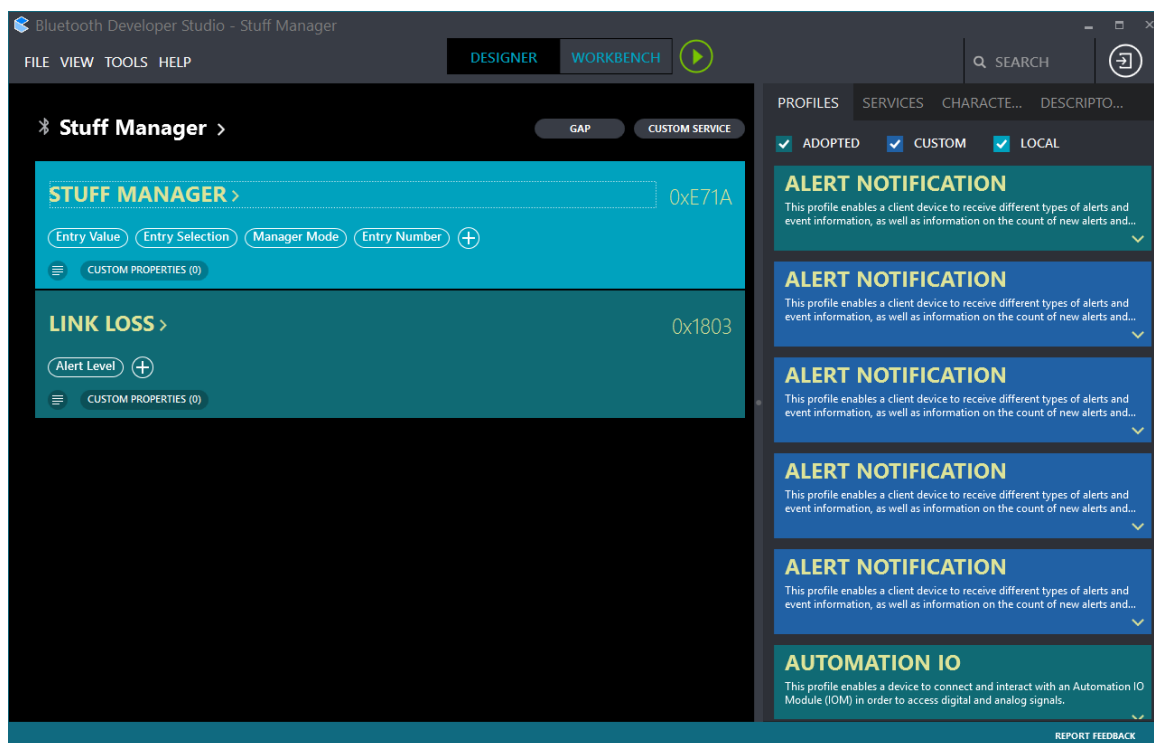


# Annexes

## 1 Bluetooth Developer Studio

BDS a été utilisé pour développer le profil Stuff Manager. C'est un programme proposé par le Bluetooth SIG pour accélérer et faciliter le développement de profils Bluetooth Low Energy. Et un autre des avantages, c'est qu'il y a un système de plugins pour générer du code selon le profil choisi. Nordic en propose un qui génère le code<sup>1</sup> de la partie Bluetooth pour les microcontrôleurs nRF51 et nRF52.



Pour faire son propre profil, on peut commencer sur la base d'un qui est déjà existant et le modifier selon nos besoins, ou tout créer depuis zéro.

L'interface est simple d'accès. Il y a deux fenêtres principales :

