiliser pour permettre des échanges de données entre plusieurs cartes.

Le BLE a été utilisé à la fois pour la diffusion d'informations (advertising) et pour la réception de données provenant d'autres dispositifs compatibles.

3.2.1 Fonctionnement du BLE sur la STTeaMi

Le Bluetooth Low Energy (BLE) est un protocole de communication sans fil optimisé pour les faibles consommations énergétiques et les échanges de données à courte portée. Il est largement utilisé dans les systèmes embarqués et les objets connectés.

Sur la carte STTeaMi, le BLE repose sur une architecture standard définie par la couche Bluetooth Low Energy, comprenant notamment les mécanismes d'**advertising** et de **scanning**.

Dans une architecture BLE classique, les échanges structurés passent par le protocole **GATT (Generic Attribute Profile)**, basé sur une relation client/serveur et l'utilisation de services et caractéristiques.

Cependant, dans le cadre de ce projet, le fonctionnement repose principalement sur un mode **sans connexion**, basé sur la diffusion de paquets (advertising) et leur écoute (scan), sans établissement de session GATT complète.

Ce choix permet :

- Une communication plus légère ;
- Une latence réduite ;
- Une meilleure scalabilité pour des échanges entre plusieurs cartes.

Les données sont encapsulées dans les trames d'advertising BLE, qui suivent un format standard composé de champs (type, longueur, valeur), permettant d'inclure un nom de périphérique et des données personnalisées.

Ce mode de fonctionnement constitue la base des communications entre cartes et sera exploité dans les systèmes multi-nœuds développés par la suite.

3.2.2 Mise en place d'un réseau BLE multi-cartes

Dans le cadre du stage, un prototype de réseau BLE multi-cartes a été développé afin d'expérimenter la communication distribuée entre plusieurs cartes STeAMi.

L'objectif du projet était de mettre en place une architecture maillée (*mesh network*) composée de :

- Trois périphériques capteurs (P1, P2, P3) ;
- Trois relais de communication (R1, R2, R3).

Chaque périphérique mesure une distance via son capteur embarqué puis transmet cette information à son relais associé grâce au Bluetooth Low Energy (BLE). Les relais communiquent ensuite entre eux afin de redistribuer les messages dans le réseau.

Le fonctionnement repose principalement sur :

- Le mode **advertising BLE** ;
- Le **scan continu** des paquets BLE ;
- Une retransmission multi-sauts entre les relais.

Les données échangées étaient encapsulées dans des trames JSON contenant :

- L'émetteur ;
- Le destinataire ;
- La donnée transmise ;
- Un compteur de saut (*hop counter*) permettant d'éviter certaines boucles de retransmission.

Un système simple de chiffrement XOR a également été ajouté afin de sécuriser minimalement les échanges entre les différents nœuds du réseau.

Fonctionnement général du réseau

1. Un périphérique lit une valeur depuis son capteur de distance ;
2. La donnée est chiffrée puis envoyée via BLE ;
3. Le relais associé reçoit le message et le retransmet aux autres relais ;
4. Les autres périphériques reçoivent ensuite la donnée propagée dans le réseau.

Chaque carte pouvait également afficher des informations locales sur l'écran OLED intégré, notamment l'état des communications et les distances reçues.

Ce projet avait pour objectif d'explorer les principes des réseaux distribués embarqués, de la propagation multi-sauts et de la communication BLE entre plusieurs nœuds indépendants.

