|  |
| --- |
| hepia |
| Smartbag - annexes |
| 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)

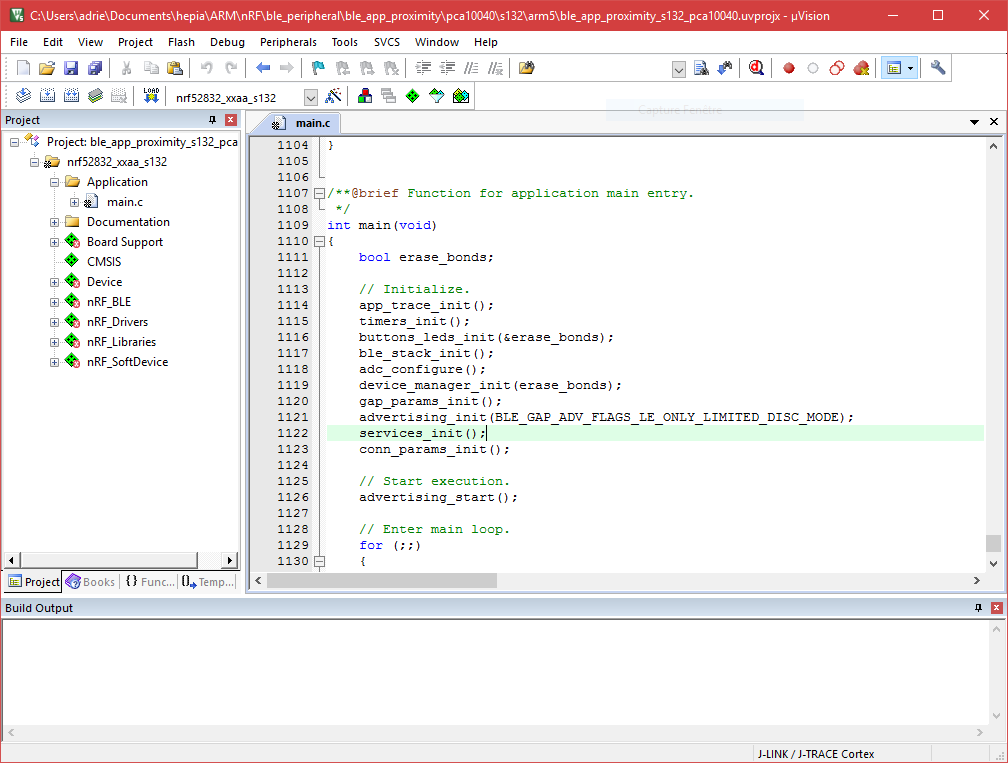
[Bluetooth 6](#_Toc477255146)

# Environnement de développement

La carte étant compatible J-Link.

## Keil uVision 5

L’environnement de développement utilisé pour développer sur la carte est uVision 5 de Keil. C’est un IDE uniquement sur Windows qui prend en charge différents processeur ARM.



L’environnement se présente avec trois fenêtres différentes : L’édition du code, l’arborescence du projet et la console.

### Compilation et upload du programme

L’étape suivante est de compiler le programme. Pour le compiler il faut utiliser le **bouton Build**. Ensuite il faut uploader le programme sur la carte de développement, cela s’effectue avec le **bouton Download**.

Par contre il y a une étape supplémentaire si on utilise le SoftDevice.

## nRFGoStudio

Le microcontrôleur possède un programme qui permet d’effectuer plusieurs actions sur la carte.

## Kinetis Protocol Analyzer

# Utilisation

## Flasher la carte

## Flasher le SoftCore

# Sources

## Bluetooth

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

# TODO

* Changer style doc
* Faire les tutos nrf
* Faire intro ! xD