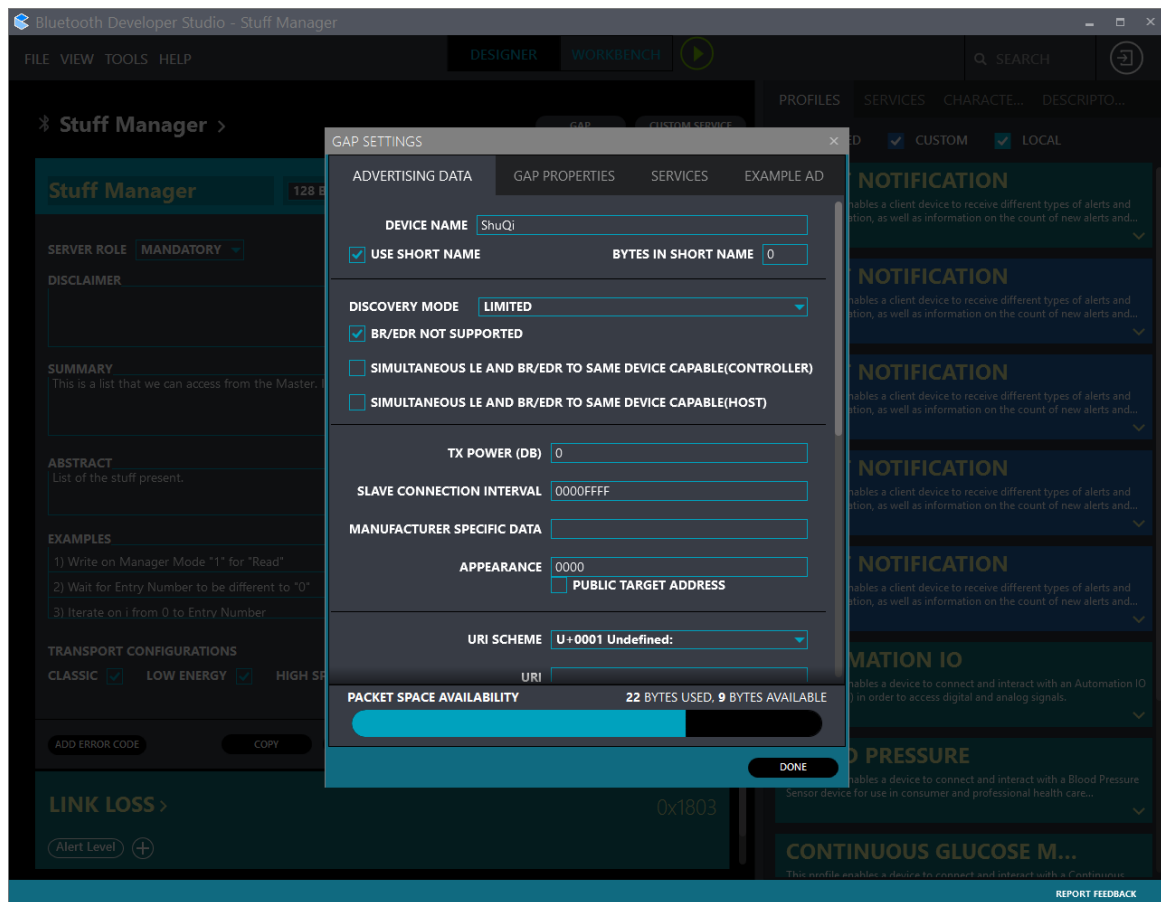
- Fenêtre d'édition
- Liste des profils/services/caractéristiques existant

Et surtout, après l'utilisation de ce logiciel, on peut générer le code pour les services Bluetooth que l'on a défini.

---

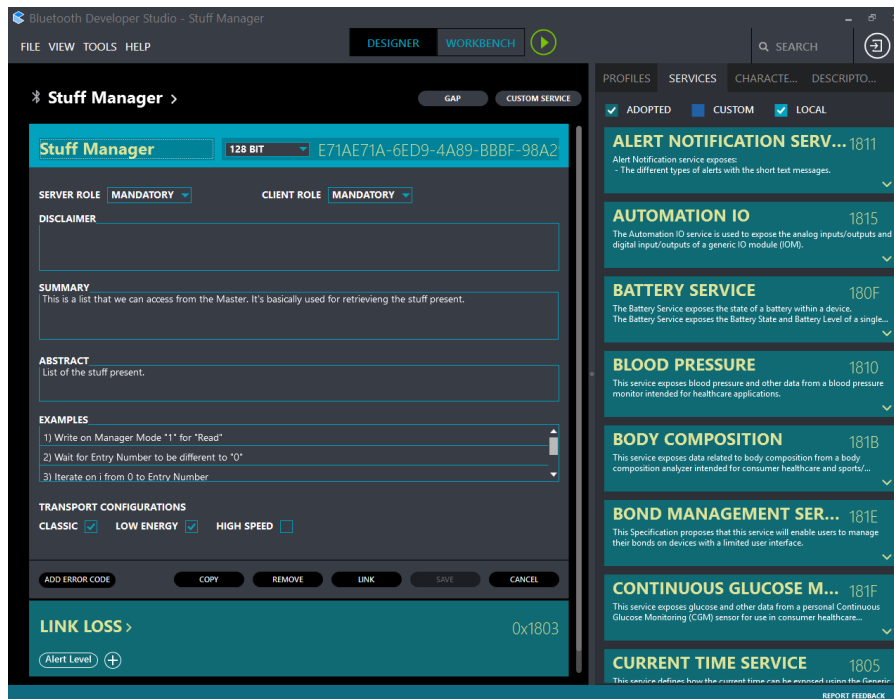
<sup>1</sup> [http://developer.nordicsemi.com/nRF51\\_bluetooth\\_development\\_studio\\_plugin/](http://developer.nordicsemi.com/nRF51_bluetooth_development_studio_plugin/)

Il possède aussi un éditeur pour les paramètres GAP.

