



1/20/2022

# Projet: Dispositif BLE Beacon



**Réalisé par :** Ayele ADAMA / EL-HOMAINI  
ISSA

**Tuteur:** Mr Eric PIGEON

## Remerciements

Nous voudrions tout d'abord adresser toute notre gratitude à M. Éric Pigeon, notre tuteur et également client, pour son encadrement et sa disponibilité et précieux conseils tout au long du projet.

Un merci particulier à M. Basile DUFAY, pour ses apports et ses conseils tout au long du projet.

Merci à STMicroelectronics pour l'approvisionnement en matériel.

Merci également à l'ESIX Normandie pour la mise à disposition de la salle projet

## Table des matières

1. Introduction.....	5
2. Missions effectuées .....	7
2.1. Développement de la bibliothèque .....	7
2.1.1. Prise en main du matériel et des logiciels .....	7
2.1.2. Communication entre les deux processeurs (CPU) du microcontrôleur (MCU) 10	
2.1.3. Différence entre le Bluetooth classique et BLE .....	11
2.1.4. Couches du BLE .....	12
2.1.5. Communication BLE .....	13
2.1.6. Les services et caractéristiques .....	14
2.1.7. Trame Beacon .....	16
2.1.8. Bibliothèque développée.....	17
2.1.9. Application .....	20
2.1.10. Site Web .....	22
2.2. Dimensionnement énergétique.....	25
2.3. Conception du boîtier .....	26
3- Conclusion.....	28

# TABLE DES FIGURES

Figure 1: Carte du pack Nucléo STM32WB55 .....	8
Figure 2: Dongle USB du pack Nucléo STM32WB55 .....	9
Figure 3: Fonctionnement des deux processeurs .....	10
Figure 4: Différents types de Bluetooth .....	11
Figure 5: Couches du BLE .....	12
Figure 6: Communication BLE .....	13
Figure 7: Etapes de la communication BLE .....	14
Figure 8: Hiérarchie service et caractéristique .....	15
Figure 9: Exemple de service et de caractéristiques .....	15
Figure 10: trame Beacon .....	16
Figure 11: Service et caractéristiques de notre projet .....	18
Figure 12: Code permettant de changer la trame .....	19
Figure 13: Applications Android développées .....	21
Figure 14: App pour la configuration de la balise .....	21
Figure 15: App pour la configuration de la balise .....	22
Figure 16: Interface du site web .....	23
Figure 17: Connexion à la balise par Bluetooth .....	23
Figure 18: Configuration de la balise .....	24
Figure 19: Balise configurée comme iBeacon .....	25
Figure 20: La conception du boîtier de la balise .....	26
Figure 21: Porte de la balise .....	27
Figure 22: Assemblage du Boîtier .....	27

# TABLE DES TABLEAUX

Tableau 1: Objectifs du projet .....	6
Tableau 2: Contraintes imposées .....	6
Tableau 3: Valeurs pour la fonction <code>aci_hal_set_tx_power_level()</code> .....	20

# 1. Introduction

Nous avons effectué notre projet supervisé de la troisième année du cycle ingénieur à l'ESIX, quelques heures par semaine du 20 octobre 2021 au 21 janvier 2022 et tous les jours du 21 janvier au 04 mars 2022. Durant ce projet supervisé, nous avons travaillé sur le projet Dispositif BLE Beacon qui est une partie du projet Vikazimut.

Vikazimut est un projet d'étudiants en informatique de l'ENSICAEN et de l'université de Caen Normandie. Il répond à une demande de l'association Vik'Azim visant à la réalisation d'une application mobile pour la pratique de la course d'orientation. Le projet est composé d'un site web et d'une application mobile. Un parcours d'orientation consiste en une suite de points de contrôles matérialisés par sur le terrain par une balise type fédération internationale de course d'orientation contenant en plus un code QR et un tag NFC. L'orienteur utilise l'application pour se repérer à partir de la carte et valider son passage aux points de contrôle avec le lecteur de code QR ou le lecteur NFC si le parcours est équipé de balises physiques ou par détection automatique de la position GPS.

Le projet Dispositif BLE Beacon consiste à intégrer la technologie Beacon aux points de contrôles. Cette technologie basée sur le BLE (Bluetooth Low Energy) permettra à la balise d'envoyer des signaux à intervalles de temps réguliers. Le téléphone de l'orienteur capte ce signal dès qu'il est proche de la balise et l'application se charge de valider son passage au point de contrôle correspondant.

Notre travail consiste à développer la bibliothèque permettant à la balise d'envoyer des signaux ou trames à un intervalle de temps donné, de se connecter au smartphone du coureur via Bluetooth à une distance donnée et de valider le point de contrôle. Le Beacon développé doit pouvoir être configuré par les utilisateurs. Après développement, la bibliothèque devra être intégrée à l'application mobile et le site web du projet Vikazimut. L'intégration de la bibliothèque ne fait pas partie de nos tâches.

Notre travail consiste également à dimensionner le stockage énergétique, à concevoir un boîtier, intégrable au format actuel des bornes, qui contiendra le dispositif.

Les objectifs dégagés sont regroupés dans le tableau suivant.

Objectifs	Eléments d'évaluation ou livrable	Critères de succès
Développement d'une bibliothèque en C permettant la configuration de la balise	Des fichiers écrits en langage C	Cette bibliothèque doit permettre de configurer tous les paramètres de la balise BLE Beacon (valeur de la portée, les informations à partager, ...)
Etude et dimensionnement énergétique	Intégration d'une alimentation ayant une autonomie de 2 à 3 ans	La balise doit avoir une autonomie de 2 à 3 ans
Concevoir un boîtier pour le beacon	Un fichier 3D Un boîtier étanche conçu en 3D	Le boîtier doit pouvoir contenir le beacon, doit être étanche et doit pouvoir être intégré au format actuel de la balise

*Tableau 1: Objectifs du projet*

Certaines contraintes nous ont également été imposées par le client. Elles sont regroupées dans le tableau suivant.

Axe	Contraintes	Limites
Contenu du projet	Intégrer notre produit au format actuel des balises	Forme et taille du format actuel des balises
Temporel	Rendre le projet dans les délais	04/03/2022
Qualité	Autonomie du dispositif	2 ans minimum

*Tableau 2: Contraintes imposées*

Dans ce rapport, les missions effectuées à savoir le développement de la bibliothèque, le dimensionnement énergétique et la conception du boîtier sont exposées et les résultats sont évalués.

## 2. Missions effectuées

### 2.1. Développement de la bibliothèque

#### 2.1.1. Prise en main du matériel et des logiciels

Le matériel utilisé pour le projet est le pack Nucléo STM32WB55 composé d'une carte (figure 1) et d'un dongle USB (figure 2). Ils contiennent des microcontrôleurs de la série STM32WB55 : STM32WB55RGV6 pour la carte et STM32WB55CGU6 pour le dongle USB.

Les microcontrôleurs STM32WB55 sont équipés d'un processeur Arm 64 bits Cortex M4 et d'un processeur Arm 32 bits Cortex M0+ dédié pour la couche radio en temps réel. Ils sont également équipés du protocole IEEE 802.15.4 et du protocole Bluetooth Low Energy 5.0 qui est le protocole utilisé pour la technologie Beacon.

