|  |
| --- |
| hepia |
| SmartBag |
| Système de gestion d’affaires |

|  |
| --- |
| Adrien Taboada  06/03/2017 |

Table des matières

[Projet CHIC 2](#_Toc477255125)

[La team 2](#_Toc477255126)

[Généralités 2](#_Toc477255127)

[Composants 2](#_Toc477255128)

[Architecture 2](#_Toc477255129)

[Microcontrolleur 2](#_Toc477255130)

[Comparatif 2](#_Toc477255131)

[nRF52 2](#_Toc477255132)

[Environnement de développement 2](#_Toc477255133)

[Keil uVision 5 2](#_Toc477255134)

[nRFGoStudio 2](#_Toc477255135)

[Kinetis Protocol Analyzer 2](#_Toc477255136)

[Bluetooth Low Energy (BLE) 3](#_Toc477255137)

[Generic Attribute Profile (GATT) 3](#_Toc477255138)

[Profil BlE 3](#_Toc477255139)

[Proximity (PXP) 3](#_Toc477255140)

[Notre profil 4](#_Toc477255141)

[SoftDevice S132 4](#_Toc477255142)

[RFID 5](#_Toc477255143)

[Alimentation 5](#_Toc477255144)

[Sources 6](#_Toc477255145)

[Bluetooth 6](#_Toc477255146)

Smartbag

# Projet CHIC

<explication CHIC> <projet dois être connecté><prendre sur doc collet>

## La team

<photo avec spécialisation>

# Généralités

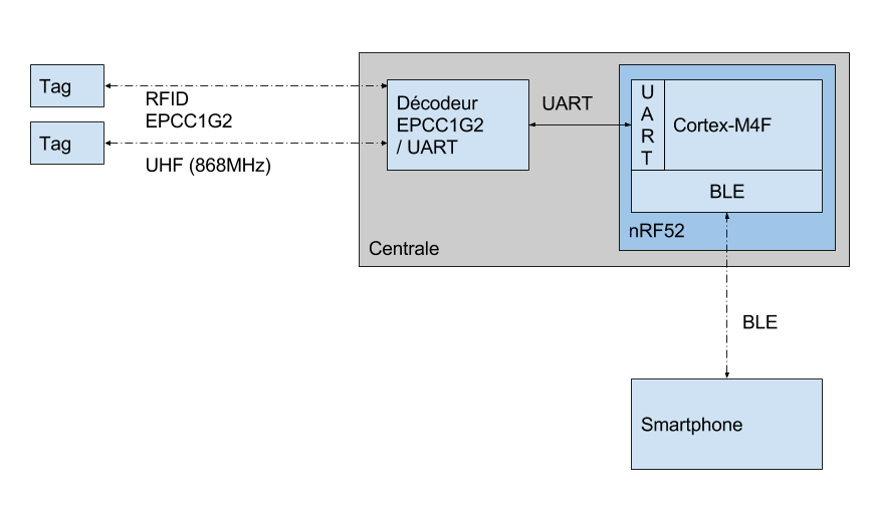
<explication concept smartbag avec image Julia>

## Composants

<liste composant>

## Architecture

Ce projet comporte une centrale qui contiendra toute la partie hardware que nous allons concevoir. Cette centrale sera composé par un microcontrolleur avec un périphérique Bluetooth intégré, d’un périphérique RFID et de toute la partie alimentation.



Sachant que la partie BLE est un périphérique intégré au SOC, nous lui communiquerons via des registres. Par contre, vu que le périphérique RFID est exterieur au SOC, il faudra communiquer avec lui à travers le bus UART.

# Microcontrolleur

## Comparatif

<comparatif des soc>

## nRF52

Nous avons choisi ce microcontrôleur car il possède un périphérique Bluetooth intégré et est spécialement optimisé pour une faible consommation. Il possède les caractéristiques suivantes :

<tableau caractéristiques>

## Environnement de développement

### Keil uVision 5

### nRFGoStudio

### Kinetis Protocol Analyzer

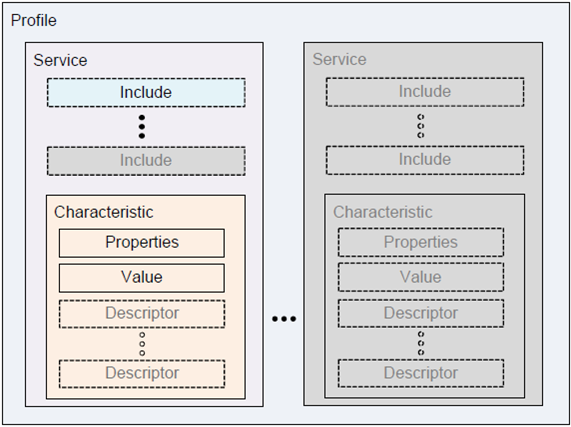
# Bluetooth Low Energy (BLE)

