Tutoriel de prise en main du CmodA7 FPGA

Etude & Conception d'un Smart Sensor sur FPGA

AUTEUR : PAUL LELOUP

TUTEUR : PASCAL BENOIT

THESE DE : GUILLAUME PATRIGEON

2019

Table des matières

Table des matières 1
1. Prérequis 2
1.1. Matériel 2
1.2. Logiciel 2
2. Tutoriel 3
2.1. Mettre en place l'IDE, le Bootload 3
Installer JAVA 3
Installation IDE Eclipse 3
Paramètres du projet (IDE Eclipse) 7
Installation du Codeloader 13
Utilisation du Codeloader 14
Faire clignoter la LED 2 16
2.2. Utiliser l'interface UART avec le module GPS 16
Initialiser l'UART2 16
Les afficher 16
Les stocker 16
2.3. Mettre sous un bon format 16
Machine d'état 16
Créer une trame 16
3. Pour approfondir 16
3.1. Utiliser les autres interfaces du CmodA7, SPI et I2C 16
3.2. Envoyer la trame par LoRa 16

1. Prérequis

1.1. Matériel

Pour le matériel vous aurez besoin de :

- Un ordinateur
- Un FPGA CmodA7 de DIGILENT
- Un module Pmod GPS de DIGILENT
- Un câble USB-microUSB

1.2. Logiciel

Au niveau du logiciel vous aurez besoin de :

- Un IDE, Eclipse CDT :
  - Lien vers le site : https://www.eclipse.org/cdt/downloads.php
  - Téléchargement d'Eclipse version 2018.2 : https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2018-12/R/eclipse-inst-win64.exe

- Le compilateur gcc-arm-none-eabi :
  - Lien vers le site : https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads
    Télécharger la version « 7-2017-q4-major » et cliquer sur « window zip »
    L'extraire et le déplacer dans un dossier à ne pas supprimer.

- Python 3 :
  - Lien vers le site : https://www.python.org/downloads/
  - Téléchargement de Python 3.7.2 : https://www.python.org/ftp/python/3.7.2/python-3.7.2.exe

- Java JRE :
  - Lien vers le téléchargement https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
    Accepter les droits
    Aller dans « Java SE Runtime Environment 8u201 » et cliquer sur la version « x64 » en « .exe »

- La librairie du cortex M0 est présente dans l'archive du tutoriel
- Les fichiers sources sont présents dans l'archive du tutoriel
- Les fichiers *linker.ld* et *startup.asm* sont présent dans l'archive du tutoriel
- L'outil Codeloader V0.1 en python est présent dans l'archive du tutoriel
- Le datasheet du CmodA7 pour avoir les PINS est présent dans l'archive du tutoriel