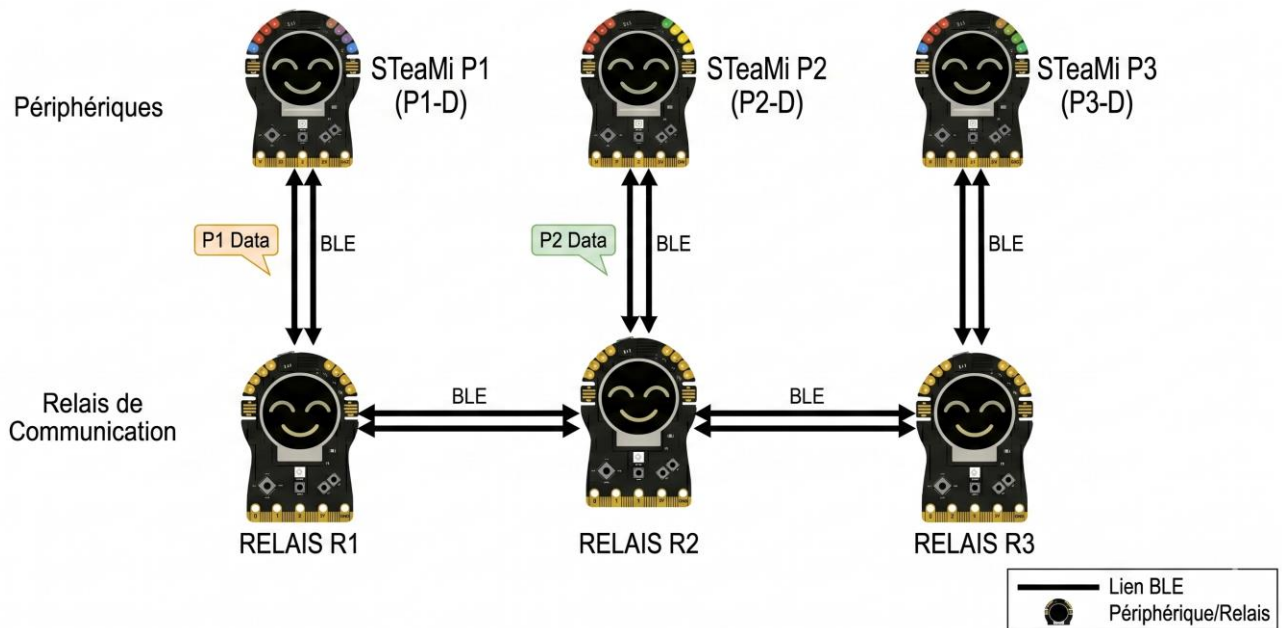


Figure 3 : Réseaux mesh BLE

3.2.3 Système de localisation par trilatération BLE

Dans le cadre du stage, un système de localisation intérieure (*indoor positioning*) basé sur le Bluetooth Low Energy (BLE) a été développé sur les cartes STeaMi.

L'objectif du projet était d'estimer la position d'une carte mobile à partir de la puissance du signal radio reçu (RSSI) provenant de plusieurs balises fixes.

Principe du modèle log-distance (RSSI -> distance)

Le système repose sur le modèle de propagation radio appelé *log-distance path loss model*. Ce modèle permet d'estimer une distance à partir de la puissance du signal BLE reçu.

La formule utilisée est la suivante :

$$d = 10^{\frac{RSSI_{ref} - RSSI}{10n}}$$

avec :

- (RSSI) : puissance du signal mesuré ;
- (RSSI_{ref}) : puissance mesurée à 1 mètre ;
- (n) : coefficient de propagation dépendant de l'environnement.

Chaque balise a été calibrée manuellement afin d'obtenir une estimation de distance plus stable dans l'environnement de test.