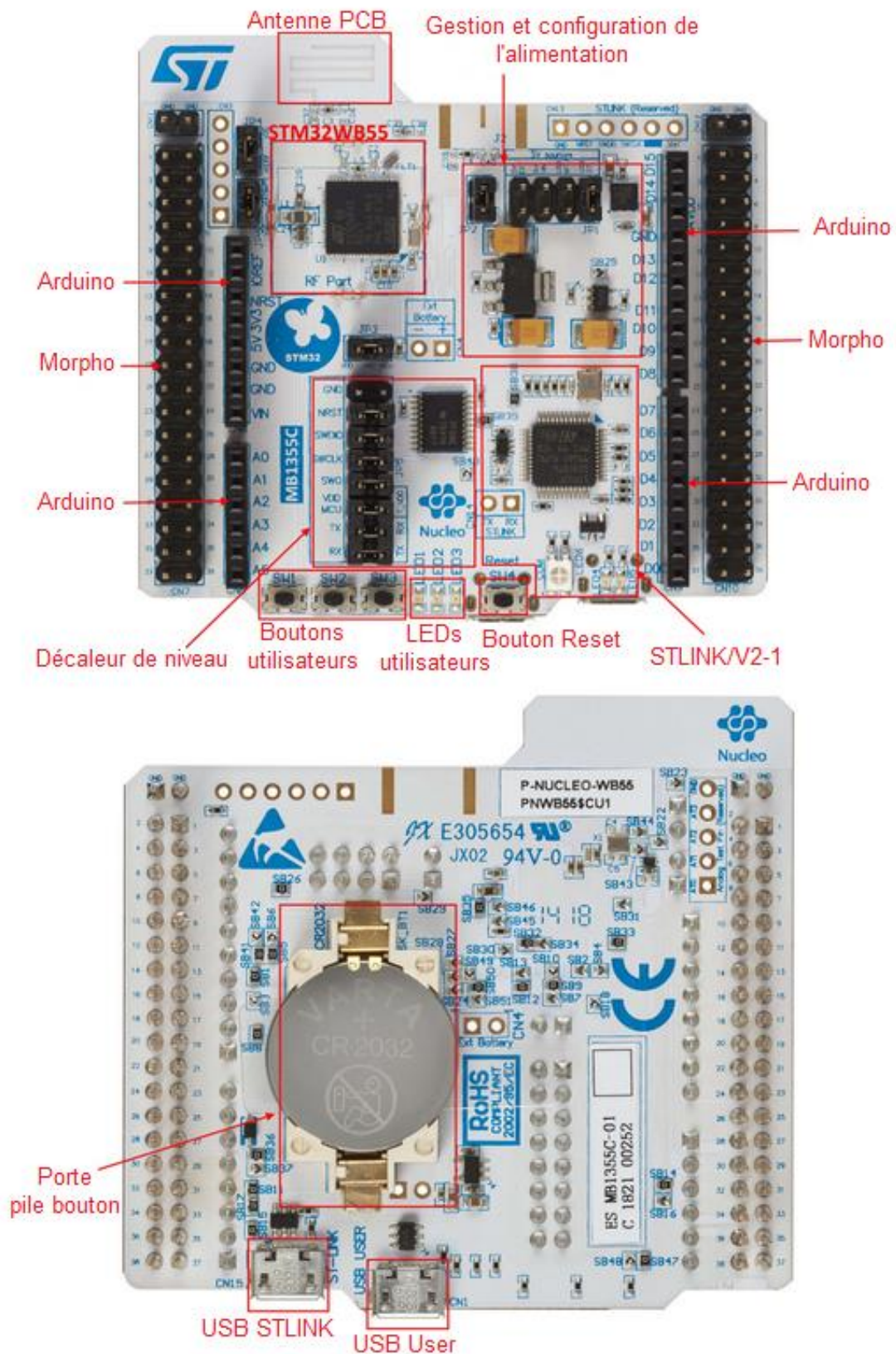


Figure 1: Carte du pack Nucléo STM32WB55



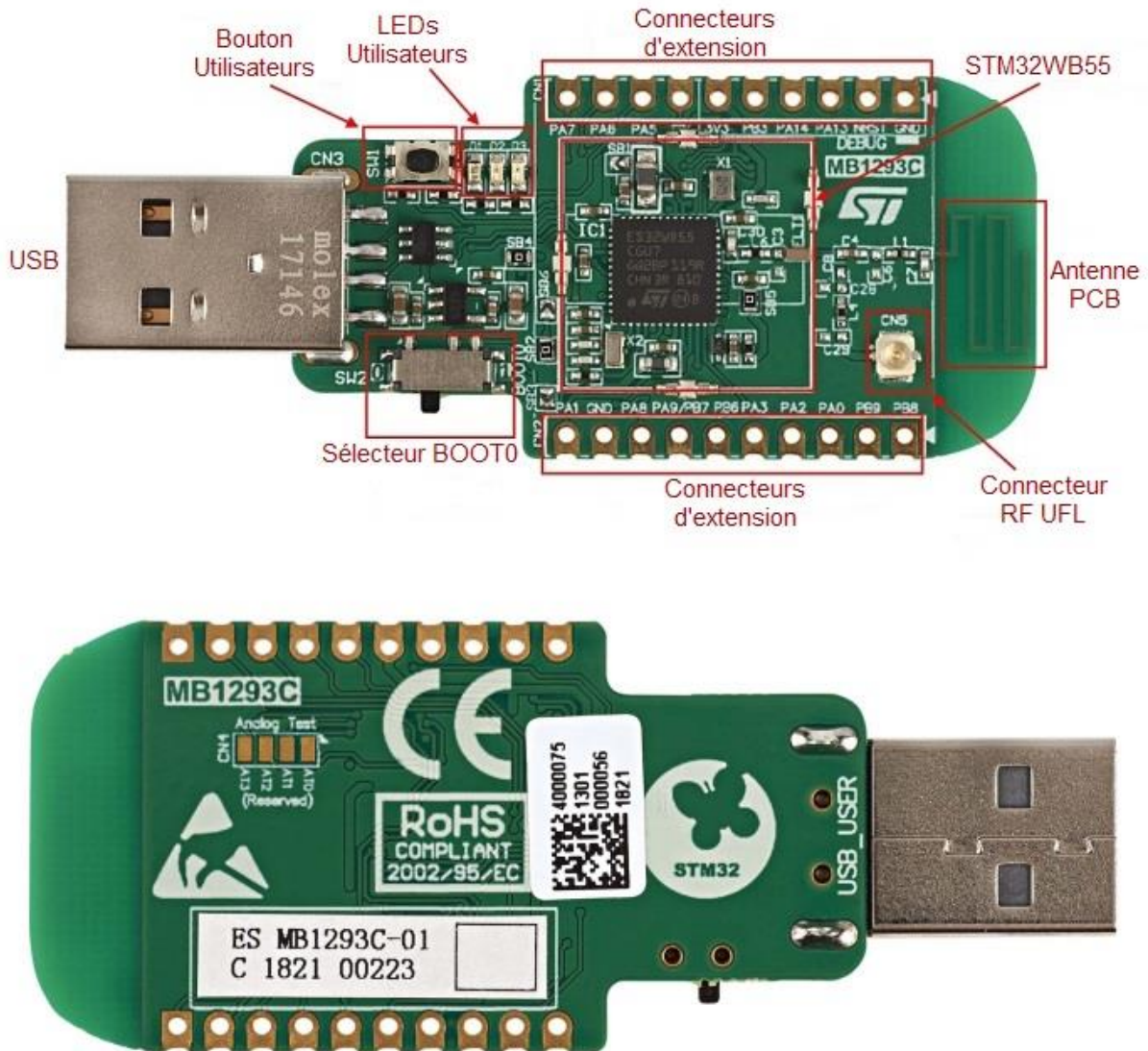


Figure 2: Dongle USB du pack Nucléo STM32WB55

Pour programmer le matériel, il a été nécessaire de télécharger certains logiciels :

- **STM32CubeMX** : c'est un outil graphique permettant de configurer les périphériques dont on a besoin, de configurer l'horloge, d'inclure des bibliothèques et de générer le code correspondant à toutes les configurations faites.
- **Atollic TrueSTUDIO** : c'est un environnement de développement intégré (IDE) qui permet de modifier et adapter le code généré par STM32CubeMX au besoin
- **STM32CubeIDE** : c'est un environnement de développement un intègre également STM32CubeMX. C'est un logiciel qui regroupe les fonctionnalités de STM32CubeMX et de Atollic TrueSTUDIO.

- **STM32CubeProgrammer** : c'est un outil qui fournit un environnement facile à utiliser et efficace pour lire, écrire et vérifier la mémoire de l'appareil via l'interface de débogage (JTAG et SWD) et l'interface du chargeur de démarrage (UART, USB, DFU, I2C, SPI et CAN). Il est utilisé pour téléverser les codes sur le microcontrôleur.
- **STM32CubeMonitor-RF** : Outils logiciels pour tester les performances du Bluetooth Low Energy (BLE) et 802.15.4 RF des appareils basés sur STM32WB.

### 2.1.2. Communication entre les deux processeurs (CPU) du microcontrôleur (MCU)

Les deux processeurs fonctionnent comme sur la figure 3.