2. Tutoriel

2.1. Mettre en place l'IDE, le Bootload

Installer JAVA

Double cliquer sur le .exe

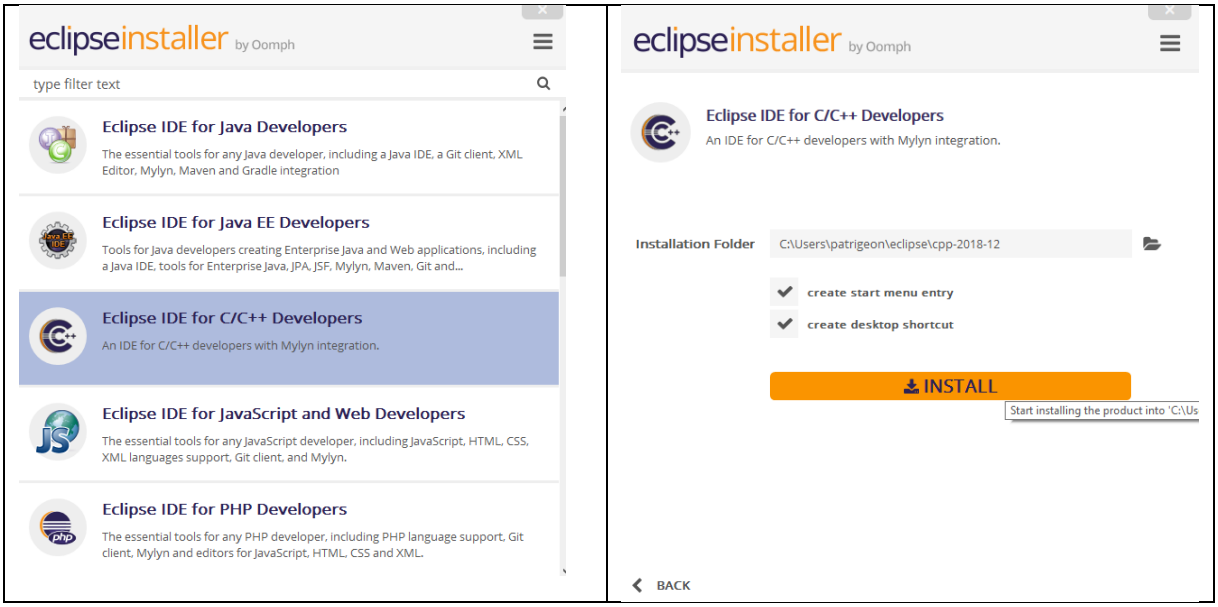
Cliquer sur installer

Cliquer sur fermer

Java est maintenant installé

Installation IDE Eclipse

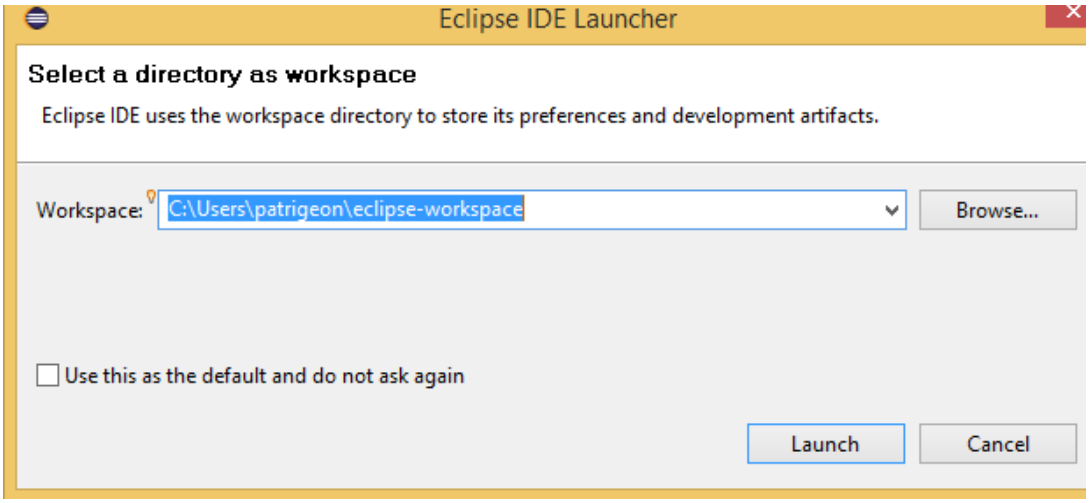
Double cliquer sur le .exe



Accepter toutes les conditions

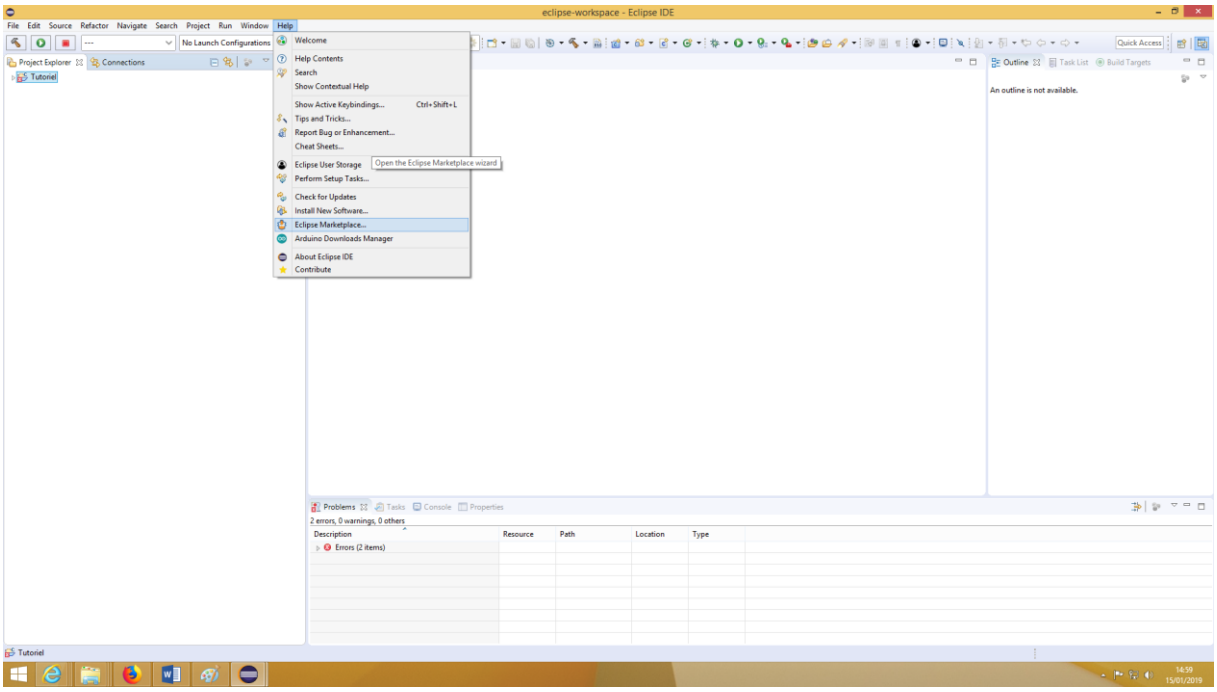
Définir le dossier où tous les projets seront enregistrés :

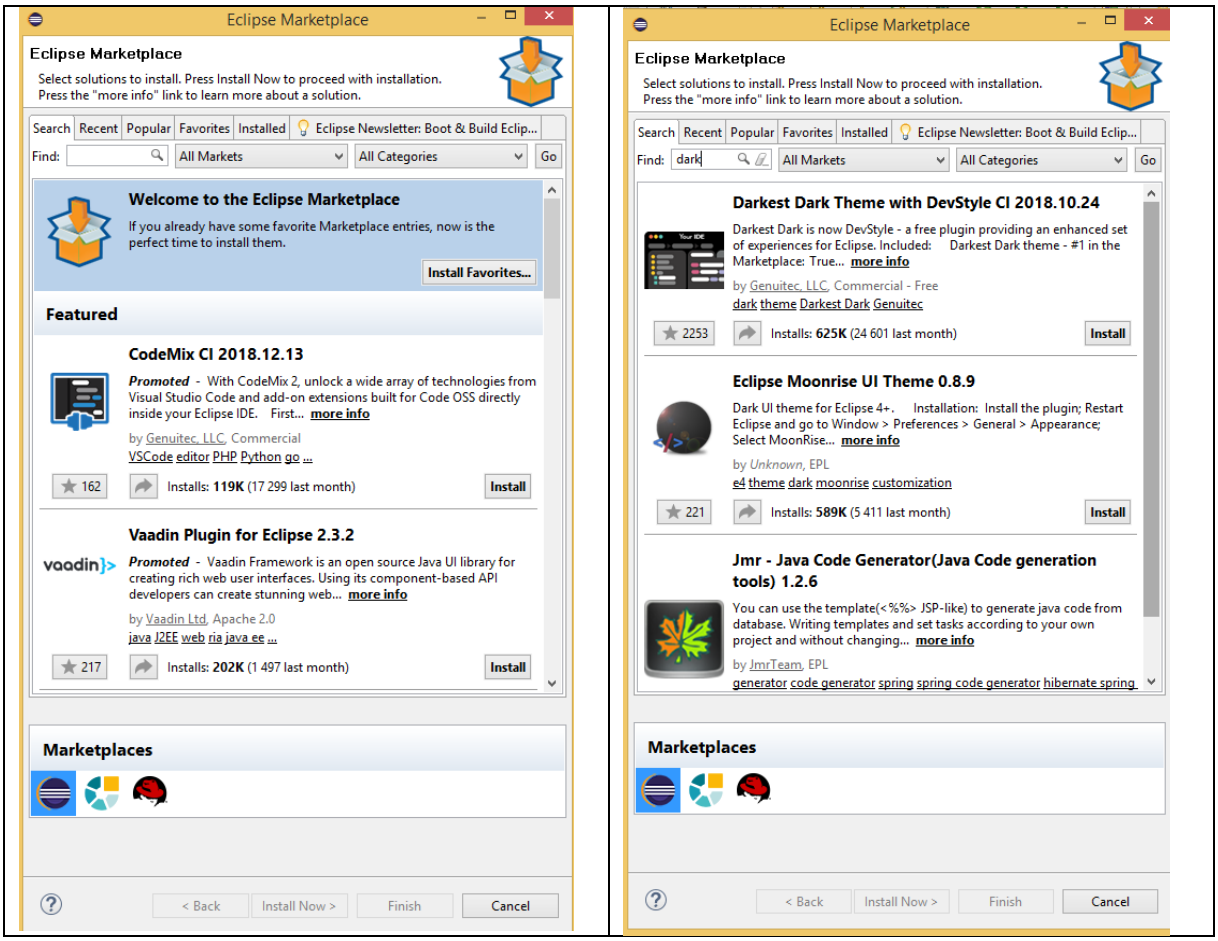
(Bien conserver le chemin qui mène aux projets)



Pour le Dark theme ;