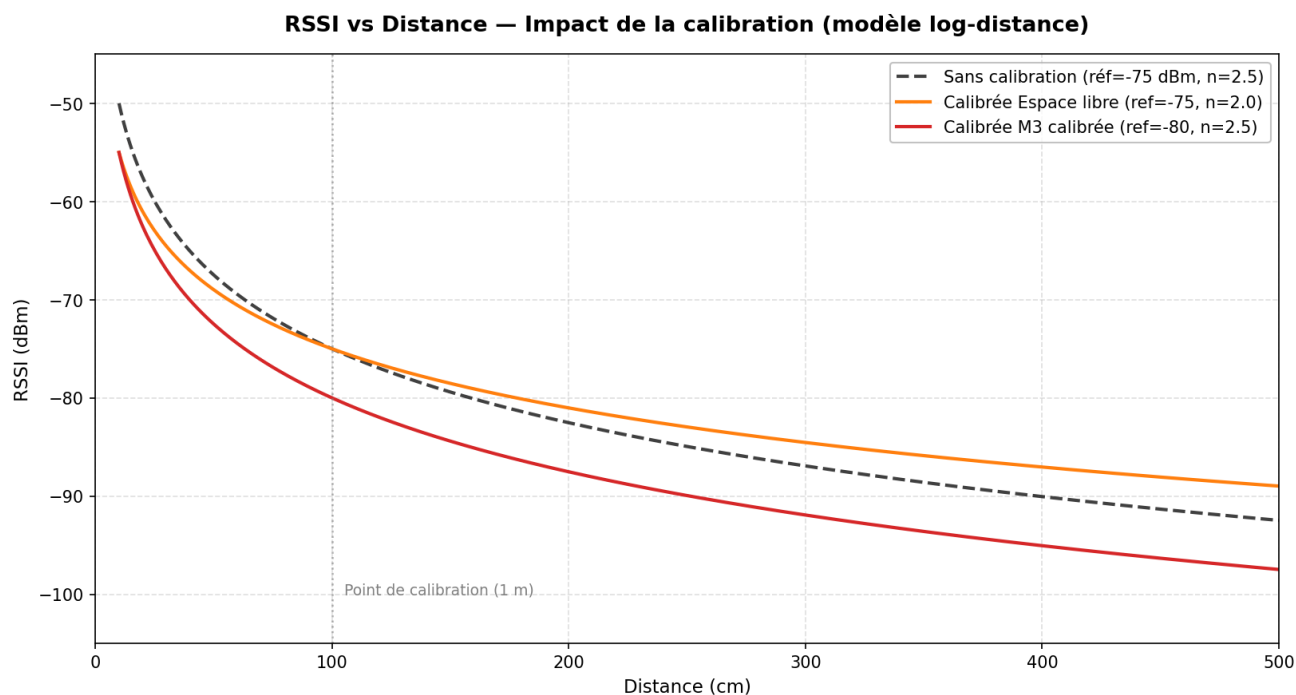


Figure 4 : Courbe représentative du Path Loss Model selon les milieux

Architecture du système

Le système est composé de :

- Trois cartes fixes utilisées comme ancres BLE (STeaMi-M1, STeaMi-M2, STeaMi-M3) ;
- Une carte mobile appelée STeaMi-Nomade.

Les cartes ancres diffusent continuellement leur identifiant via BLE grâce au mode advertising. La carte mobile scanne les signaux reçus, mesure le RSSI de chaque balise puis calcule une estimation de distance.

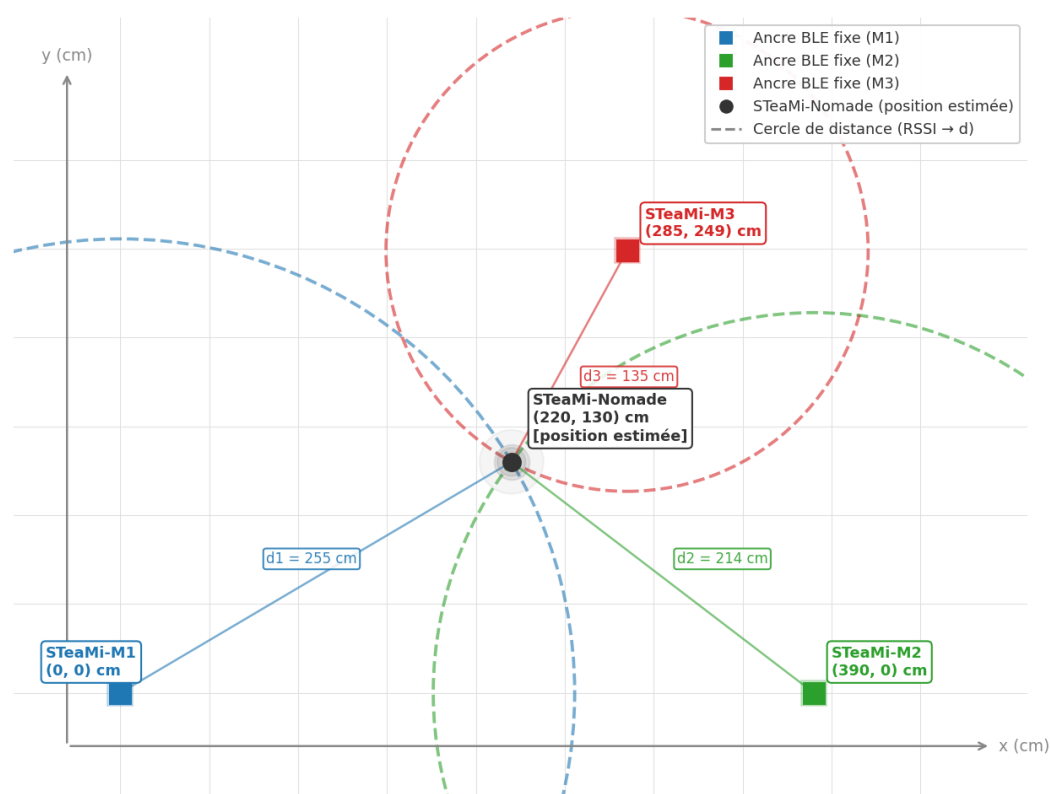


Figure 5 : schéma de l'architecture du système de trilatération BLE.

