



Université
Lille1
Sciences et Technologies

Projet de fin d'étude :

Prototypage d'interactions localisées et contextualisées

Tuteurs : Mr Peter et Mr Vantroys



POLYTECH[®]
LILLE

Olivier Tailliez – IMA 5

Table des matières

Introduction.....	1
Présentation du projet	2
1. Les scénarios étudiés.....	2
2. Le cahier des charges	3
3. Le Bluetooth Low Energy.....	4
Différences avec le Bluetooth « classique »	4
Le protocole d'échange : Generic Attribute Profile (GATT)	5
Travail effectué.....	6
1. Installation de la balise.....	6
2. Les applications androids	7
L'exemple BluetoothLeGatt	8
L'application Musée	9
3. Les difficultés rencontrées	10
L'échange de données en Bluetooth	10
La réalisation des applications	10
Bilan.....	11
Annexe.....	12
Annexe 1 : Différences entre le Bluetooth et le BLE.....	12
Références.....	13

Introduction

Aujourd'hui, il existe plusieurs méthodes de communication sans fil pour les objets connectés comme le wifi, le Bluetooth ou le protocole NFC.

Ces techniques sont en constante évolution, et depuis 2001 la société Nokia recherche une solution pour rendre le Bluetooth moins consommateur en énergie. En 2010, le Bluetooth 4.0 ou Bluetooth Low Energy ou BLE, est disponible.

Les premiers à avoir utilisé cette technologie sont les iPhones. Mais Apple la bride pour ne pas être utilisable par des produits d'autres marques.

Il existe maintenant beaucoup de téléphones possédant cette technologie. On peut donc imaginer des applications pour le grand public. De plus la faible consommation du BLE permet la mise en place de balise fonctionnant sur batterie ou pile. On est donc possible de réaliser un réseau de détection des personnes et des interactions en fonction de la localisation de la personne.

Par exemple, en installant une balise sur ces clés, il est possible de détecter leur position grâce à un téléphone.



Avec des balises, impossible de perdre ces affaires

Présentation du projet

1. Les scénarios étudiés

Le premier scénario à atteindre est celui d'un supermarché :

Un client arrive à l'entrée et active son application. Il s'identifie comme étant un client. Il envoie sa liste des courses à l'aide du BLE. Cette liste peut s'afficher sur un écran. En fonction de cette liste, il reçoit un coupon de réduction.

Il s'agit d'un cas d'interaction avec une Balise BLE, où le principal objectif est l'échange de données.

Le deuxième scénario est :

Dans un musée, un visiteur veut avoir plus d'information sur une œuvre. Grâce à la balise implantée à côté, l'utilisateur a directement sur son smartphone les informations complémentaires sur l'œuvre en face de lui.

Cette fois-ci, le BLE sert à localiser l'utilisateur. L'échange de données n'est pas nécessaire si les données sont dans l'application ou accessibles par le wifi.

2. Le cahier des charges

L'objectif de ce projet est tout d'abord de concevoir une balise BLE à l'aide d'une raspberry Pi, afin qu'elle soit détectée par un téléphone. Puis d'établir une communication bidirectionnelle entre la balise et un smartphone.

Il faut donc réaliser un programme sur la raspberry ainsi qu'une application android afin de communiquer entre les deux appareils.

Par la suite, il s'agit de mettre en scène les scénarios proposés.

Afin de réaliser ces scénarios je dispose du matériel suivant :

- Une ordinateur raspberry Pi
- Un module Bluetooth compatible BLE
- Un téléphone Samsung galaxy S3
- Un téléphone Huawei Ascend P7



La raspberry Pi et le module Bluetooth

3. Le Bluetooth Low Energy

Différences avec le Bluetooth « classique »

Le BLE est un protocole d'échange de données entre deux appareils basés sur le protocole Bluetooth « classique ». On parle aussi de Bluetooth 4.0.

Le principal avantage de cette nouvelle technologie est la consommation en énergie allant de 0.1 à 0.5W contre 1W pour le Bluetooth.

Par contre, les anciens modules Bluetooth ne sont pas compatibles BLE. De plus, le débit de données en BLE est plus faible qu'en Bluetooth. Les applications sont donc différentes. Par exemple, le débit ne permet pas encore l'écoute de musique en streaming. Heureusement les modules BLE développés aujourd'hui sont généralement rétro compatible Bluetooth.

Il existe aussi des différences au niveau de la bande passante du BLE ainsi que sur la technique de modulation du signal¹.