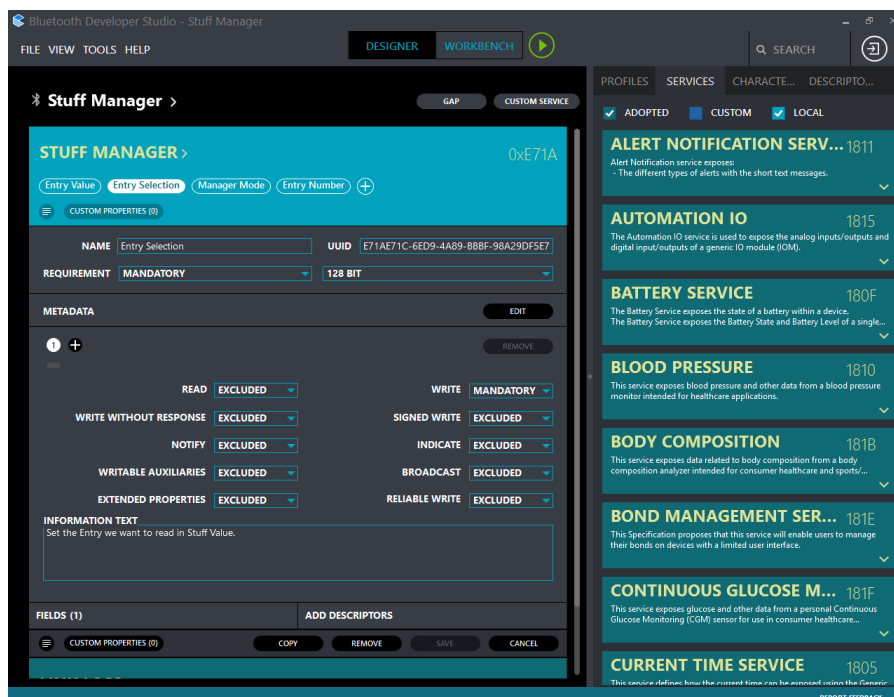


Ça permet de gérer plus facilement l'advertising du Bluetooth.

Il est possible d'éditer le profil, lui rajouter des commentaires et lui définir sur quel type de Bluetooth il peut fonctionner.



Ensuite les services sont éditables aussi, on peut leurs rajouter des caractéristiques. Il peut y avoir plusieurs type d'accès à ces caractéristiques : On peut les lire, écrire, et on peut faire que ce soit obligatoire ou pas.



Pour chaque caractéristique, il faut leurs ajouter des champs, ce qui permet de définir les formats de données qui seront échangés.

## 2 Cordova et Ionic

Pour commencer il faut installer Node.js (version 6 minimum). Pour ça il faut le télécharger sur son site et ensuite le décompresser dans un répertoire. Il faut ensuite aussi installer un JDK pour compiler sur Android. Le paquet gradle a été installé. De même que Java JDK depuis le site officiel. Pour rajouter les exécutable à l'environnement, on fait des liens symboliques dans /usr/bin :

```
ln -s /opt/node-v6.11.0-linux-x64/bin/node node
ln -s /opt/node-v6.11.0-linux-x64/bin/npm npm
ln -s /home/tab/Android/Sdk/tools/android android
```

Et il faut rajouter des variables d'environnement dans le .bashrc :

```
ANDROID_HOME=/home/tab/Android/Sdk
JAVA_HOME=/usr/lib/jvm/java-9-openjdk-amd64/
```

Ensuite il faut installer Cordova et Ionic :

```
npm install -g ionic cordova
```

Pourquoi avoir choisi Cordova et pas Ionic ou Android Studio vanilla ? Parce que en étant sur Cordova ou Ionic ça permet de créer une seule application multi-plateforme. Et ensuite pour le choix entre Ionic et Cordova, c'est le fait que Julia maîtrise mieux le HTML/CSS que le JS.