Algorithme de trilatération

Une fois les distances estimées pour les trois balises, un algorithme de trilatération 2D est utilisé afin de calculer la position de la carte mobile dans l'espace.

Le calcul repose sur l'intersection de trois cercles centrés sur les balises BLE.

Afin de limiter les variations importantes du RSSI, plusieurs techniques de filtrage ont été expérimentées :

- Moyenne glissante ;
- Filtrage exponentiel ;
- Filtre de Kalman ;
- Filtre particulaire.

Le filtrage exponentiel a été retenu pour sa simplicité et son bon compromis entre stabilité et réactivité sur MicroPython.

Interface OLED

L'écran OLED intégré à la carte STeaMi permet d'afficher en temps réel :

- Les balises détectées ;
- Les distances estimées ;
- La position calculée de la carte mobile ;
- Une représentation graphique simplifiée de l'espace.

La position estimée est affichée sous forme de point mobile sur une carte 2D représentant la zone de test.

Extrait de code de trilatération

```
def rssi_to_distance(rssi, beacon_name):  
  
    ref = RSSI_REF.get(beacon_name, -75)  
    n = PATH_LOSS_N.get(beacon_name, 2.5)  
  
    d_m = 10 ** ((ref - rssi) / (10 * n))  
  
    return d_m * 100
```

Résultats obtenus et limites du système

Les expérimentations réalisées ont permis d'obtenir une précision moyenne de l'ordre de 20 cm dans un environnement intérieur contrôlé.

Cependant, plusieurs limitations ont été observées :

- Forte variabilité du RSSI ;
- Perturbations liées à l'environnement ;
- Instabilité des mesures radio ;
- Limitations de performance de MicroPython.

En effet, MicroPython étant un langage interprété, de nombreuses opérations introduisent de la latence :

- Traitement asynchrone ;
- Calculs de filtrage ;
- Gestion BLE ;
- Affichage OLED via SPI.

Par exemple, certaines mises à jour de l'écran pouvaient prendre plusieurs millisecondes, ce qui impactait directement la fréquence de rafraîchissement des calculs.

Le système reste néanmoins adapté à des usages nécessitant une localisation approximative ou une détection de présence dans une zone, plutôt qu'un positionnement absolu de haute précision.

Une extension envisagée du projet consistait à mettre en place une trilatération en trois dimensions (3D) afin d'obtenir une estimation plus complète de la position dans l'espace.

Cependant, les limitations de performances observées sous MicroPython ont conduit à privilégier une future implémentation en C++, permettant des calculs plus rapides, une meilleure gestion temps réel et une fréquence d'actualisation plus élevée.

3.3 Développement de projets ludiques et algorithmiques

Dans le cadre du stage, plusieurs projets à vocation ludique et pédagogique ont été développés afin d'exploiter les capteurs de la carte STeaMi tout en illustrant des concepts algorithmiques de base ou avancés.

Ces projets avaient également pour objectif de proposer des supports réutilisables dans un contexte éducatif, notamment pour l'apprentissage d'algorithmes classiques en NSI.

3.3.1 Maze Game et algorithme de Dijkstra

Un jeu de labyrinthe interactif a été développé en exploitant l'accéléromètre **ISM330DL** et l'écran OLED **SSD1327** de la carte STeaMi.

Le joueur contrôle un personnage en inclinant physiquement la carte, chaque inclinaison étant interprétée comme un déplacement dans une grille.

Le labyrinthe est généré dynamiquement à l'aide d'un algorithme de type **profondeur d'abord (DFS / Recursive Backtracker)** garantissant un chemin unique entre le départ et l'arrivée.

Afin d'évaluer la performance du joueur, un algorithme de plus court chemin basé sur **Dijkstra** est appliqué sur le labyrinthe généré. Celui-ci permet de calculer le nombre minimal de déplacements nécessaires pour atteindre l'objectif.

Le score final est ensuite calculé en fonction de l'écart entre le chemin optimal et le chemin réellement parcouru.