¹ Voir Annexe 1

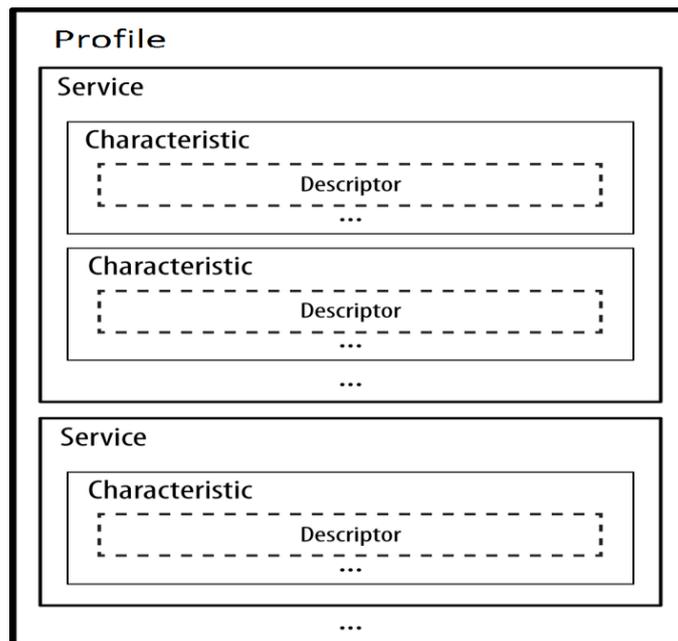
Le protocole d'échange : Generic Attribute Profile (GATT)

Avec le BLE, les appareils sont séparés en deux catégories, les clients et les serveurs. Le client initialise les requêtes GATT et reçoit des réponses des serveurs. Il n'y a qu'un client pour N serveurs. Il s'agit donc d'une relation maître-esclaves

Une requête GATT est un profil, il s'agit d'un ensemble de services qui est un ensemble de caractéristiques.

Chaque caractéristique contient une valeur. Les caractéristiques au sein d'un service sont généralement liées, par exemple, la mesure d'un capteur de température et l'intervalle entre deux mesures.

Une caractéristique peut aussi être associée avec un descripteur, donnant des informations complémentaires sur cette première.



Les caractéristiques, descripteurs et services sont désignés comme les attributs du profil.

Le profil est identifié par un UUID (Universally Unique Identifier). Un UUID est composé de 128 bits.

Travail effectué

Les objectifs étant définis et le matériel disponible, j'ai pu commencer le projet rapidement, mais pas directement par la programmation.

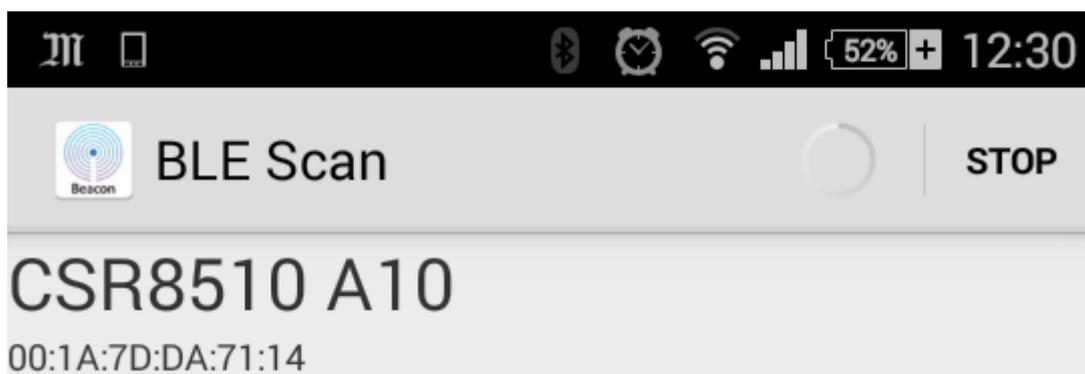
1. Installation de la balise

La première étape du projet est de transformer la raspberry Pi en balise BLE. Mr Peter m'ayant fourni en plus du matériel un tutoriel pour réaliser l'installation du module Bluetooth².