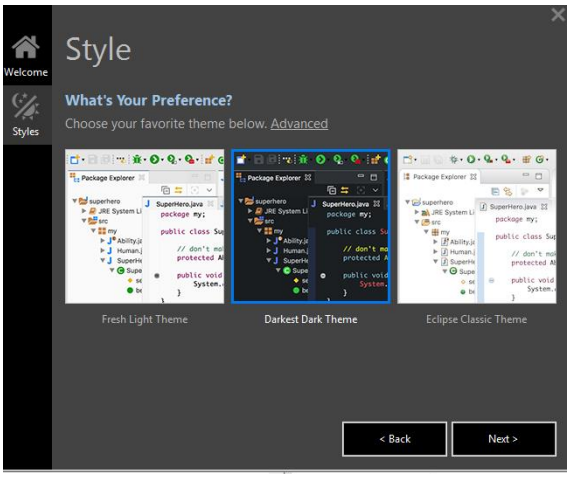
Help/marketplace





Confirmer les conditions

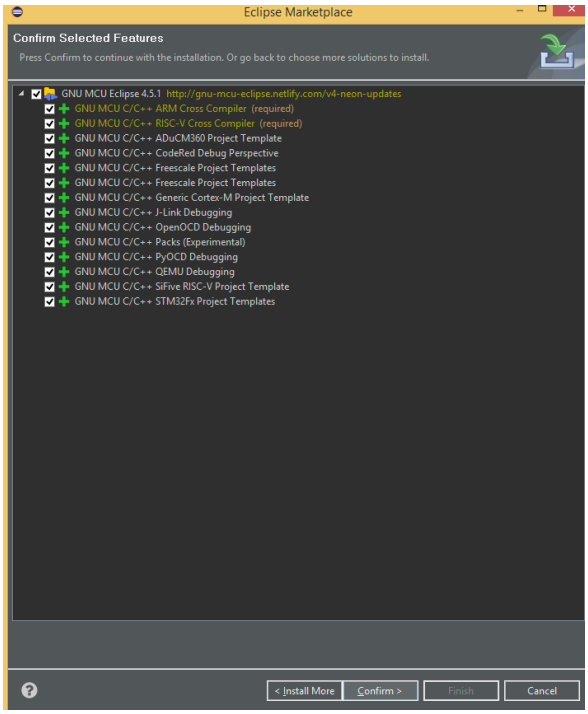
Restart



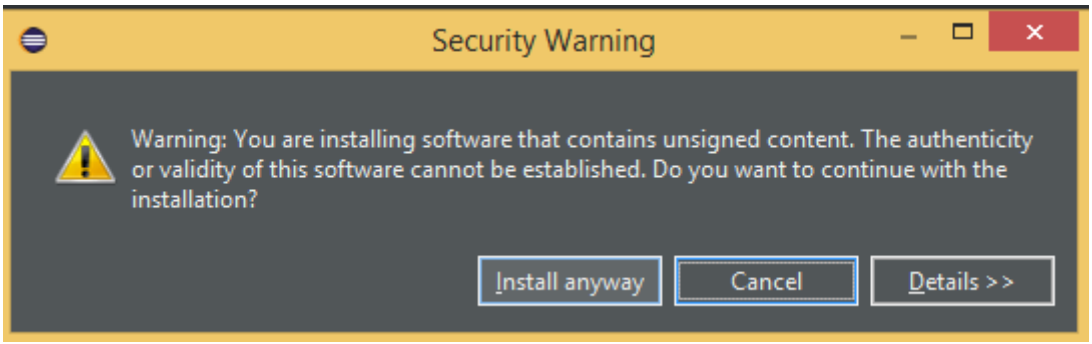
Installer GNU ARM ;

Help/marketplace

Confirmer les conditions



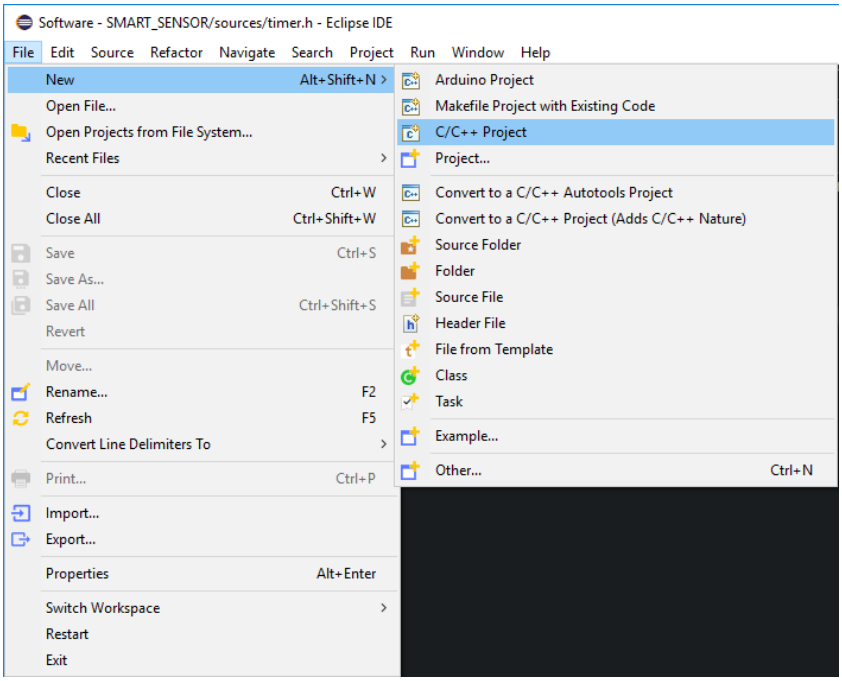
Install anyway



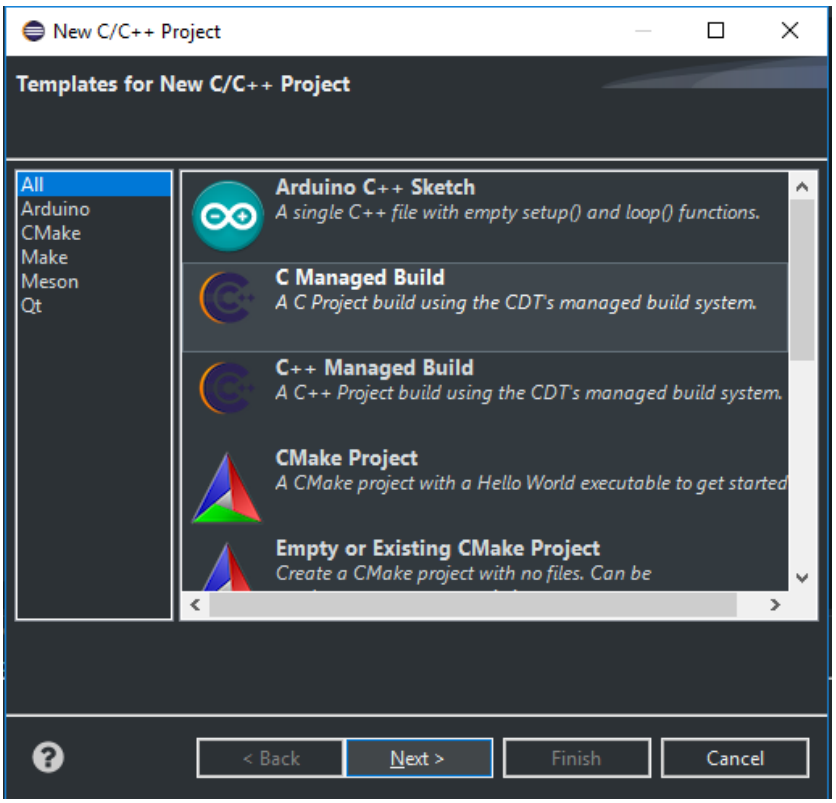
Restart

Paramètres du projet (IDE Eclipse)