Principe de Dijkstra utilisé

$\text{score} = f(\text{steps}, \text{optimal_steps})$

Filtrage et optimisation

Pour garantir une expérience fluide, la détection des mouvements a été ajustée via un seuil de tilt afin d'éviter les déplacements parasites liés aux variations de l'accéléromètre.

Interface et affichage OLED

L'écran OLED permet de visualiser en temps réel :

- La grille du labyrinthe ;
- La position du joueur ;
- La destination ;
- Les informations de score et de performance.

Une représentation graphique simple est utilisée afin de rester lisible sur un écran de petite taille.

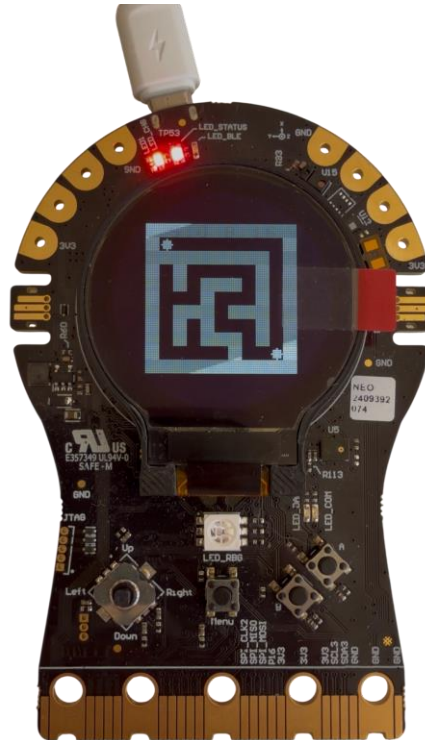


Figure 6 : Affichage Maze Game

Résultats et intérêt pédagogique

Ce projet permet d'illustrer plusieurs notions importantes :

- Génération procédurale de labyrinthe ;
- Recherche de plus court chemin (Dijkstra) ;
- Interaction capteur → mouvement ;
- Visualisation embarquée sur écran OLED.

Il a également été conçu comme une base pédagogique exploitable pour l'enseignement des algorithmes en NSI.

3.4 Optimisation logicielle et migration vers le C++

Dans une seconde phase du projet, une partie des développements a été optimisée en migrant certaines briques logicielles de MicroPython vers le langage C++. L'objectif principal était d'améliorer les

performances, la réactivité du système et la stabilité des calculs temps réel, notamment pour le système de localisation par trilatération BLE.

3.4.1 Comparaison MicroPython / C++

MicroPython est un langage interprété, conçu pour simplifier le développement sur systèmes embarqués. Il permet un prototypage rapide et une grande simplicité d'utilisation, mais introduit une surcharge d'exécution liée à l'interprétation du code.

À l'inverse, le C++ est un langage compilé offrant un contrôle plus fin sur le matériel et une exécution beaucoup plus rapide. Il est particulièrement adapté aux applications embarquées nécessitant des performances élevées et une gestion optimisée des ressources.

Dans le cadre du projet, MicroPython a été utilisé pour la phase de prototypage, tandis que le C++ a été privilégié pour les parties critiques en termes de performance.

3.4.2 Optimisation mémoire et performances

Les optimisations réalisées ont principalement porté sur :

- La réduction des traitements inutiles dans la boucle principale ;
- La limitation des allocations dynamiques ;
- L'amélioration de la fréquence de calcul des données BLE ;
- La diminution de la latence globale entre acquisition et affichage.

Ces optimisations ont permis d'améliorer significativement la stabilité du système de localisation en temps réel.

3.4.3 Portage du système de localisation en C++ : gains concrets

Le système de localisation basé sur la trilatération BLE a été porté en C++ afin d'améliorer les performances globales par rapport à l'implémentation MicroPython.

Le code reprend les mêmes principes : filtrage du RSSI, conversion RSSI → distance via le modèle log-distance, trilatération 2D et filtrage exponentiel de la position.

$$d = 10 \frac{RSSI_{ref} - RSSI}{10n} \quad d = 10 \frac{10 - RSSI}{10n}$$

Cette version en C++ permet une exécution beaucoup plus rapide et plus stable des calculs, ce qui améliore la réactivité du système et réduit les variations de position.

La précision obtenue est de l'ordre de quelques centimètres dans des conditions contrôlées, avec une stabilité nettement supérieure à la version MicroPython.