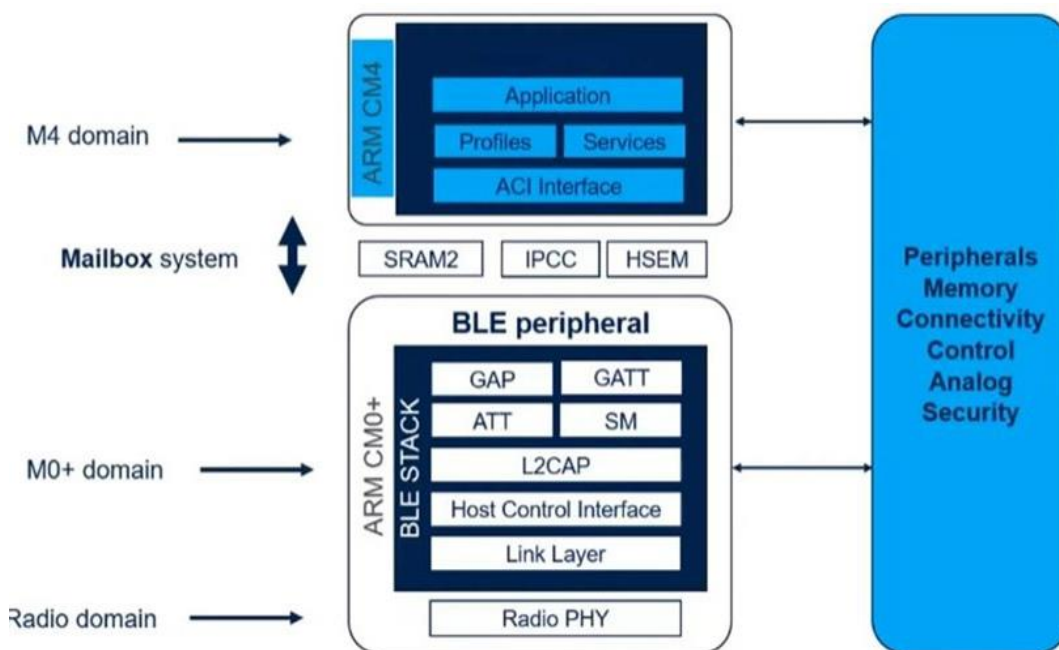


Figure 3: Fonctionnement des deux processeurs

Les éléments importants au bon fonctionnement des deux processeurs sont :

**HSEM** : c'est un sémaphore Hardware qui permet de sécuriser les périphériques et d'éviter que les deux CPU n'accèdent pas aux mêmes périphériques au même temps

**IPCC** : l'Inter-Processor Communication Controller permet d'échanger des interruptions entre les deux CPU

**SRAM2** : c'est une zone de RAM partagée entre les deux CPU

**Mailbox** : c'est une méthode d'échange des données entre les deux CPU il consiste à mettre la commande à envoyer dans SRAM2 et envoyer une interruption au deuxième CPU par l'intermédiaire de l'IPCC, une fois le CPU2 reçoit l'interruption il récupère les données qui se trouve dans la SRAM2

### 2.1.3. Différence entre le Bluetooth classique et BLE

Il existe 3 normes de Bluetooth (Figure 4) :

- Bluetooth « classique » : compatible uniquement avec le Bluetooth traditionnel
- Bluetooth Smart Ready : permet de supporter à la fois le Bluetooth «classique» et le Bluetooth Low Energy (BLE)
- Bluetooth Smart : ne supporte que le Bluetooth Low Energy (BLE).

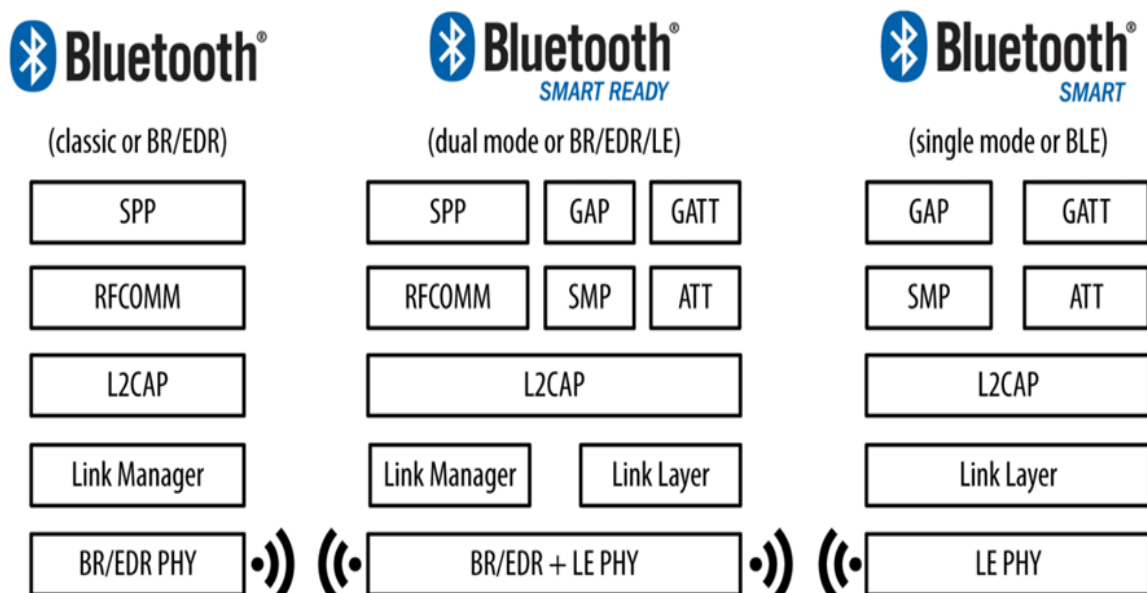


Figure 4: Différents types de Bluetooth

Le BLE est complètement différent de Bluetooth Classique au niveau de la connexion (GAP) et de l'échange des données (GATT) et aussi au niveau physique (LE PHY).

Ce qui permet au BLE d'avoir un débit du même ordre de grandeur (1 Mbit/s) pour une consommation d'énergie 10 fois moindre. Cela permet d'intégrer cette technologie dans de nouveaux types d'équipements tels que montres, appareils de surveillance médicale ou capteurs pour sportifs. La technologie permet aux appareils de se connecter dans un rayon d'environ 10 mètres.

Les modes BLE (bande passante plus limitée et très faible consommation) et Bluetooth Standard (niveau d'émission plus élevé et portée plus grande) sont donc des technologies complémentaires.

#### 2.1.4. Couches du BLE

Les différentes couches du BLE sont présentées sur la figure 5.

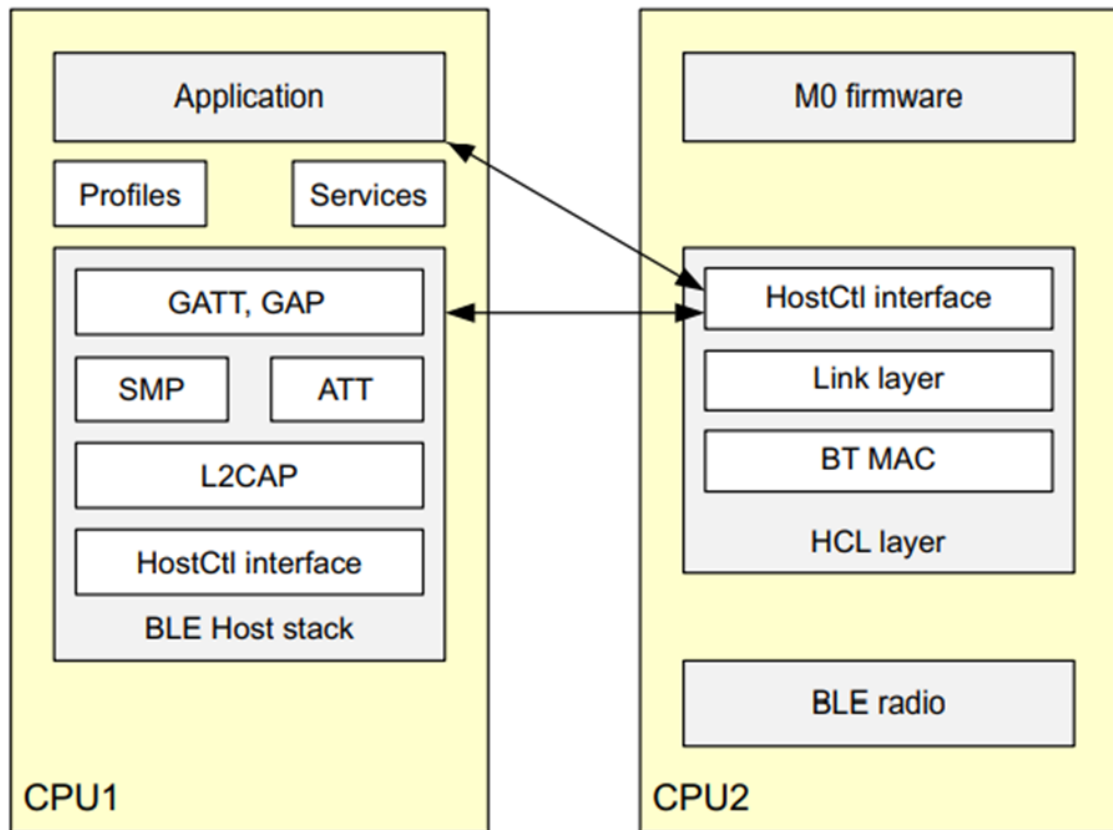


Figure 5: Couches du BLE

Les couches essentielles pour la communication BLE sont :