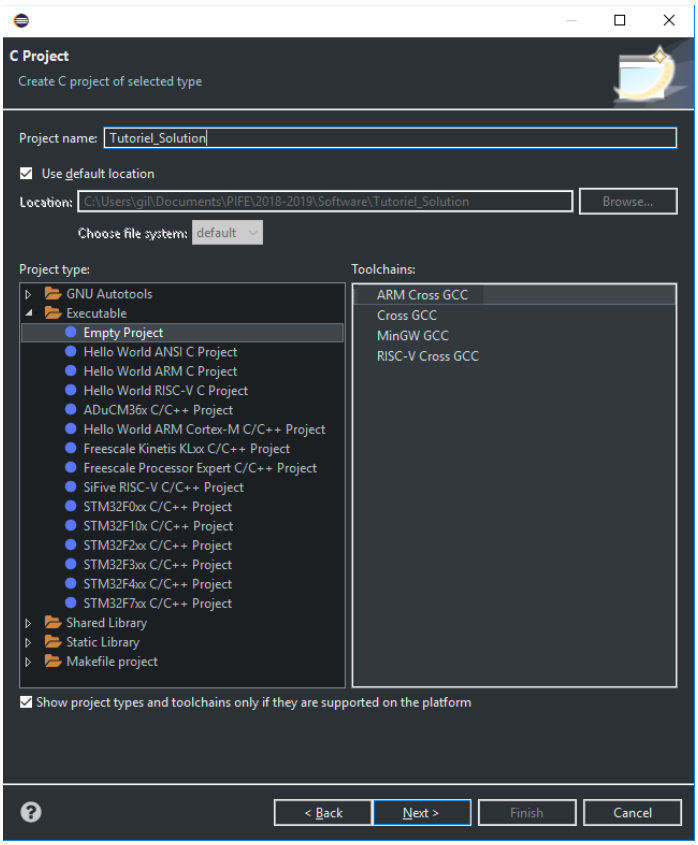
Faire un nouveau projet C/C++



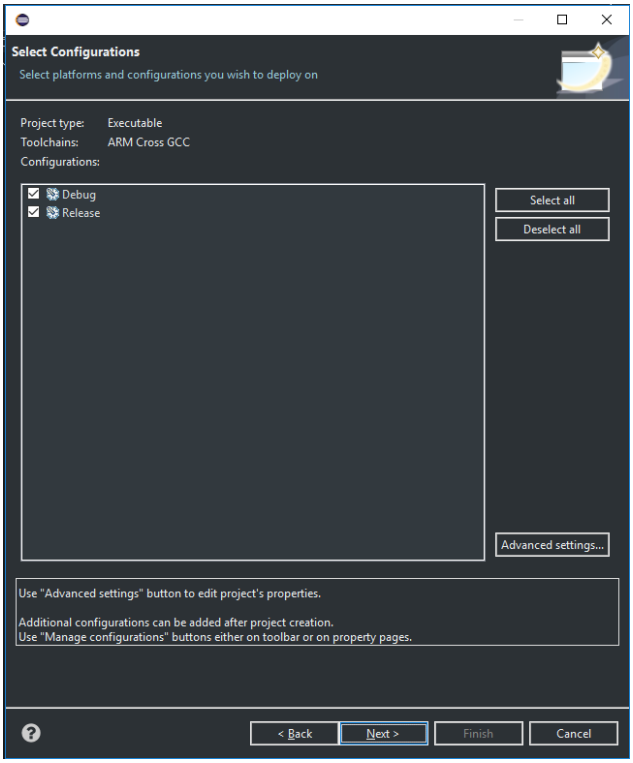
Sélectionner « C Managed Build »



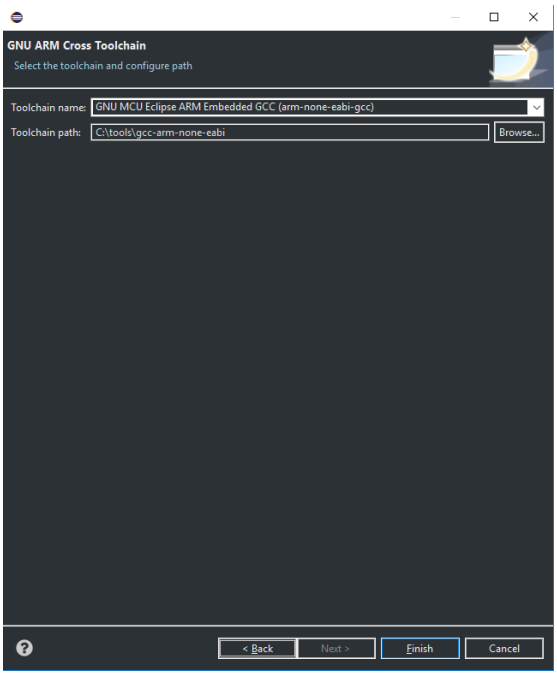
Choisir le nom du projet, selectionner ARM Cross GCC et cliquer sur « Next »



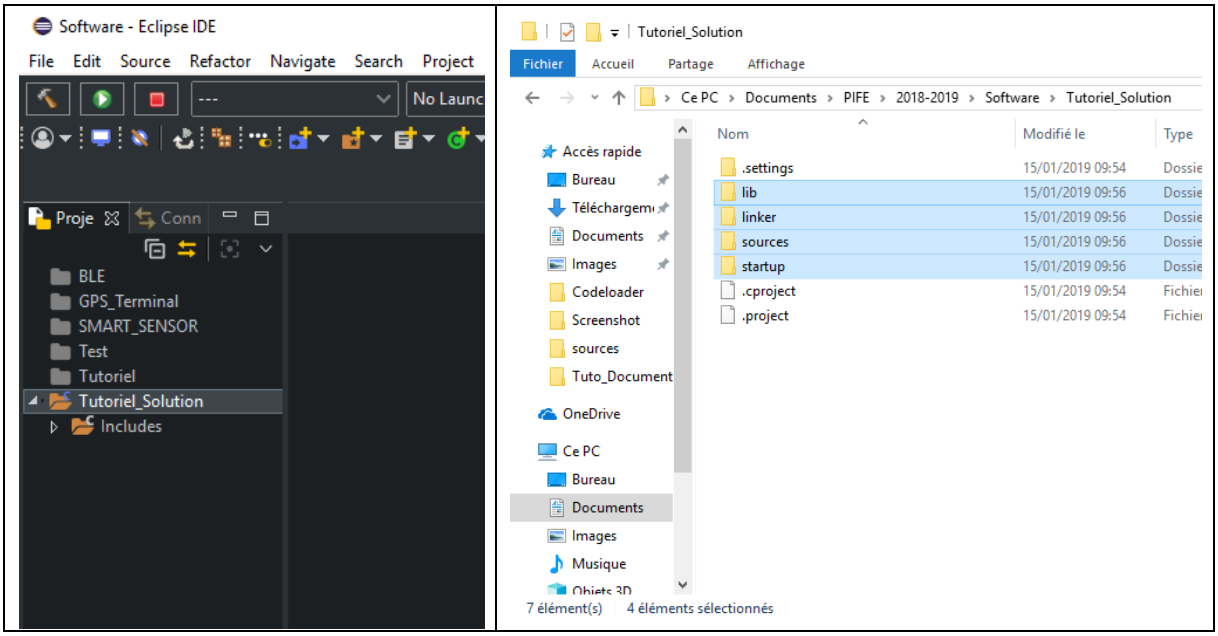
Puis cliquer encore sur « Next »



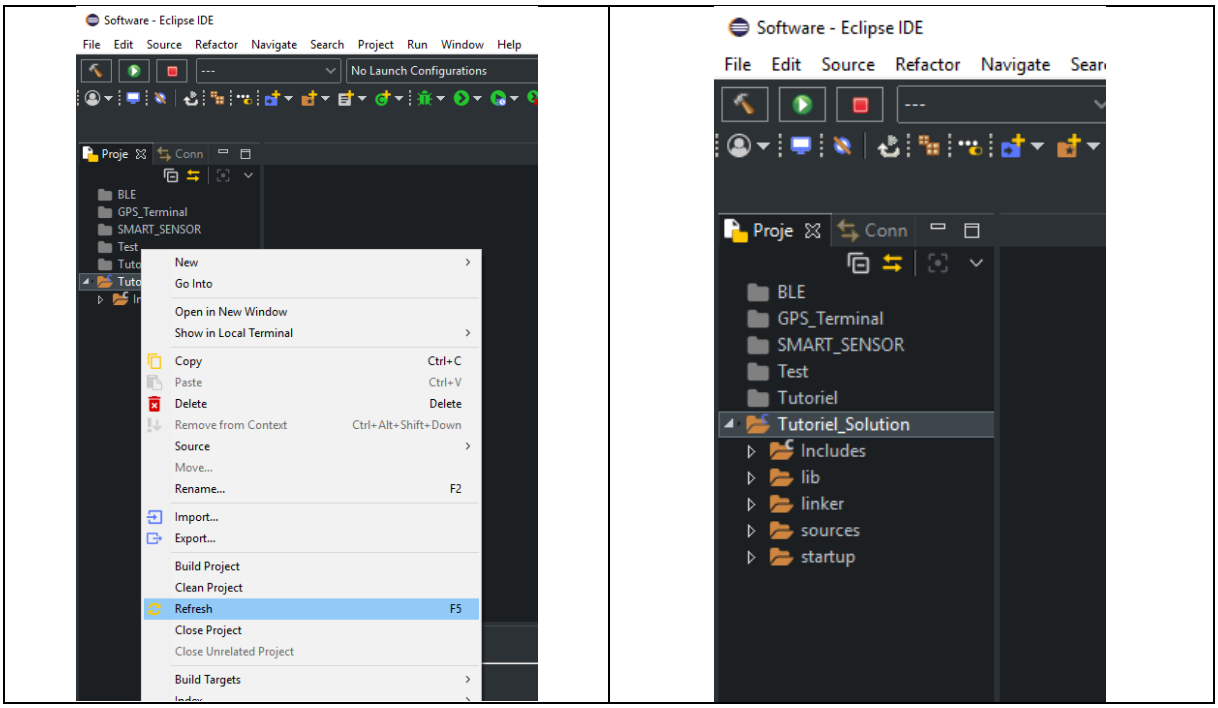
Pour le GNU ARM Cross Toolchain, sélectionner le path de gcc-arm-none-eabi-7-2017-q4-major-win32 préalablement téléchargé.