Installation c'étant bien dérouler, on peut afficher l'état du Bluetooth de la raspberry Pi ainsi que sa détection par le téléphone :

```
pi@raspberrypi ~ $ sudo hciconfig
hci0:  Type: BR/EDR  Bus: USB
      BD Address: 00:1A:7D:DA:71:14  ACL MTU: 310:10  SCO MTU: 64:8
      UP RUNNING
      RX bytes:1118 acl:0 sco:0 events:58 errors:0
      TX bytes:829 acl:0 sco:0 commands:58 errors:0
```

La balise est en fonctionnement (UP RUNNING) et son adresse est 00 :1A :7D :DA :71 :14



La balise est détecté par le téléphone

La raspberry Pi peut maintenant être utilisé pour des applications de localisation, comme le musée par exemple.

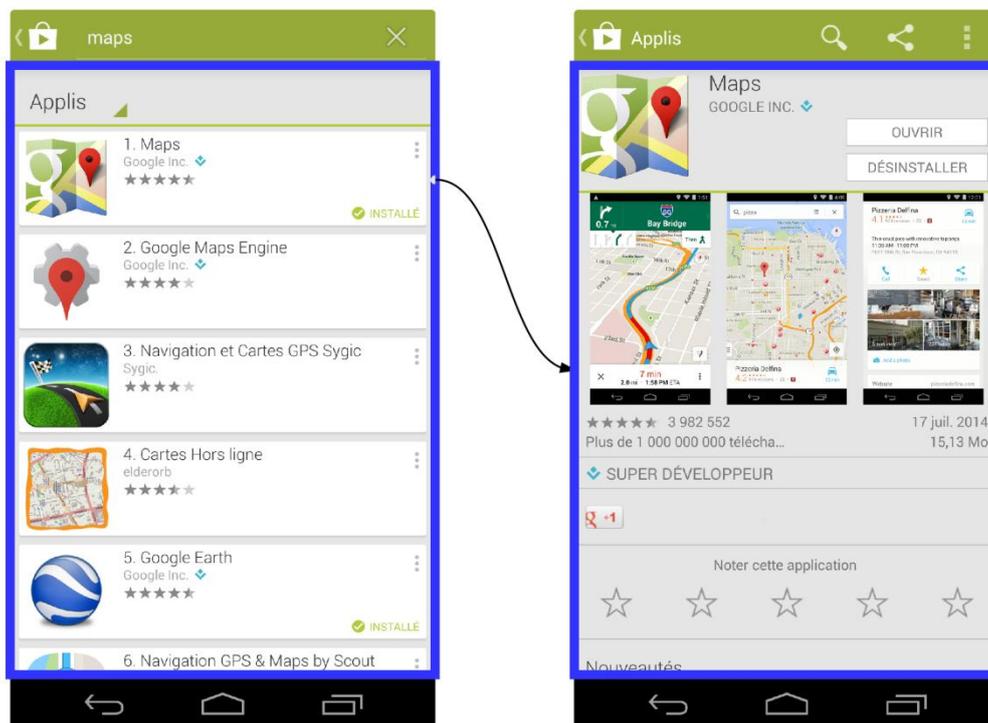
² "piBeacon -DIY Beacon with a Raspberry Pi" de Kevin Townsend

2. Les applications androids

Sachant dès le départ que je possédais des lacunes en langage JAVA et sur le fonctionnement des applications androids, j'ai tout d'abord suivi durant 2-3 semaines des tutoriels afin d'acquérir des bases pour la suite du projet.

Les applications androids sont écrites en JAVA et en XML. Le JAVA est le langage principal des applications, définissant tous les objets utilisés. Le XML sert principalement à la réalisation de l'interface graphique de l'application.

Une application est généralement une suite d'activités. Il s'agit par exemple de la fenêtre affichée lors du lancement de n'importe quelle application. Une activité est composée d'une interface graphique, des fonctions et un contexte. L'interface graphique sert à l'utilisation des fonctions. Le contexte définit l'état de l'activité, par exemple en cours d'utilisation, en pause ou arrêtée.



En cliquant sur un élément de la liste à gauche, on ouvre une nouvelle activité (Source : Openclassroom.com)

Afin de réaliser mes applications, j'ai utilisé le logiciel Android Studio. Sur ce dernier, il est possible de télécharger des exemples d'applications diverses.

C'est ainsi que je me suis appuyé sur l'application BluetoothLeGatt pour réaliser l'application du musée.

L'exemple BluetoothLeGatt

Cette application permet de détecter les appareils ayant le BLE activé, de se connecter à eux et de lire des données reçues.