- GAP (Generic Access Profile) : elle est responsable de l'établissement du lien et du contrôle de connexion entre deux appareils. Le GAP est ce qui rend votre appareil visible au monde extérieur et détermine la manière dont deux appareils peuvent (ou ne peuvent pas) interagir. Le GAP définit différents rôles pour les appareils dont : les appareils centraux (central device) et les appareils périphériques (peripheral device).
- GATT (Generic Attribute Profile) : il s'agit d'un ensemble de règles décrivant comment structurer, présenter et transférer des données. La structure définit les éléments de base, tels que les services et les caractéristiques, utilisés dans un profil.
- ATT (Attribute Protocol) : L'ATT repose sur une relation client/serveur. Le serveur dispose des informations (telles que les valeurs des capteurs). Le client

est celui qui veut accéder à ces informations. ATT permet à un serveur d'exposer à un client un ensemble d'attributs. Un attribut est une valeur associée aux trois propriétés suivantes : un type d'attribut défini par un UUID, un descripteur d'attribut et un ensemble d'autorisations.

- HCI : est la couche qui permet l'interaction entre les deux CPU (CPU1 Host, CPU2 Controller)

### 2.1.5. Communication BLE

Pour établir une communication BLE il faut avoir un appareil qui diffuse (Broadcast/advertise) des trames BLE (Peripheral) et un appareil qui scanne pour se connecter (Central) comme sur la figure 6.

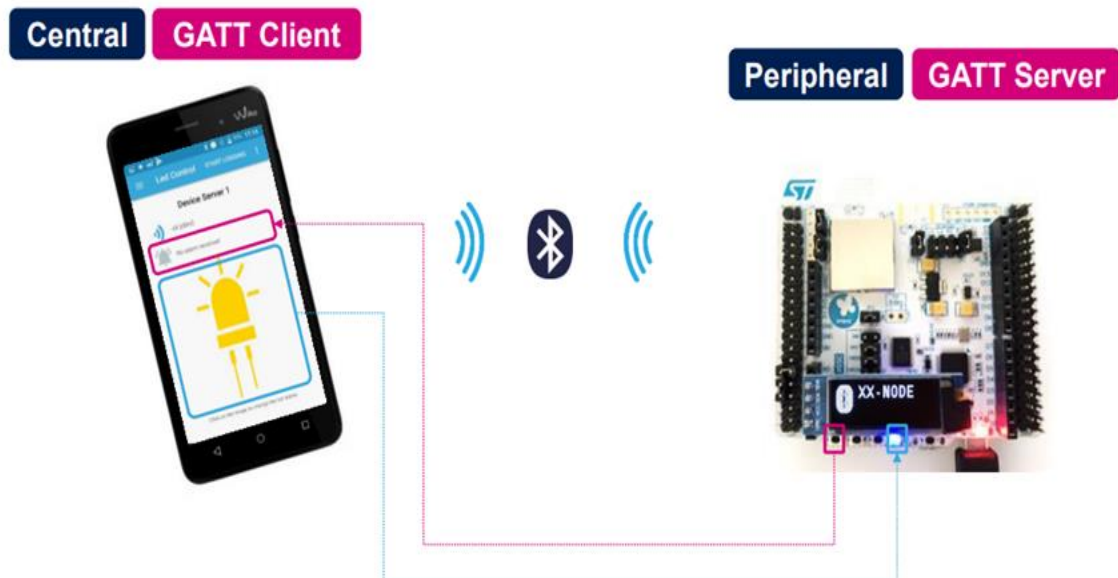


Figure 6: Communication BLE

Les étapes pour établir une communication Bluetooth (figure) sont :

- Le serveur (GAP) envoie des trames Bluetooth
- Le client (GAP) envoie une requête de connexion
- Le serveur accepte la connexion et elle s'établit
- Le serveur(GATT) propose ses services et caractéristiques
- Le client (GATT) se connecte aux services et caractéristiques
- La communication se fait via les caractéristiques

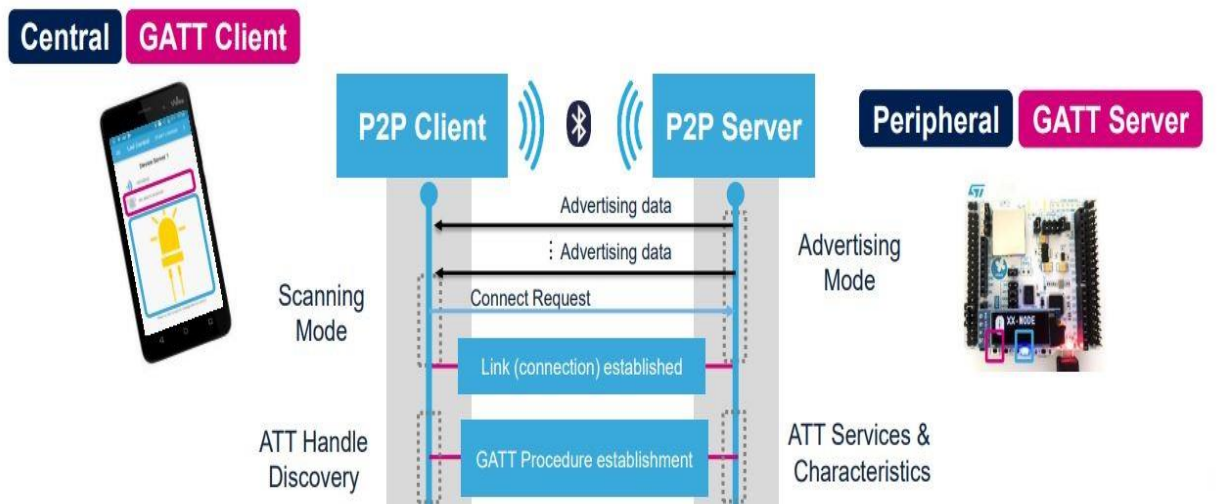


Figure 7: Etapes de la communication BLE

### 2.1.6. Les services et caractéristiques

Un service est défini comme un ensemble de données et de comportements associés permettant d'accomplir une fonction ou une fonctionnalité particulière. Un service peut contenir des caractéristiques obligatoires et des caractéristiques facultatives.

Une caractéristique est définie comme une valeur utilisée dans un service, il possède des propriétés et des informations de configuration sur le mode d'accès à la valeur et sur le mode d'affichage ou de représentation de la valeur. C'est une valeur de données transférée entre le client et le serveur.