En revanche, l'affichage OLED n'a pas encore été porté en C++ (absence de driver SSD1327), et les résultats sont actuellement affichés via le terminal série.

4 Participation aux activités pédagogiques

4.1 Ateliers scolaires CE2 / CM1

Lors des ateliers scolaires réalisés à l'école Marcel Pagnol à Aix-en-Provence, j'ai participé à l'animation et à l'encadrement d'élèves de CE2 et CM1 autour d'activités de découverte de l'informatique et de la logique algorithmique.

Mon rôle consistait principalement à :

- Accompagner les élèves dans la compréhension des consignes,
- Les aider dans la réalisation des activités pratiques,
- Expliquer de manière simplifiée des notions techniques abordées.

Par exemple, la notion de **bit de parité**, issue des codes correcteurs d'erreurs étudiés en BUT Réseaux et Télécommunications, a été vulgarisée afin d'être accessible à un public jeune. Cela a permis d'introduire de manière intuitive les notions de fiabilité et de transmission de l'information.

Activités pédagogiques réalisées

4.1.1 Scratch déconnecté

Les activités de type Scratch déconnecté ont permis aux élèves de comprendre les bases de l'algorithmique sans ordinateur, en manipulant des instructions simples et des séquences logiques.

4.1.2 Bits de parité

L'activité sur les bits de parité a introduit la notion de détection d'erreurs dans les messages. Les élèves ont compris qu'un message peut être vérifié grâce à un bit supplémentaire indiquant la parité des données.

4.1.3 Tours de Hanoï

Le puzzle des Tours de Hanoï a été utilisé pour travailler la logique de résolution de problèmes et la notion de récursivité de manière intuitive.

4.2 Journée ITER Robots — Stand STeaMi

Lors de la journée ITER Robots, la STeaMi a été présentée sur un stand dédié afin de faire découvrir la STeaMi à un public composé principalement d'écopiliers, d'enseignants et de professionnels du secteur éducatif et scientifique.

L'objectif de cette participation était de valoriser les travaux du laboratoire et de présenter les possibilités pédagogiques offertes par la carte STeaMi dans des contextes d'apprentissage de l'informatique et des systèmes embarqués.

Les démonstrations ont suscité un fort intérêt de la part des visiteurs. De nombreux échanges ont permis de présenter les différents projets réalisés ainsi que les usages possibles en milieu scolaire.

Cette journée a également permis de recueillir plusieurs contacts intéressés par la démarche, contribuant ainsi à renforcer la visibilité du laboratoire et de ses projets.



Figure 7 : Stand STeaMi lors de la journée ITER Robots

4.3 Création de ressources pédagogiques

Dans le cadre du projet InnovMicro-EXAO #2, j'ai participé à la création et à l'enrichissement de ressources pédagogiques destinées à la plateforme du laboratoire. Ces ressources ont été publiées sous forme de fiches accessibles via le wiki du projet.

Ces fiches pédagogiques ont pour objectif de faciliter la prise en main de la carte STeaMi et de ses différentes fonctionnalités, notamment les capteurs, les actionneurs et certains modules de communication.

Chaque ressource est conçue de manière progressive et pédagogique afin d'être compréhensible par un public jeune, à partir d'environ 10 ans. Le contenu est volontairement simplifié et illustré afin de rendre les notions techniques accessibles.

Enfin, chaque fiche propose des pistes d'amélioration ou d'approfondissement permettant d'aller plus loin dans la compréhension et d'encourager la réflexion autonome.

5 Bilan du stage

5.1 Compétences techniques acquises

5.2 Difficultés rencontrées et solutions apportées

5.3 Apports professionnels et personnels

6 Conclusion

Ce stage au sein du LAB m'a permis de mener à bien des projets techniques ambitieux, au cœur desquels le développement d'un système de localisation par trilatération BLE constitue la réalisation la plus aboutie. De la prise en main de la STeaMi jusqu'au portage en C++ du système de localisation, chaque étape a représenté un défi technique concret, résolu avec les outils et méthodologies propres au monde de l'embarqué.

Au-delà de l'aspect purement technique, ce stage m'a confronté à une réalité professionnelle plus large : la transmission et la vulgarisation. Que ce soit lors des ateliers scolaires CE2/CM1, du stand ITER Robots ou de l'adaptation des fiches pédagogiques, j'ai dû adapter mon discours et mes productions à des publics très différents, ce qui constitue une compétence essentielle dans le milieu de la médiation technologique.