## 2.1 Cordova

Pour créer un projet Cordova on fait :

```
cordova create ShuQi ch.hepia.ShuQi ShuQi
```

Ensuite il faut ajouter les plateformes. Ici c'est Android que l'on veut (on rajoute la plateforme Browser pour des raisons de débbuging).

```
cordova platform add android
```

Et maintenant on peut lancer l'application avec :

```
cordova run android
```

```
cordova run browser --target=firefox
```

Pour rajouter des plugins, rien de plus simple. On va rajouter le plugin pour interagir avec le BLE :

```
cordova plugin add cordova-plugin-ble-central
```

Et maintenant le projet est prêt !

## 2.2 Ionic

Avec Ionic c'est un peu différent. On créer le projet Ionic avec :

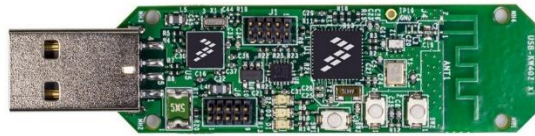
```
ionic start ShuQi
```

Et ensuite pour tester l'application dans un navigateur il faut faire dans le dossier du projet :

```
ionic serve
```

### 3 Kinetis Protocol Analyzer

Un autre matériel très utile, c'est une clé Bluetooth USB associé à Wireshark. Ça permet de capturer les paquets Bluetooth pour pouvoir les analyser.

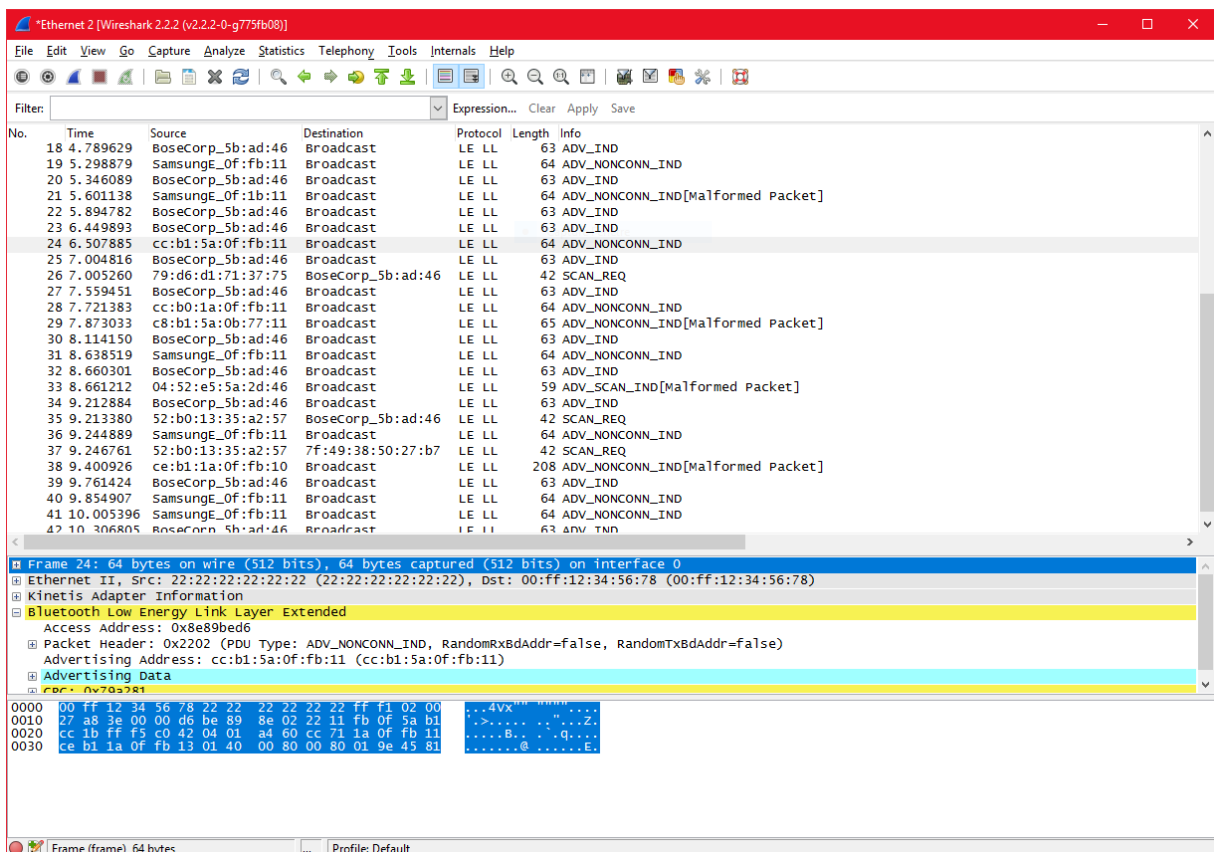


Il faut tout d'abord lancer le Kinetis protocol analyzer adapté qui se présente comme ça :