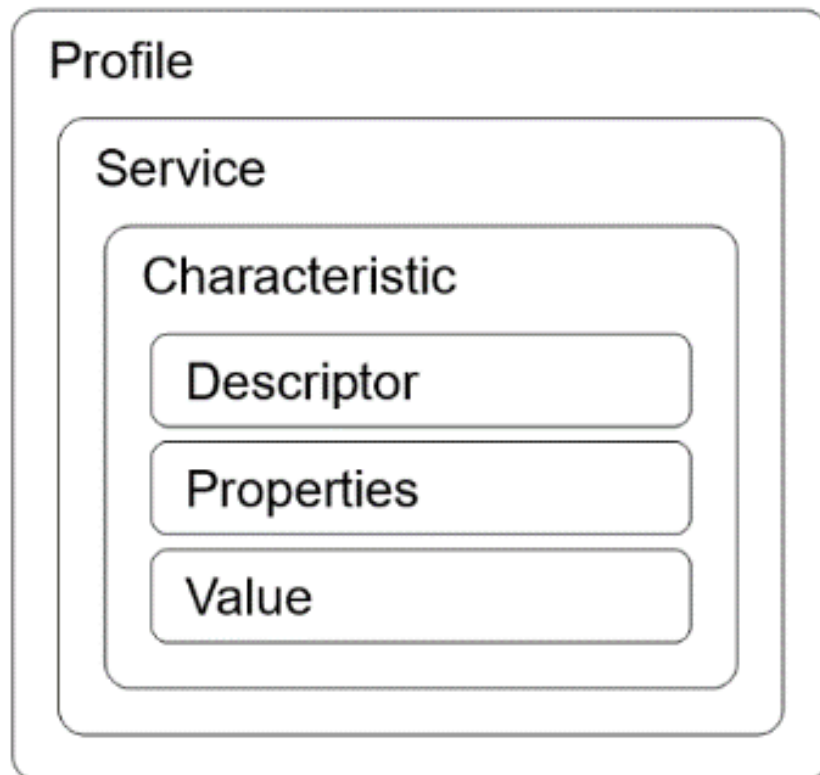


Figure 8: Hiérarchie service et caractéristique

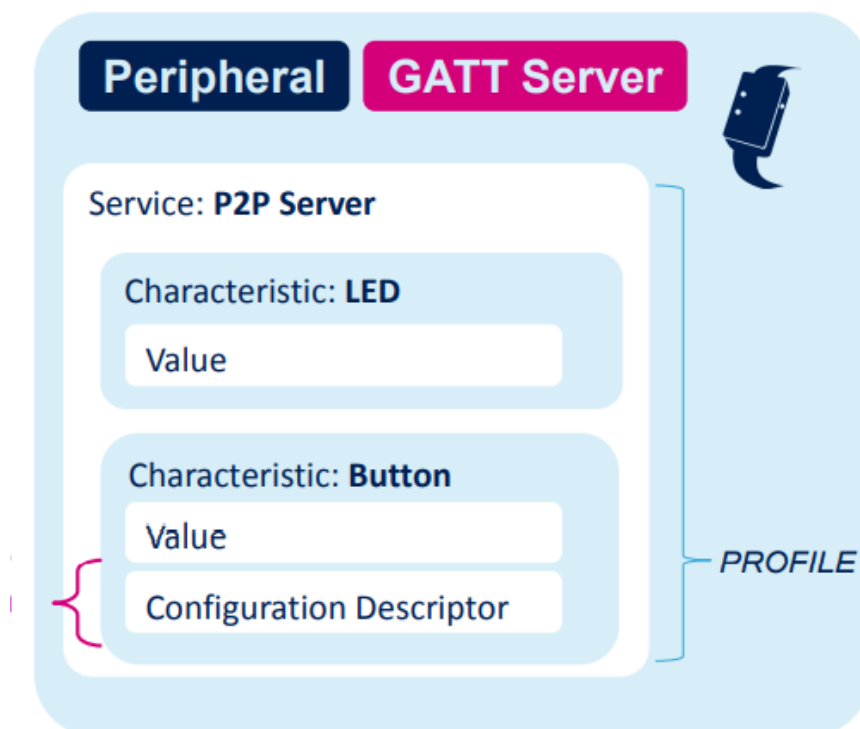


Figure 9: Exemple de service et de caractéristiques

### 2.1.7. Trame Beacon

Il existe 4 types de protocole Beacon :

- iBeacon : annoncé par Apple en 2013, il est le premier protocole beacon du marché
- Eddystone : annoncé par Google en 2015, il était appelé anciennement UriBeacon.
- AltBeacon : annoncé par Radius Networks en 2014
- GeoBeacon : annoncé par Tecno-World annoncé en 2017

Pour atteindre notre objectif, nous avons choisi d'utiliser le protocole iBeacon car il est largement pris en charge et il est simple et facile à mettre en œuvre.

La trame Beacon a une longueur pouvant aller jusqu'à 47 octets (figure 10) :

- 1 octet de préambule
- 4 octets pour l'adresse d'accès
- 2 à 39 octets de données utiles encore appelée PDU (Protocol Data Unit) ;
- 3 octets pour le CRC.

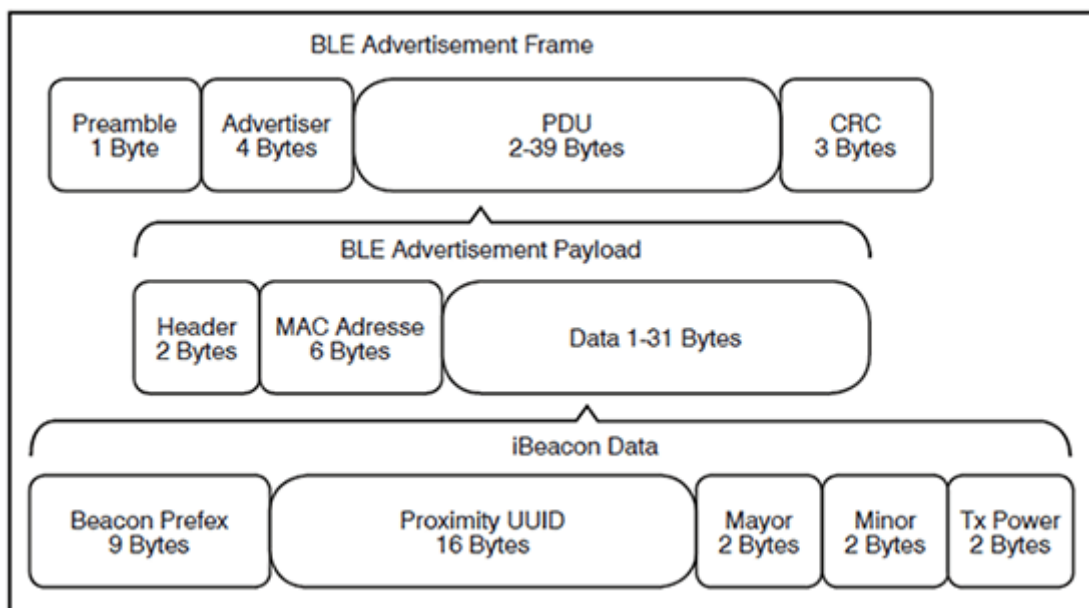


Figure 10: trame Beacon

Le PDU est composé de :

- 2 octets pour l'en-tête ;
- 6 octets pour l'adresse MAC ;
- 1 à 31 octets pour les données.

Les données sont constituées de :

- 9 octets pour le préfixe iBeacon ;



- 16 octets pour le UUID qui correspond à un identifiant unique ;
- 2 octets pour le Major qui est un identifiant qui correspond à un sous-ensemble dont la valeur varie entre 0 à 65535 ;
- 2 octets pour le Minor qui est un identifiant qui correspond à un endroit spécifique d'un sous-ensemble dont la valeur varie entre 0 et 65535 ;
- 1 octet pour TX qui est utilisé pour calibrer la force du signal.

Dans la trame iBeacon, les éléments les plus importants pour le projet sont :

- Le UUID : dans notre cas, les balises auront toutes le même UUID car elles appartiennent à la même « série ». Par exemple : *00010203-0405-0607-0809-0a0b0c0d0e0f* ;
- Le Major qui pourra représenter l'identifiant d'une course ;
- Le Minor qui pourra représenter l'identifiant d'un point de contrôle ;
- Le TX qui définira la portée du iBeacon.

Le Major, le Minor et TX doivent pouvoir être modifiable par les organisateurs de la course.