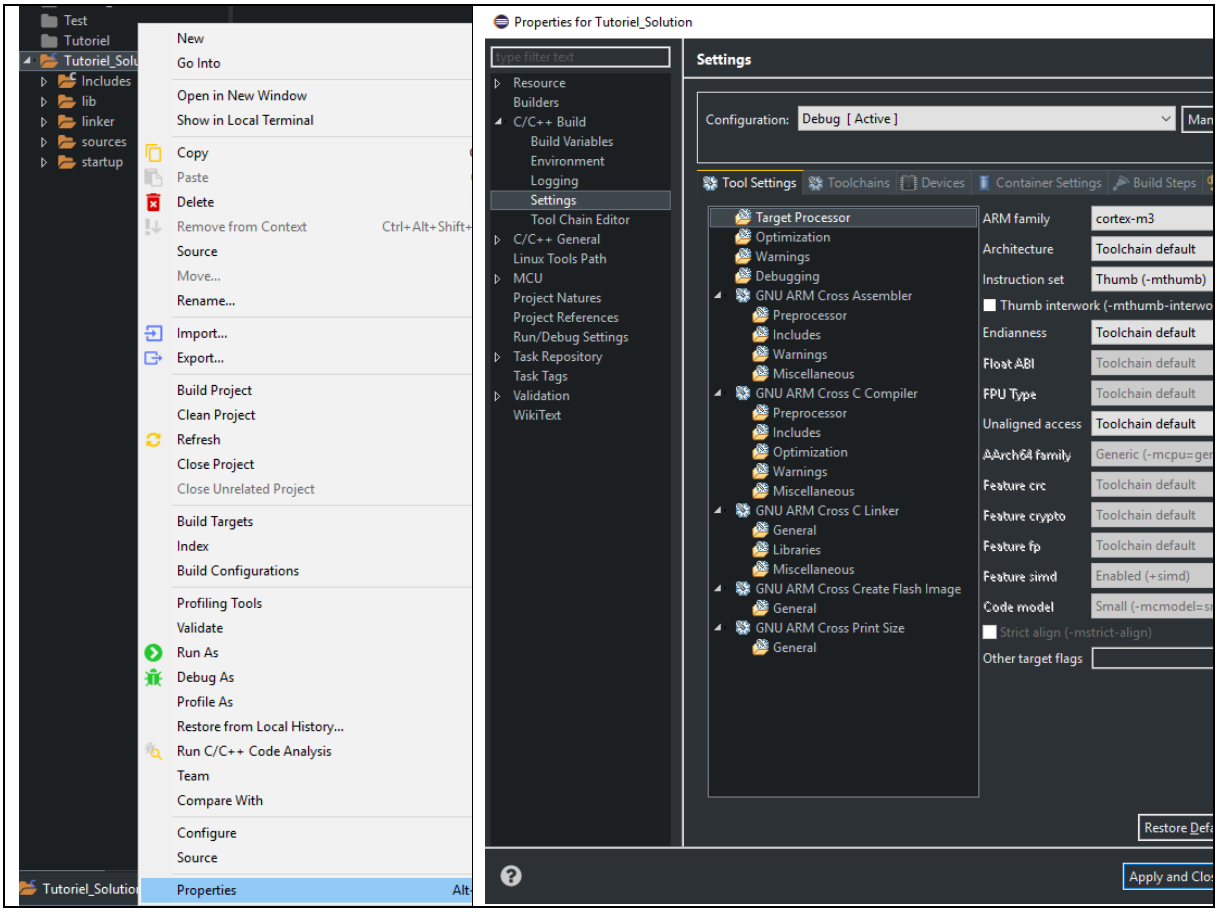
Le dossier compressé « Tutoriel_vX.X » qui vous a été envoyé par mail contient le dossier « Software ». A l'intérieur de celui-ci , vous trouverez 4 dossiers : les fichiers sources, la librairie, le linker et le startup. Ensuite copier/coller ces dossiers dans le dossier de votre projet



Clic droit sur le projet dans Eclipse et cliquer sur Refresh.



Dans l'IDE, selectionner le projet et aller dans les Properties :



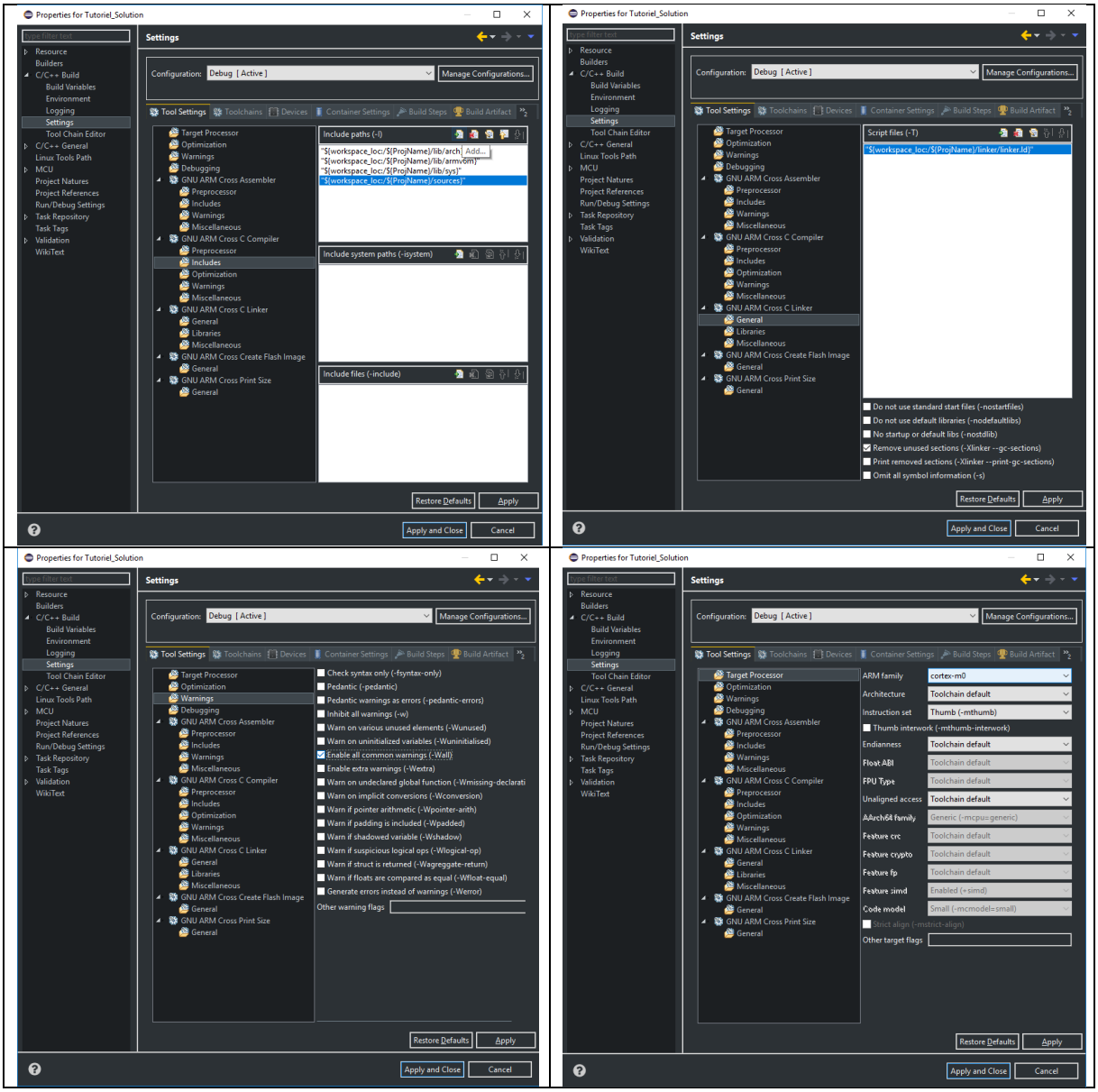
- Properties/C/C++ Build/Settings/Tool Settings/GNU ARM Cross C Compiler/Includes/Include paths
- Ajouter le dossier source et les 3 dossier librairies.