Elle est divisée en 4 activités :

- **DeviceScanActivity :**
Il s'agit de l'activité de démarrage. Elle initialise le Bluetooth et lance la recherche des appareils disponibles. Elle affiche ensuite les appareils détectés et leurs adresses MAC.
En cliquant, par exemple, sur la raspberry Pi détectée on lance l'activité DeviceControlActivity
- **DeviceControlActivity :**
Cette activité montre de nouveau le nom et l'adresse MAC des appareils mais aussi si des données sont reçues. Elle permet de se connecter à un appareil.
- **BluetoothLeService :**
Cette dernière fonctionne de pair avec DeviceControlActivity, qui gère surtout l'affiche à partir des informations transmises par l'activité BluetoothLeService, qui gère les interactions avec les autres appareils.
- **SampleGattAttributes :**
Elle propose un exemple d'attributs à envoyer ou recevoir.

L'application Musée

Afin de réaliser cette application, je n'ai gardé que l'activité DeviceScanActivity. En effet j'ai besoin de détecter les appareils, pas de transférer de données.

Lorsque l'application s'allume, DeviceScanActivity cherche les appareils et affiche leurs noms.

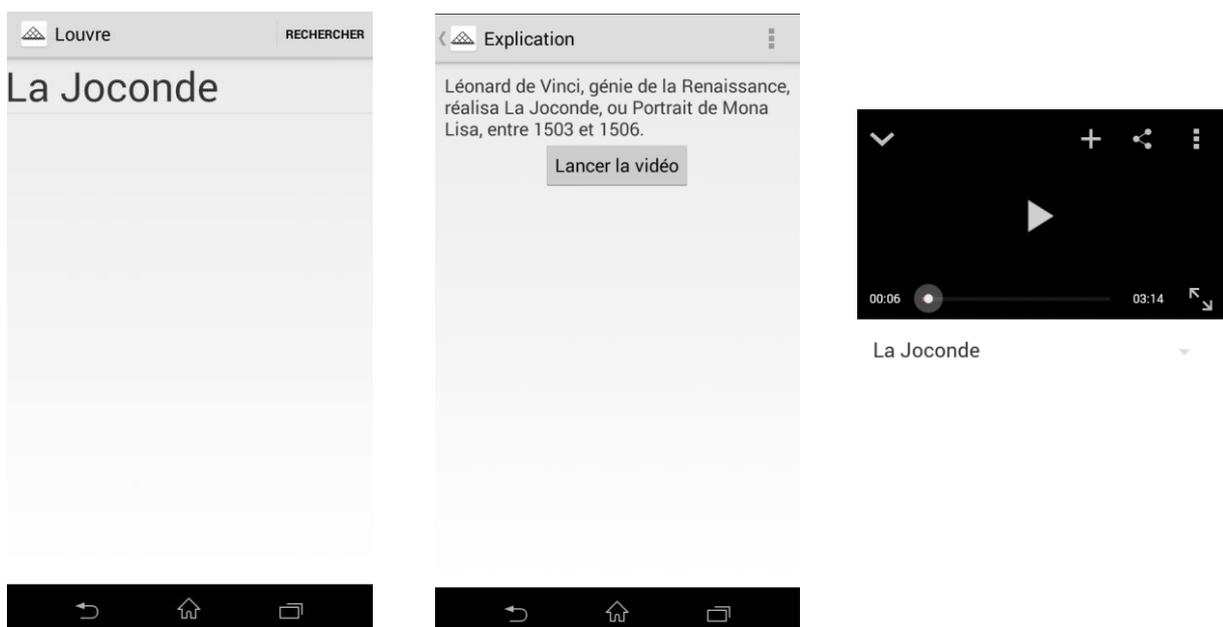
En appuyant sur le nom on affiche la seconde activité possédant une explication de l'œuvre sélectionnée et un bouton afin de lancer une vidéo explicative.

On ne peut accéder à cette seconde activité que si l'adresse MAC de la balise est enregistrée dans l'application dans un tableau de chaînes de caractères initialisé lors de la programmation. Sinon l'application affiche « adresse inconnue ».

Lors du passage à la seconde activité, cette dernière reçoit le nom, l'adresse MAC ainsi que sa place dans le tableau de valeur.

Dans la seconde partie de l'application, il existe un autre tableau répertoriant les URL des vidéos et les textes dans le même ordre que les adresses MAC. Il est ainsi facile de lancer la vidéo correspondante.

La vidéo est lancée par l'application Youtube.



Les deux activités de l'application et le lancement de la vidéo

3. Les difficultés rencontrées

L'échange de données en Bluetooth

Durant ce projet, je n'ai réussi à réaliser qu'une seule des deux applications demandées. En effet j'ai eu des difficultés à réaliser un échange de données entre les deux appareils.