### 2.1.8. Bibliothèque développée

Pour répondre à notre objectif, nous avons créé 1 service d'UUID ***00000000-cc7a-482a-984a-7f2ed5b3e58f*** (pris au hasard) et 4 caractéristiques (figure 11) :

- Power : dont le UUID est ***00000001-8e22-4541-9d4c-21edae82ed19*** pour choisir la portée. Elle peut prendre 3 valeurs correspondant à la zone de la portée, 0 pour la zone proche, 1 pour la zone normale et 2 pour la zone éloignée;
- Major : dont le UUID ***00000002-8e22-4541-9d4c-21edae82ed19*** pour changer le major. Elle peut prendre une valeur entre 0 et 65535 ;
- Minor : dont le UUID ***00000003-8e22-4541-9d4c-21edae82ed19*** pour changer le minor. Elle peut prendre une valeur entre 0 et 65535 ;
- Validate : dont le UUID ***00000004-8e22-4541-9d4c-21edae82ed19*** pour appliquer les valeurs choisies dans les caractéristiques Minor, Major et Power. Elle peut prendre une valeur entre 0 et 9.

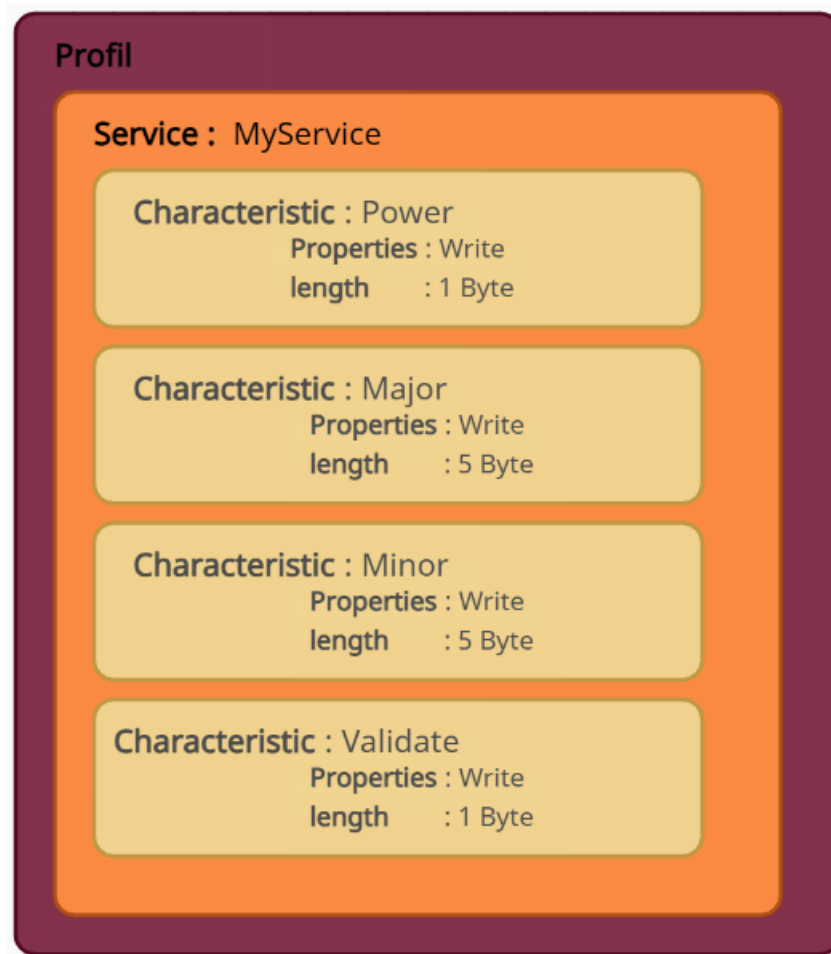


Figure 11: Service et caractéristiques de notre projet

Pour configurer la balise en iBeacon et adapter la trame aux valeurs choisies, trois fonctions sont nécessaires (figure 12) :

- **IBeacon\_Process()** : est une fonction qui permet de basculer vers iBeacon ;
- **aci\_hal\_set\_tx\_power\_level (1, Power[indexPower])** : cette fonction permet de changer la puissance d'émission. Elle prend en entrée la valeur de la puissance. Le tableau 3 présente les différentes valeurs ;
- **aci\_gap\_update\_adv\_data (sizeof(service\_data), service\_data)** : cette fonction permet de changer la trame iBeacon.

Dès qu'on écrit un caractère quelconque dans La caractéristique Validate, la balise arrête de communiquer en BLE donc elle n'est plus configurable et commence à envoyer des trames iBeacon paramétrées selon les valeurs de Minor et Major choisies.

Pour pouvoir configurer à nouveau la balise, il faut appuyer sur le reset ou enlever l'alimentation et la remettre.

```

IBeacon_Process();
uint8_t service_data[] =
{
    26, //
    AD_TYPE_MANUFACTURER_SPECIFIC_DATA, //
    0x4C, 0x00, 0x02, 0x15, //
    0, 1, 2, 3, 4, 5, 6, 7, //Location UUID
    8, 9, 10, 11, 12, 13, 14,15,|
    major[0], major[1], // Major number
    minor[0], minor[1], // Minor number
    tx_power[indexPower] //the Tx power};
};

aci_hal_set_tx_power_level(1, Power[indexPower]);
aci_gap_update_adv_data(sizeof(service_data), service_data);
memset(attribute_modified->Attr_Data,0,sizeof(attribute_modified->Attr_Data));

```

Figure 12: Code permettant de changer la trame

Valeurs pour la fonction	Puissance correspondante
0x00	-40 dBm
0x01	-20.85 dBm
0x02	-19.75 dBm
0x03	-18.85 dBm
0x04	-17.6 dBm
0x05	-16.5 dBm
0x06	-15.25 dBm
0x07	-14.1 dBm
0x08	-13.15 dBm
0x09	-12.05 dBm
0x0A	-10.9 dBm
0x0B	-9.9 dBm
0x0C	-8.85 dBm
0x0D	-7.8 dBm
0x0E	-6.9 dBm
0x0F	-5.9 dBm
0x10	-4.95 dBm
0x11	-4 dBm
0x12	-3.15 dBm
0x13	-2.45 dBm
0x14	-1.8 dBm
0x15	-1.3 dBm
0x16	-0.85 dBm
0x17	-0.5 dBm
0x18	-0.15 dBm
0x19	0 dBm

0x1A	+1 dBm
0x1B	+2 dBm
0x1C	+3 dBm
0x1D	+4 dBm
0x1E	+5 dBm
0x1F	+6 dBm

Tableau 3: Valeurs pour la fonction *aci\_hal\_set\_tx\_power\_level()*

### 2.1.9. Application

Pour intégrer notre dispositif au projet Vikazimut, il est important d'adapter l'application Vikazimut en fonction. Nous avons créé des applications dont le code sera intégré à l'application Vikazimut.

Deux applications Android ont été créées : une application « administrateur » pour la configuration de la balise et l'autre application a pour rôle la récupération de la trame (figure 13, 14 et 15). La validation du passage se fera donc en fonction du major et du minor récupérés. Seuls les organisateurs auront accès à la première application et les coureurs auront accès à la deuxième.

Pour le développement des applications, l'environnement de développement **Android Studio** a été utilisé. Il permet de développer des applications Android en utilisant **JAVA** pour le fonctionnement de l'application et le **XML** pour tout ce qui concerne le design.