Ce programme doit détecter tout d'abord la **clé associée**. Ensuite il faut sélectionner les canaux que l'on veut capturer. On peut capturer les canaux 802.15.4 ou plus précisément ceux spécifique au BLE. Pour ça il faut sélectionner les **BLE channels**. Quand tout est configuré, on peut lancer la capture Wireshark avec le **bouton Wireshark**.

Ensuite c'est un Wireshark tout à fait standard qui se lance. Et pour lancer la capture il faut sélectionner la bonne carte réseau (Ici **Ethernet 2**).

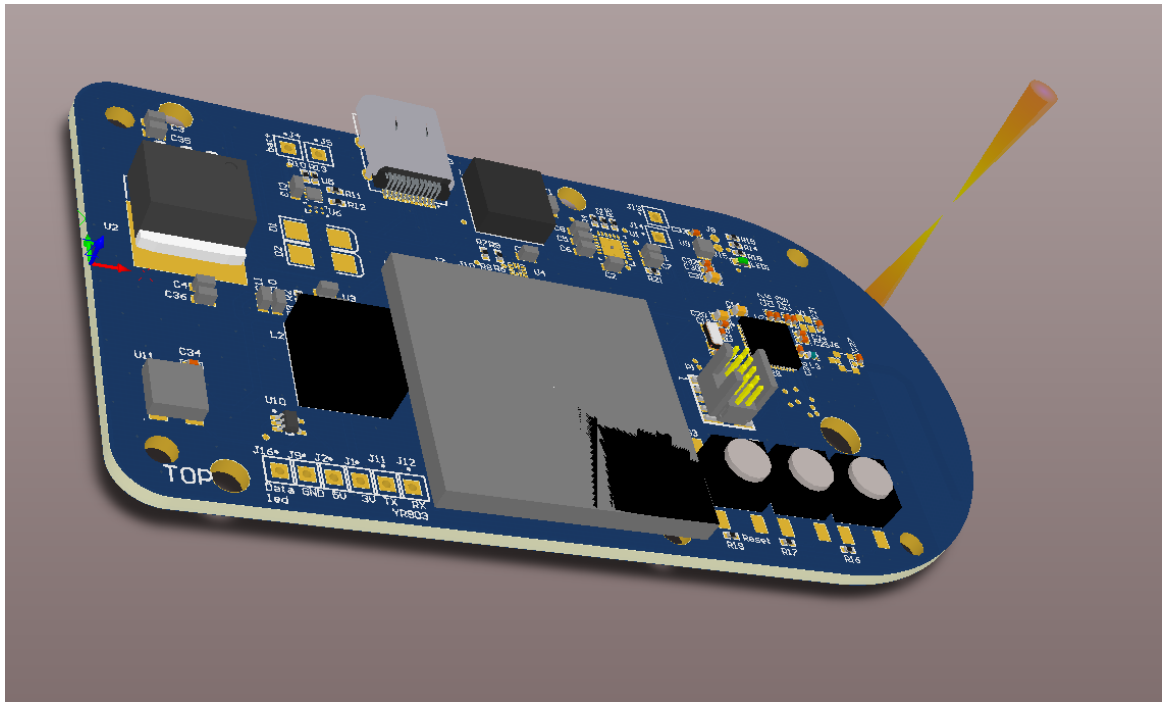


Ce Wireshark est composé de trois fenêtres principales. Il y a tout d'abord la liste des paquets capturés. Ensuite, en sélectionnant un paquet nous pouvons voir les détails du paquet, en décomposé ou en brut.

## 4 Imprimante 3D

Une nouveauté qui a été beaucoup utilisé, l'imprimante 3D. Dans le cadre de la coopération entre les différents corps de métier, il y a toute une partie mécanique : le boîtier.

Pour faciliter le développement du boîtier, on a imprimé le PCB avec l'imprimante. Il est bien plus facile de voir les dimensions avec un objet physique dans la main.



L'imprimante a aussi servi à imprimer un prototype du boîtier, dont la dernière version est actuellement en cours d'impression.

Pour que l'on puisse imprimer, elle a besoin de fichier STL.

Il faut faire attention au filament utilisé, la matière utilisée était le PLA.