D'une part, la raspberry Pi n'a jamais détecté les différents smartphones connectés.

J'ai donc essayé de réinstaller les pilotes Bluetooth de la raspberry, prendre un téléphone plus récent, jailbreaker mon téléphone personnel afin d'installer les roms de mon choix et modifier les fichiers systèmes. Mais aucune de mes tentatives n'a fonctionné.

Parfois lorsque je lance le scan Bluetooth de la raspberry, je vois une adresse s'affichée. Mais habitant à coté d'un bar, je pense qu'il s'agit d'un des clients.

D'autre part, j'ai essayé d'implanter une fonction d'écriture à l'application BluetoothLeGatt. Mais je n'ai pas réussi à finir cette partie avant la fin du projet.

Je n'aurais pas dû me concentrer sur la raspberry Pi durant autant de temps sachant que la durée de mon projet est réduite.

La réalisation des applications

J'ai eu aussi des difficultés à comprendre l'exemple BluetoothLeGatt.

De plus, lors de la réalisation d'une application il arrive parfois que l'on bloque à cause d'un oubli ou d'une inattention. Il s'agit de mon premier code en JAVA et j'ai réalisé certaines erreurs typiques du langage comme ne pas employer l'opérateur « == » pour comparer deux chaînes de caractères.

Bilan

Une des deux applications du cahier des charges initial a été réalisé. Mais cette application est encore perfectible. Par exemple, pour l'instant les adresses MAC des balises ainsi que les explications sont stockées dans l'application. On peut imaginer la mise en place de fichiers externes ou de bases de données. J'ai essayé la deuxième solution mais sans succès.

De plus, je n'ai pu réalisé de test qu'avec une seul balise. Même si j'ai confiance dans la robustesse de mon programme, je ne peux pas être certain qu'il n'y aura pas des bugs avec plusieurs balises.

Personnellement, j'ai beaucoup apprécié ce projet car il m'a permis d'étudier les applications androids ainsi que les langages JAVA et XML. Il s'agit de 3 sujets où il est important d'avoir des notions en tant qu'élève-ingénieur.

Ensuite, j'ai pu expérimenter le travail en autonomie. Il faut savoir tester, « bidouiller » pour apprendre ou aller chercher les réponses aux bons endroits. Par contre, il peut parfois être frustrant de bloquer plusieurs jours sur un même problème.

Je compte regarder les autres exemples d'application proposer par Android Studio et continuer à tester des nouvelles applications.

Annexe

Technical Specification	Classic Bluetooth technology	Bluetooth Smart technology
Distance/Range (theoretical max.)	100 m (330 ft)	>100 m (>330 ft)
Over the air data rate	1–3 Mbit/s	1 Mbit/s
Application throughput	0.7–2.1 Mbit/s	0.27 Mbit/s
Active slaves	7	Not defined; implementation dependent
Security	56/128-bit and application layer user defined	128-bit AES with Counter Mode CBC-MAC and application layer user defined
Robustness	Adaptive fast frequency hopping, FEC, fast ACK	Adaptive frequency hopping, Lazy Acknowledgement, 24-bit CRC, 32-bit Message Integrity Check
Latency (from a non-connected state)	Typically 100 ms	6 ms
Minimum total time to send data (det.battery life)	100 ms	3 ms
Voice capable	Yes	No
Network topology	Scatternet	Scatternet
Power consumption	1 W as the reference	0.01 to 0.5 W (depending on use case)
Peak current consumption	<30 mA	<15 mA
Service discovery	Yes	Yes
Profile concept	Yes	Yes
Primary use cases	Mobile phones, gaming, headsets, stereo audio streaming, smart homes, wearables, automotive, PCs, security, proximity, healthcare, sports & fitness, etc.	Mobile phones, gaming, PCs, watches, sports and fitness, healthcare, security & proximity, automotive, home electronics, automation, Industrial, etc.

Annexe 1 : Différences entre le Bluetooth et le BLE

Références

Pour apprendre à développer des applications :

<http://openclassrooms.com/courses/creez-des-applications-pour-android>

<http://openclassrooms.com/courses/apprenez-a-programmer-en-java>

<http://developer.android.com/training/index.html>

<http://developer.android.com/index.html>

Pour réaliser la balise :

<https://learn.adafruit.com/downloads/pdf/pibeacon-ibeacon-with-a-raspberry-pi.pdf>

Pour en apprendre plus sur le BLE :

http://en.wikipedia.org/wiki/Bluetooth_low_energy