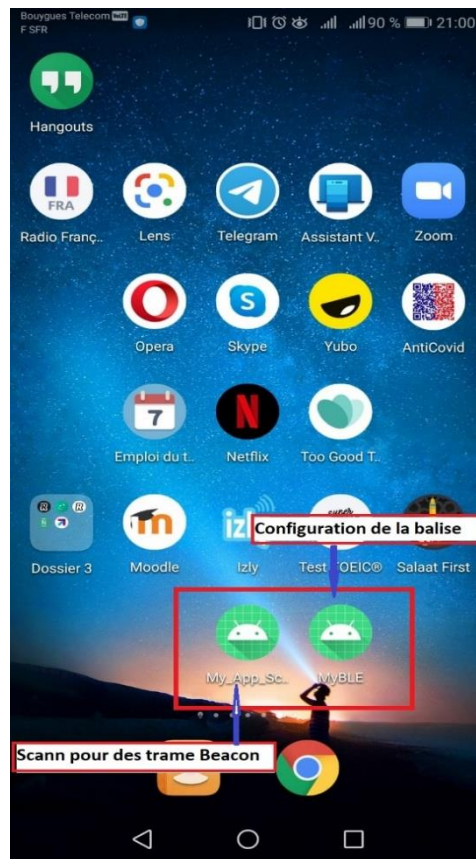


Figure 13: Applications Android développées

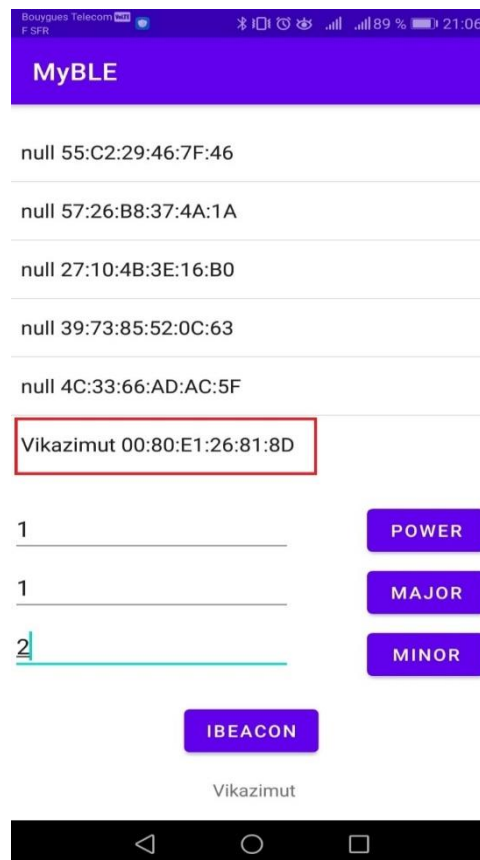


Figure 14: App pour la configuration de la balise

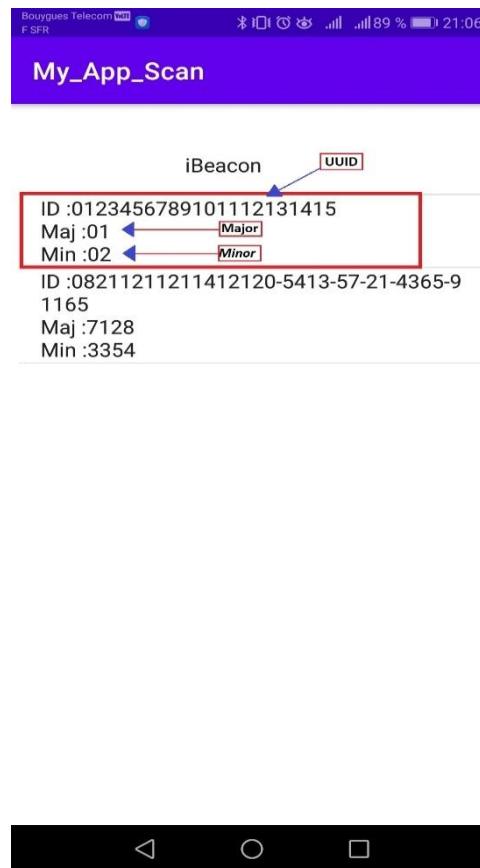


Figure 15: App pour la récupération de la trame iBeacon

### 2.1.10. Site Web

Le développement d'un site web était important également pour adapter le site web existant du projet Vikazimut.

Le site web développé permet de se connecter en Bluetooth à la balise et de la configurer (figure 17, 18, 19). Il a été développé en **html** et en **javascript** pour le fonctionnement.

Web Bluetooth demo

← → ↻ ⓘ Fichier | C:/Users/ornel/OneDrive/Documents/Beacon.html

Device connection...

Entrez la zone de la portée (0 pour zone proche, 1 pour zone normale et 2 pour zone lointaine):

Entrez le major:

Entrez le minor:

Figure 16: Interface du site web

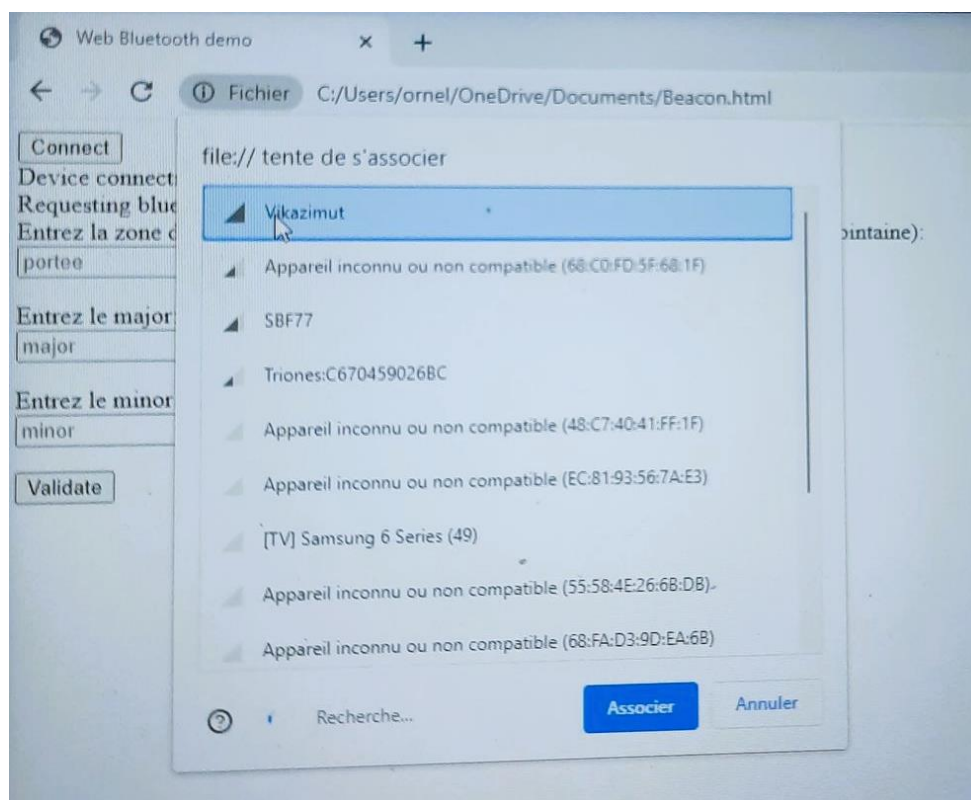


Figure 17: Connexion à la balise par Bluetooth

Web Bluetooth demo

← → ↻ ⓘ Fichier | C:/Users/ornel/OneDrive/Documents/Beacon.html

Connect

Device connection...

Requesting bluetooth device...

"Vikazimut" bluetooth device selected

Wait a minute please...

You can modify values...

Entrez la zone de la portée (0 pour zone proche, 1 pour zone normale et 2 pour zone lointaine):

1 Send

Entrez le major:

213 Send

Entrez le minor:

321 Send

Validate

---

Web Bluetooth demo

← → ↻ ⓘ Fichier | C:/Users/ornel/OneDrive/Documents/Beacon.html

Connect

Device connection...

Requesting bluetooth device...

"Vikazimut" bluetooth device selected

Wait a minute please...

You can modify values...

le major est: 213

le minor est: 321

Entrez la zone de la portée (0 pour zone proche, 1 pour zone normale et 2 pour zone lointaine):

portee Send

Entrez le major:

major Send

Entrez le minor:

minor Send

Validate

Figure 18: Configuration de la balise





Figure 19: Balise configurée comme iBeacon

## 2.2. Dimensionnement énergétique

Il est impossible de calculer exactement la portée de la balise compte tenu de l'environnement et des obstacles. Pour cela, trois zones de portée ont été définies :

- Zone proche avec une puissance de -20.85 dBm soit 8.22 uW ;
- Zone normale avec une puissance de -12.05 dBm soit 0.062 mW ;
- Zone éloignée avec une puissance de -3.15 dBm soit 0.484 mW.

L'objectif est d'avoir une autonomie d'au moins 2 ans soit 17520 heures.