- Properties/C/C++ Build/Settings/Tool Settings/GNU ARM Cross C Linker/General
- Ajouter le fichier linker

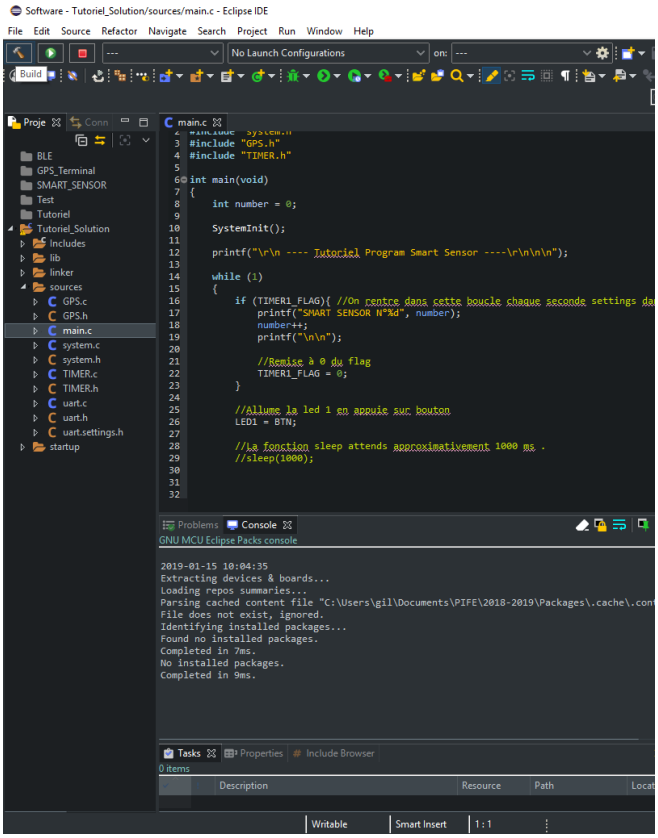
- Properties/C/C++ Build/Settings/Tool Settings/Warnings
- Cocher la case « Enable all common warnings (-Wall)

- Properties/C/C++ Build/Settings/Tool Settings/Target Processor
- Selectionner le cortex-m0 dans ARM family

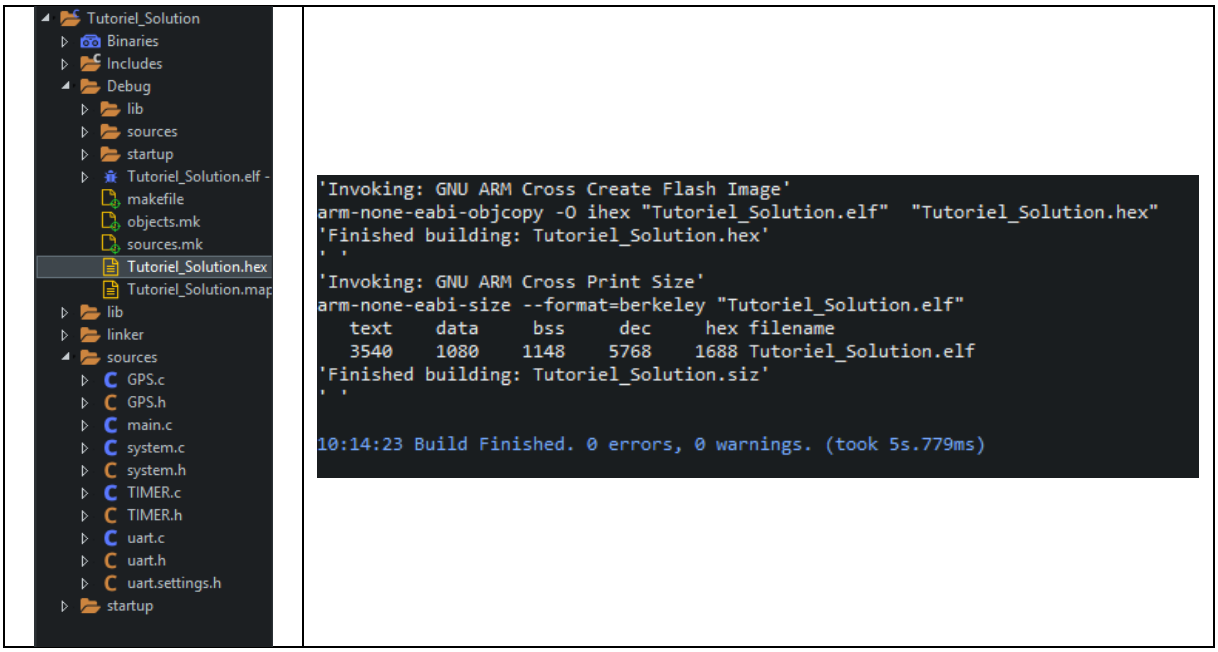
Voici le résultat attendu :



Compiler le projet en cliquant sur le marteau en haut à gauche :

