MISE EN PLACE DE L’ARCHITECTURE DU MICROCONTROLEUR STM32

Atollic/choisir son workspace\_application

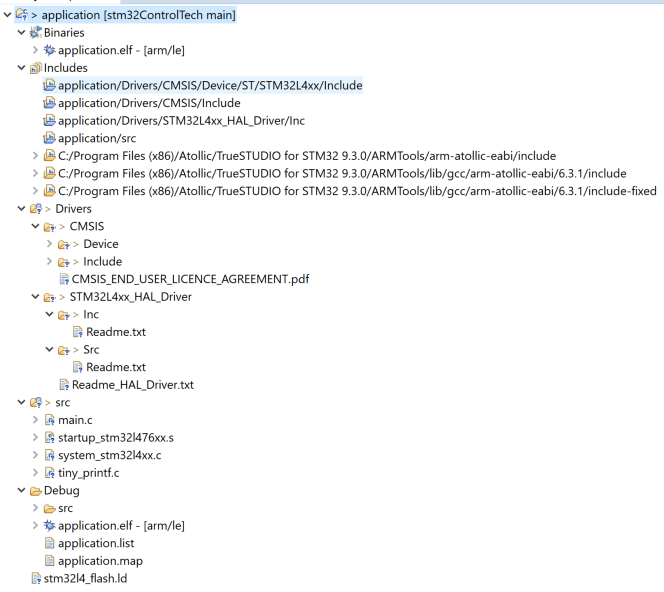
File>New CProject>name (application)

Executable:

Embedded V Project>Atollic ARM Tools (ce qui nous amène à choisir le stm32L476RG)

STM32L4>NUCLEO-L476RG

Voici l’architecture d’origine :



Afin de coller à l’architecture nous allons ajouter des dossiers et fichiers.

Ajout dans dossier Drivers

Le dossier Modules avec les dossiers suivants(GPS, LED, PSU, SD).

Le dossier Sensors avec les dossier suivants (External, Internal)

A la racine nous ajoutons

Le dossier Inc

Middlewares (Environnement, JSON, Peripherals, timer, Network)

Boards

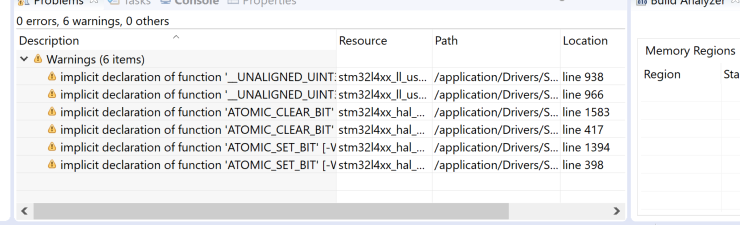
Codecs avec les dossiers suivant (connect>datatype)

Common

Startup

Scripts

Apres resolution HAL il me reste



Installation de l’environnement

Dossier/git clone <https://forge.uca.fr/git/lorawan-node-stepat>

Créer un workspace et ouvrir Atollic dessus. //le workspace contient la configuration.

Configuration de TrueStudio

Help>Eclipse Marketplace (installer le plugin)

Json editor > Install

Configurer l’ide pour utiliser les préférences du projet :

File>Import puis General>Preferences puis Next localiser le fichier uca-connecsens-truestudio.epf (importer le)

Configuration sonde de débogage :

File>Import puis Run/Debug>Launch Configuration

indiquez le répertoire true-launcher cochez le dossier dans la vue de gauche toutes les config>fin.

Une partie des lanceurs font partie de la catégorie External Tools et non de celle de Debug donc

Ajoutons une icône dans la barre d’outils pour accéder plus rapidement :

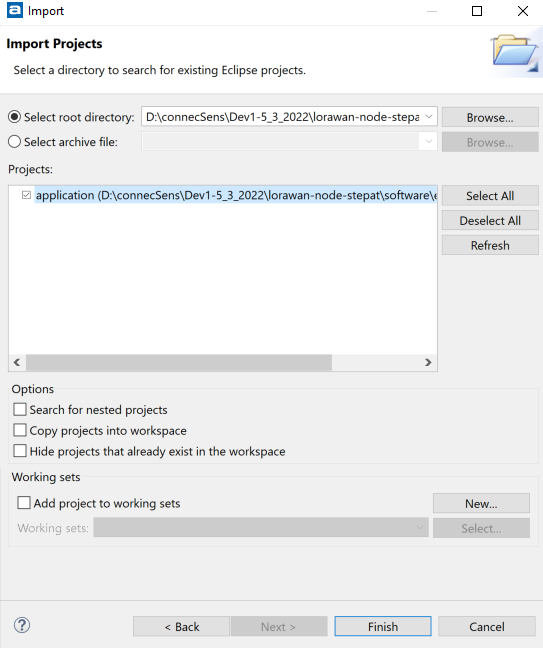
Window>Perspective>Customize Perspective

Cochez Launch>External Tools

Et Cocchez Launch>External Debug

Exporter l’application (de même identique pour le bootloader)

File>Import puis General>Existing projects into Workspace



Nous utilisons trois configurations:

* **release**, qui produit les micrologiciels «officiels» distribuables.
* **debug**, qui se distingue par l'ajout d'un suffixe «**-dev**» dans l'indication de version du micrologiciel.
* **debug-no\_watchdog**, qui désactive le chien de garde, pour qu'il soit possible de lancer une séance de débogage sans être interrompu par celui-ci.

Beaucoup plus simple après le clone vous ouvrez le projet

Le fichier application.elf [arm/le] n’est pas présent il suffit d’aller dans projet/Build setting>C/C++ Build puis dans Tool Chain Editor sélectionner dans Current builder CDT Internal Builder

Et à la compilation il va créer application.elf.