Ce stage m'a ainsi confirmé mon attrait pour les systèmes embarqués et les communications sans fil, tout en m'ouvrant sur une dimension pédagogique que je n'anticipais pas et qui s'est révélée particulièrement enrichissante.

Il faut que cela soit sur une **page impaire**
Car le document sera imprimé en recto-verso.

PAGE BLANCHE

Le document sera imprimé en recto-verso, donc il ne doit pas y avoir de texte sur cette feuille.
Et il faut supprimer ces 3 lignes !

7 Remerciements

Je tiens tout d'abord à remercier **Sébastien Nedjar**, mon maître de stage, pour m'avoir accueilli au sein du LAB et encadré tout au long de cette expérience, en me faisant confiance sur des projets techniques exigeants.

Je remercie également **Manon Ballester** pour sa collaboration lors des ateliers scolaires CE2/CM1 ainsi que pour son implication dans le travail d'adaptation des fiches pédagogiques.

Une pensée particulière pour **Kaan Ozen**, collègue et partenaire de travail, avec qui j'ai partagé de nombreux moments tout au long du stage, tant sur le plan technique qu'humain.

Je remercie aussi **Aline Dumont** pour notre collaboration sur les fiches pédagogiques et pour avoir co-animé le stand STeaMi lors de la journée ITER Robots au collège Marie Mauron de Pertuis.

Je souhaite remercier **Arnaud Fevrier**, mon tuteur pédagogique à l'IUT, pour le suivi assuré durant cette période.

Enfin, je tiens à exprimer ma reconnaissance à **Yannick Marietti**, sans qui je n'aurais jamais eu connaissance du LAB ni obtenu cette opportunité de stage.

Il faut que cela soit sur une **page impaire**
Car le document sera imprimé en recto-verso.

PAGE BLANCHE

Le document sera imprimé en recto-verso, donc il ne doit pas y avoir de texte sur cette feuille.
Et il faut supprimer ces 3 lignes !

8 Glossaire

BLE — Bluetooth Low Energy, protocole de communication sans fil basse consommation dérivé du Bluetooth classique, conçu pour les objets connectés et les systèmes embarqués.

RSSI — Received Signal Strength Indicator, indicateur de la puissance du signal radio reçu, exprimé en dBm. Utilisé dans ce projet pour estimer la distance entre les cartes.

Advertisement — Trame de diffusion BLE émise périodiquement par un appareil pour signaler sa présence sans établir de connexion.

OLED — Organic Light-Emitting Diode, technologie d'affichage utilisée sur la STeaMi pour l'interface utilisateur.

MicroPython — Implémentation allégée du langage Python conçue pour fonctionner sur microcontrôleurs à ressources limitées.

Beacon — Dispositif ou carte configuré pour émettre des advertisements BLE de manière continue, utilisé comme point de référence dans le système de localisation.

DAPLink — Interface microprogrammée intégrée à la STeaMi permettant la programmation et le débogage du microcontrôleur via USB, sans programmeur externe.

Firmware — Programme embarqué bas niveau stocké dans la mémoire flash du microcontrôleur, constituant la couche logicielle fondamentale du système.

IoT — Internet of Things, désigne l'ensemble des objets connectés capables de collecter et d'échanger des données via un réseau.

STeaMi — Carte de développement embarqué basée sur le microcontrôleur STM32WB55, conçue par le LAB pour des usages pédagogiques et intégrant BLE, capteurs, écran OLED et interface USB.

PAGE BLANCHE

Le document sera imprimé en recto-verso, donc il ne doit pas y avoir de texte sur cette feuille.
Et il faut supprimer ces 3 lignes !

9 Bibliographie

Python Software Foundation. (n.d.). *Python Documentation*. <https://www.python.org/>
MicroPython Documentation. (n.d.). *mpremote reference*.
<https://docs.micropython.org/en/latest/reference/mpremote.html>
Olimex Ltd. (n.d.). *Thonny Python IDE*. <https://thonny.org/>
LAB Aix Bidouille (2026). *Catalogue InovMicro-EXAO*.
<https://wiki.labaixbidouille.com/catalogue?proj=inovmicro-exao>

Ici j'écris en **Style Normal**



Figure 1 : Ceci est un switch