L'énergie nécessaire correspond à :

$$\text{Energie(mWh)} = \text{Puissance (mW)} * \text{durée (h)}$$

- Zone proche : 144.01 mWh
- Zone normale : 1086.24 mWh
- Zone éloignée : 8479.68 mWh

En supposant qu'on utilise une pile bouton de 3V, la capacité de la pile est :

$$\text{Capacité(mAh)} = \text{Energie (mWh)}/3V$$

- Zone proche : 48 mAh

- Zone normale : 362.08 mAh
- Zone éloignée : 2826.56 mAh

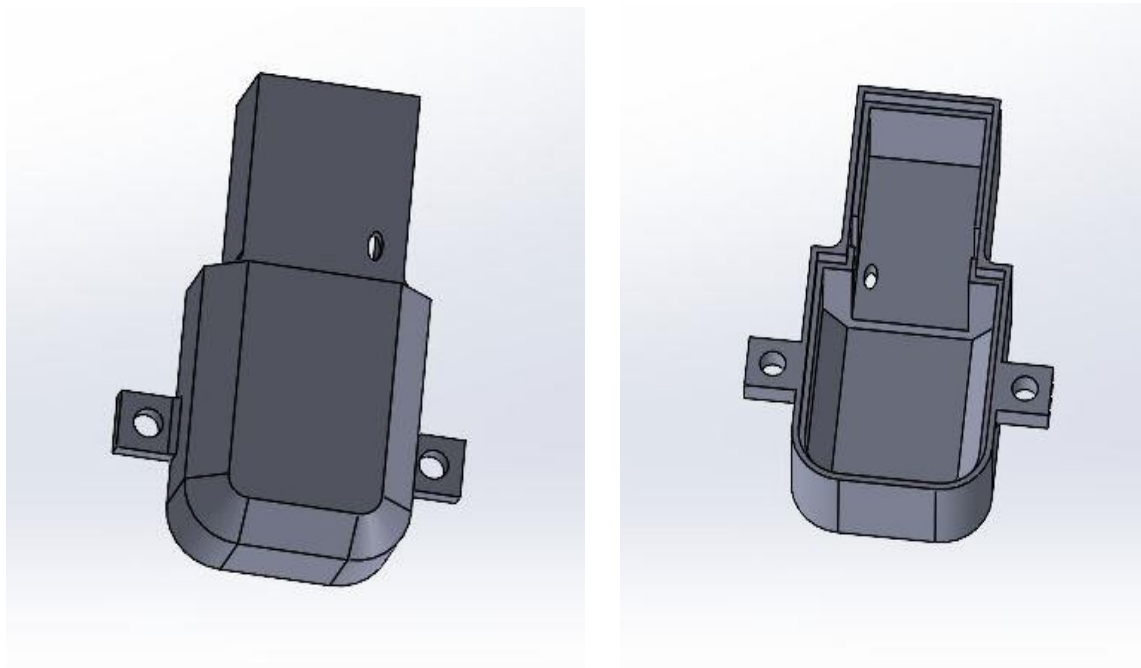
## 2.3. Conception du boîtier

Le dernier objectif est de concevoir le boîtier qui contiendra notre balise qui est le dongle USB.

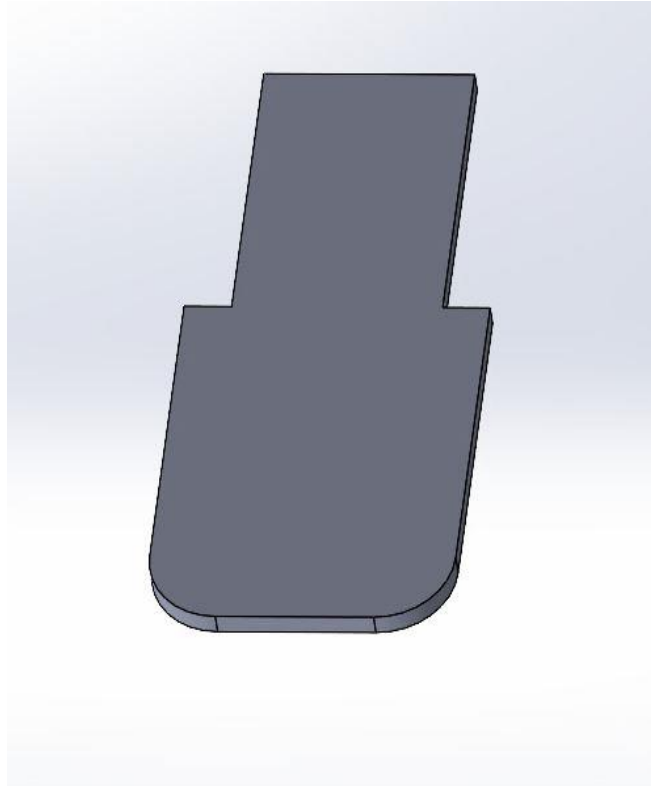
Pour la conception 3D on a utilisé le logiciel SolidWorks, on a opté pour un design simple qui respecte nos exigences c'est à dire qu'il soit étanche et permette d'intégrer le format de la balise.

Les carrés de part et d'autre permettront d'intégrer le boîtier à la balise déjà existante.

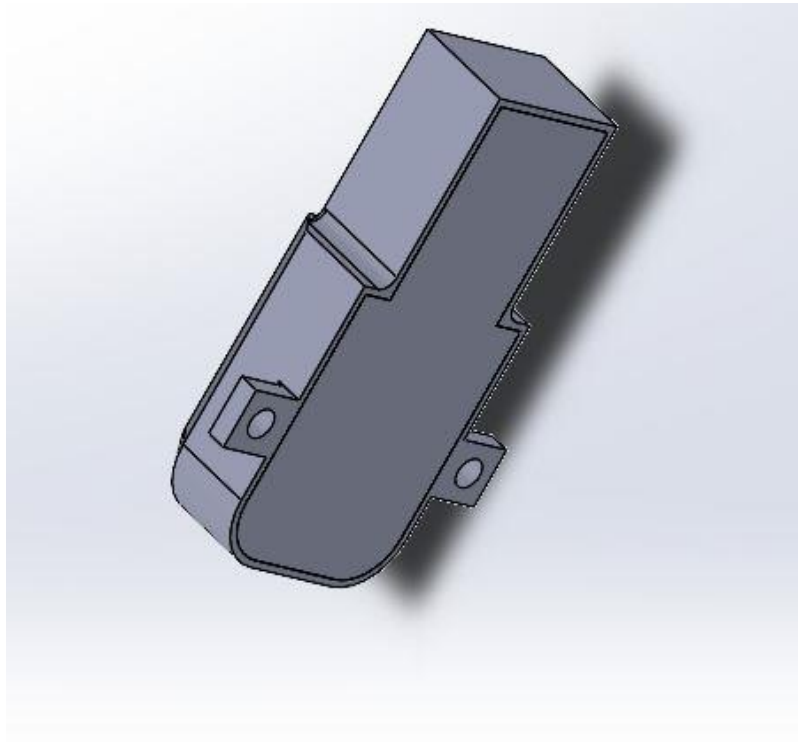
Le trou dans le boîtier est fait pour le bouton reset. Une matière en caoutchouc pourra le couvrir pour conserver l'étanchéité et permettre l'appui du bouton.



*Figure 20: La conception du boîtier de la balise*



*Figure 21: Porte de la balise*



*Figure 22: Assemblage du Boitier*

### 3- Conclusion

Ce projet n'a été que très bénéfique pour nous. Nous avons été confrontés à une technologie qui nous était inconnue et dont les documents sont difficiles à trouver. Nous avons pris du temps pour bien maîtriser la technologie. Nous avons aimé le cadre du travail, l'autonomie dans le travail, la confiance de la part de M. PIGEON, ses apports et conseils et ceux de M. DUFAY.

Ainsi, après avoir compris comment fonctionne la technologie, nous avons développé la bibliothèque répondant à nos objectifs et plus. Nous avons développé les applications et site web permettant d'utiliser notre dispositif. Après validation par le tuteur, nous avons fait le dimensionnement énergétique et la conception du boîtier. Nous avons dû modifier le design pour répondre aux attentes du tuteur et client. Nous avons effectué toutes les tâches prévues. Des iBeacons du marché ont également été commandés, mais il était impossible de les configurer soit parce qu'on n'avait pas la documentation ou parce qu'on n'avait pas le firmware adapté.

Mais le projet est loin de sa phase finale. Le client aimerait intégrer des panneaux solaires pour l'alimentation de la balise. Ce travail pourra se baser sur les calculs effectués lors du dimensionnement énergétique. La question d'optimisation se pose également. Au lieu d'utiliser le dongle USB, on pourrait utiliser le microcontrôleur STM32WB5MMG qui a une taille de l'ordre de quelques millimètres mais qui possède les mêmes caractéristiques que le pack Nucléo. Mais STMicroelectronics ne pourra nous en procurer qu'en septembre pour des questions de sécurité.