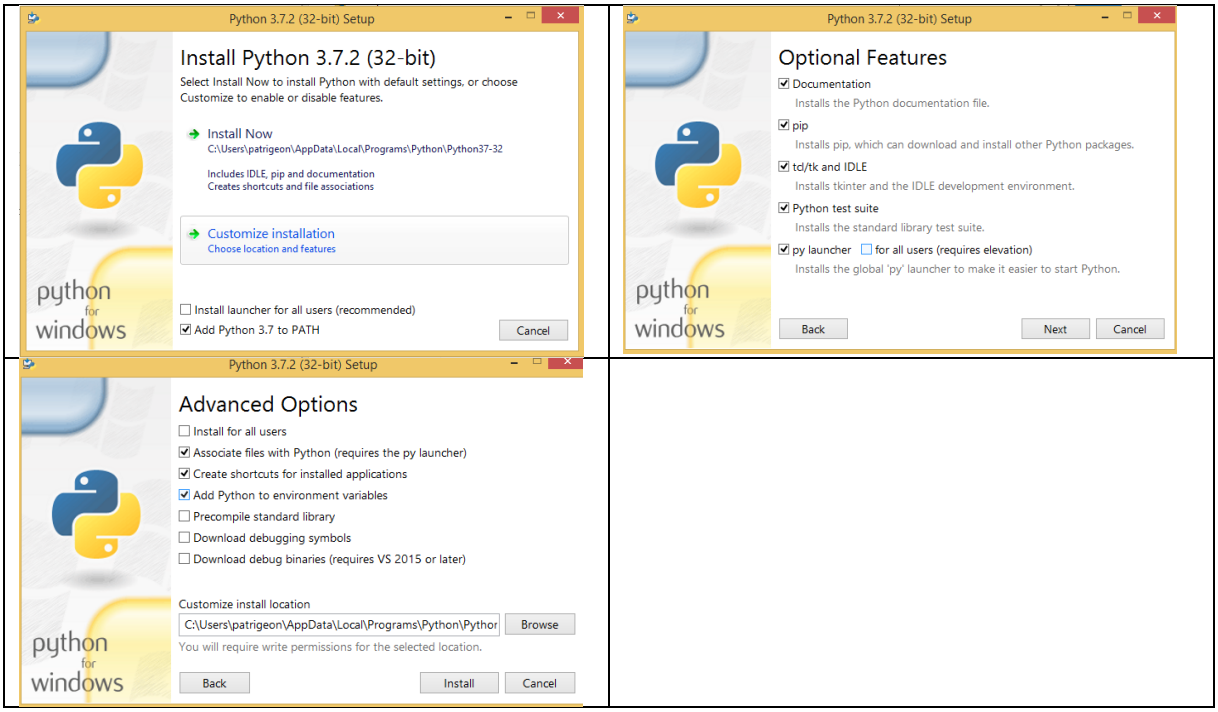


Si tout a été fait sans erreur, l'IDE doit compiler le projet et créer un dossier Debug qui contiendra le programme à télécharger dans le FPGA qui sera le fichier .hex.



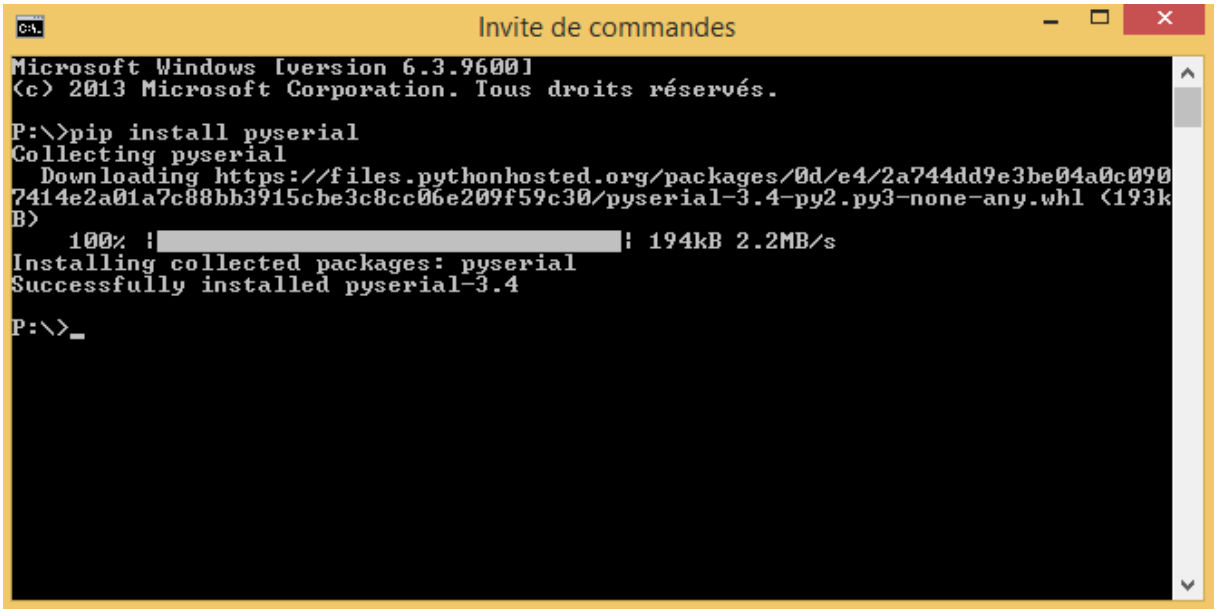
Installation du Codeloader

Lancer Python .exe



Taper cmd dans la barre de recherche windows

Taper : pip install pyserial et taper enter

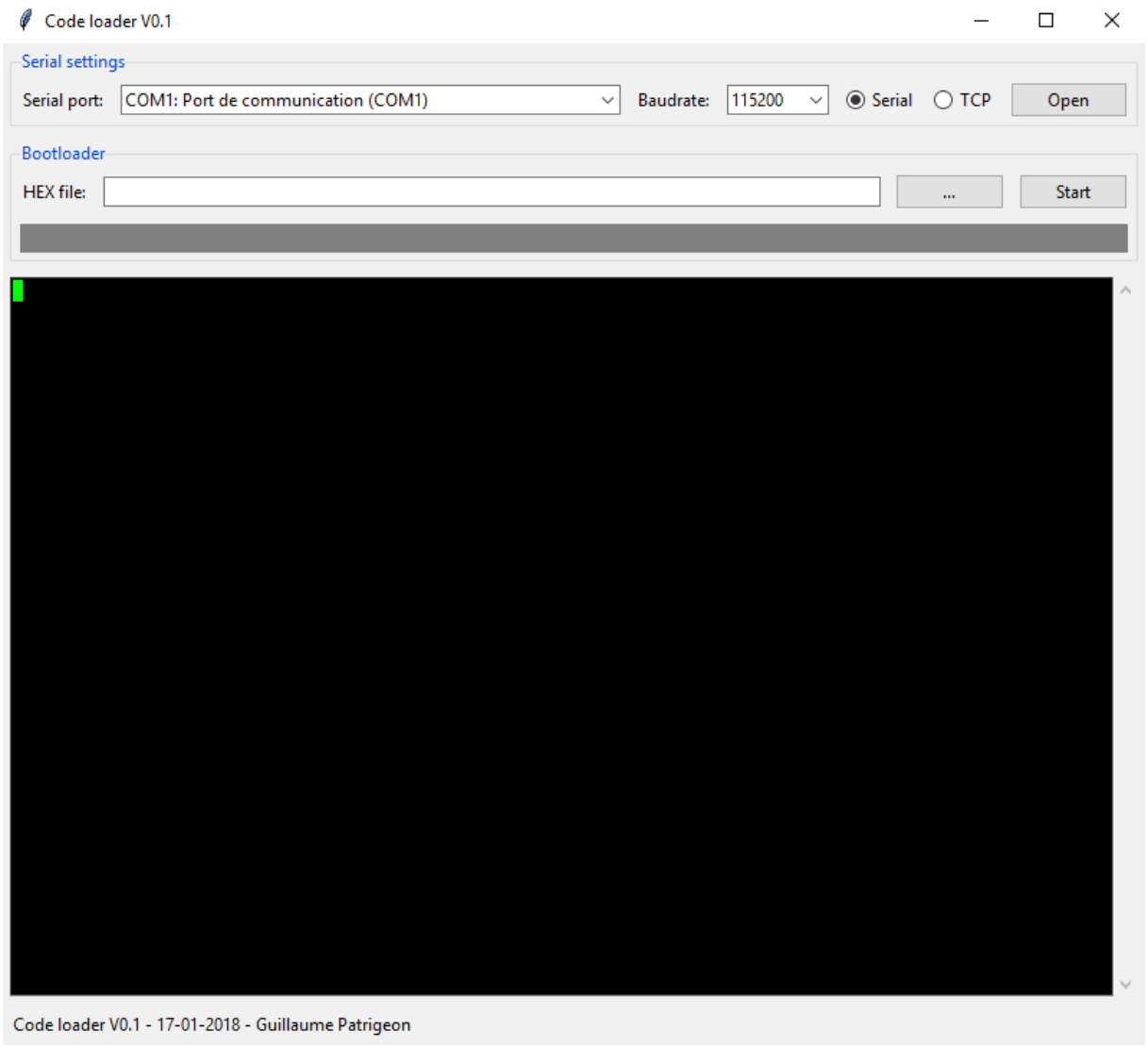


Lancer main.pyw pour verifier la bonne installation !!!!