Le smartbag aura une connectivité *Bluetooth Low Energy* pour pouvoir communiquer avec un smartphone. De ce fait, nous allons utiliser le périphérique Radio 2.4GHz intégré dans le nRF52 avec la pile Bluetooth *SoftDevice S132* de *Nordic Semi*. Ce *SoftDevice* est une pile complète supportant le *Bluetooth 4.2* avec plusieurs rôles *BLE* intégré.

Attention !

Le Bluetooth Low Energy est pas mal différents du Bluetooth 2.1EDR

## Generic Attribute Profile (GATT)

Le *GATT* est une structure de données utilisé pour le *BLE* qui définit les messages que les deux périphériques Bluetooth vont s’envoyer.

On peut voir que cette structure complète représente un profil. Et que chaque profil propose des services. Et chaque service possède ses propres caractéristiques.

Un périphérique Bluetooth à l’obligation de s’annoncer, et dans cette annonce il y a l’*ID* de son *GATT* mais pas le contenu du profil. Ce qui implique que pour communiquer avec d’autres périphériques, ils doivent avoir aussi le bon profil.

De ce fait, nous allons créer un profil spécifique pour ce projet, ce qui impliquera qu’il faudra implémenter le *GATT* dans le maitre et l’esclave.

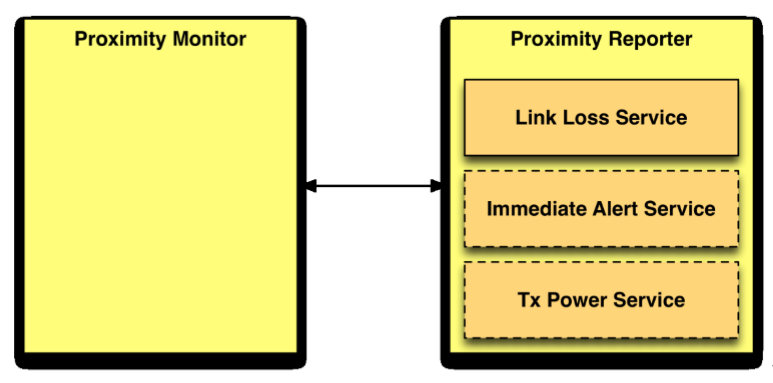
## Profil BlE

Le BLE possède plusieurs profils adoptés pour différentes utilisations. Il n’y a malheureusement pas de profil qui prends en charge tout selon dont nous avons besoin. Attardons-nous sur une particularité du BLE, les profils adoptés existent uniquement pour avoir un standard pour que différents produits qui effectuent les mêmes taches puissent communiquer ensemble. Mais si nous ne trouvons pas le profil qui colle à nos besoins, nous sommes libres de créer notre propre profil et c’est ce que nous allons faire là.

Nous n’allons pas créer un profil complet mais plutôt modifier un profil existant pour lui donner les fonctions qui lui manque. Le profil de base que nous allons utiliser est le *Proximity (PXP)*.

### Proximity (PXP)

Le profil *Proximity* est un profil qui permet d’alerter un maitre si l’esclave ne se trouve plus dans sa zone d’émission. La base de ce profil nous sera utile si on implémente une fonction qui alertera l’utilisateur si son sac se retrouve loin de lui.



Ce profil possède deux rôles : Le proximity Monitor et le Proximity Reporter. Il y a deux rôles qui ont chacun une cible différente.

Le *Proximity Monitor* est dédié au maitre.

Et le *Proximity Reporter* est dédié à l’esclave. <…> Ce rôle possède un service obligatoire : le Link Loss Service

### Notre profil

Notre profil va rajouter plusieurs services. Concrètement le maitre va devoir envoyer certaines commandes à la base, ce qui fait que nous allons ajouter un service qui enverra des commandes. < ?>

Et l’esclave va devoir répondre avec une liste d’objets, ce qui fait que nous allons rajouter un service de liste d’objet.

## SoftDevice S132

Nous allons utiliser cette pile dans son rôle de *Peripheral* pour que notre produit soit vu comme un esclave par les smartphones.

# RFID

<collet>

# Alimentation

<batterie avec recharge usb-c>

# Sources

## Bluetooth

https://www.bluetooth.com/specifications/adopted-specifications