Il se trouve dans le dossier compressé du tutoriel dans le dossier Codeloader. Décompresser le dossier préalablement.

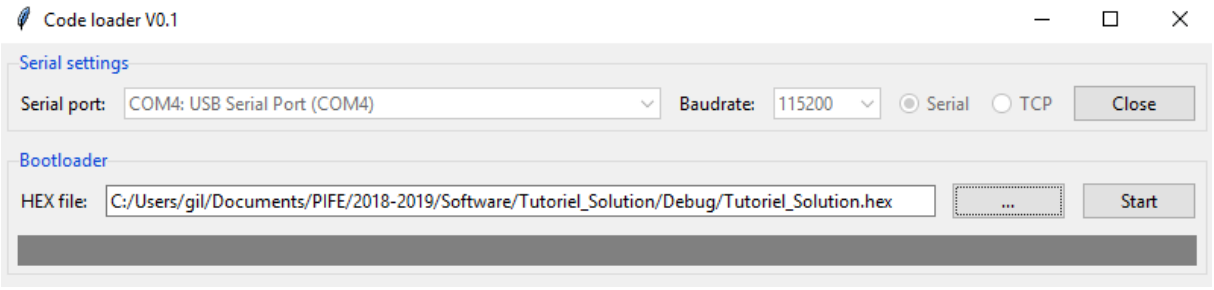
Utilisation du Codeloader

Double cliquez sur le main.pyw pour ouvrir le codeloader, cette fenêtre devra apparaitre :

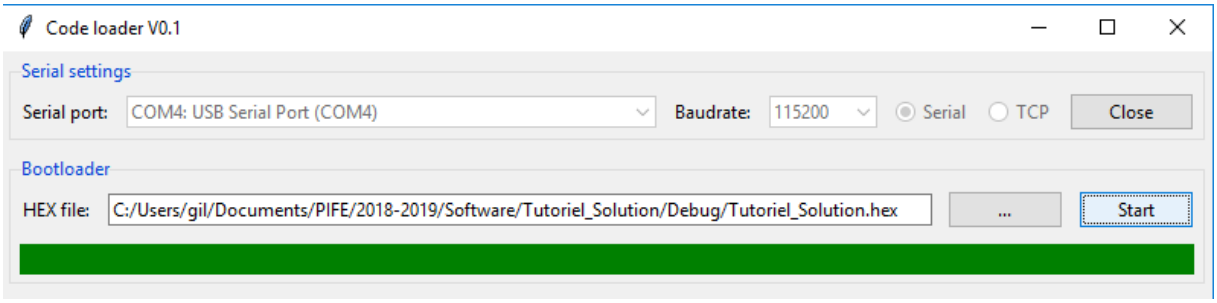


Une fois le Codeloader lancé, sélectionner le port COM où est branché le CmodA7, mettre un Baudrate de 115200 en liaison Serial puis ouvrir la liaison série.

Sélectionner ensuite le fichier hex file et télécharger le code.

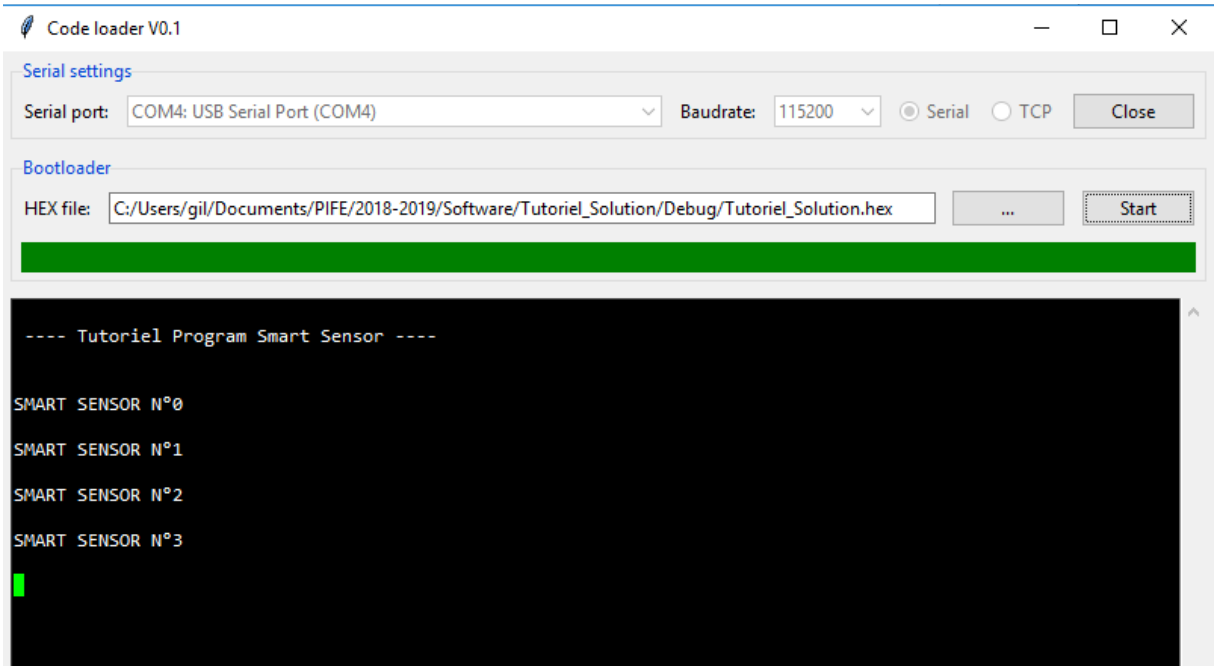


Pour télécharger le code, il faut appuyer sur start et rapidement appuyer sur le reset du CmodA7 qui est le bouton le plus proche de la connectique Pmod.



Si le téléchargement est réussi, la barre de chargement est verte, si le reset n'as pas été appuyé à temps, la barre deviendra rouge.

Si tout a été fait correctement, des informations s'affichent à l'écran et lorsque l'on appuie le bouton BTN1, la LED1 s'allume.



Faire clignoter la LED 2

Adapter le code pour faire clignoter la LED2 avec une période de 2 secondes

2.2. Utiliser l'interface UART avec le module GPS

Initialiser l'UART2

Il faut aller dans le system.c pour initialiser les PIN en entrée/sortie de l'UART2 puis définir l'utilisation de l'UART2 dans uart.setting.h

Les afficher

Renvoyer les données de l'UART2 sur l'UART1 pour l'afficher à l'écran.

Les stocker

Renvoyer les données de l'UART2 dans un tableau, puis afficher le tableau à l'écran.

2.3. Mettre sous un bon format

Machine d'état

Faire une fonction dans GPS.c pour n'afficher qu'une seule trame. Celle qui est intéressantes pour le projet (date, heure, latitude, longitude). Voir la signification des trames sur le datasheet du module.

Mettre cette trame dans un tableau et afficher ce tableau.

Ensuite afficher l'heure, la date de manière lisible pour quelqu'un qui ne connait pas la structure de la trame.

Créer une trame

Créer une trame avec les informations intéressantes dans le but de l'envoyer plus tard par un module LoRa.

3. Pour approfondir

3.1. Utiliser les autres interfaces du CmodA7, SPI et I2C

3.2. Envoyer la trame par LoRa