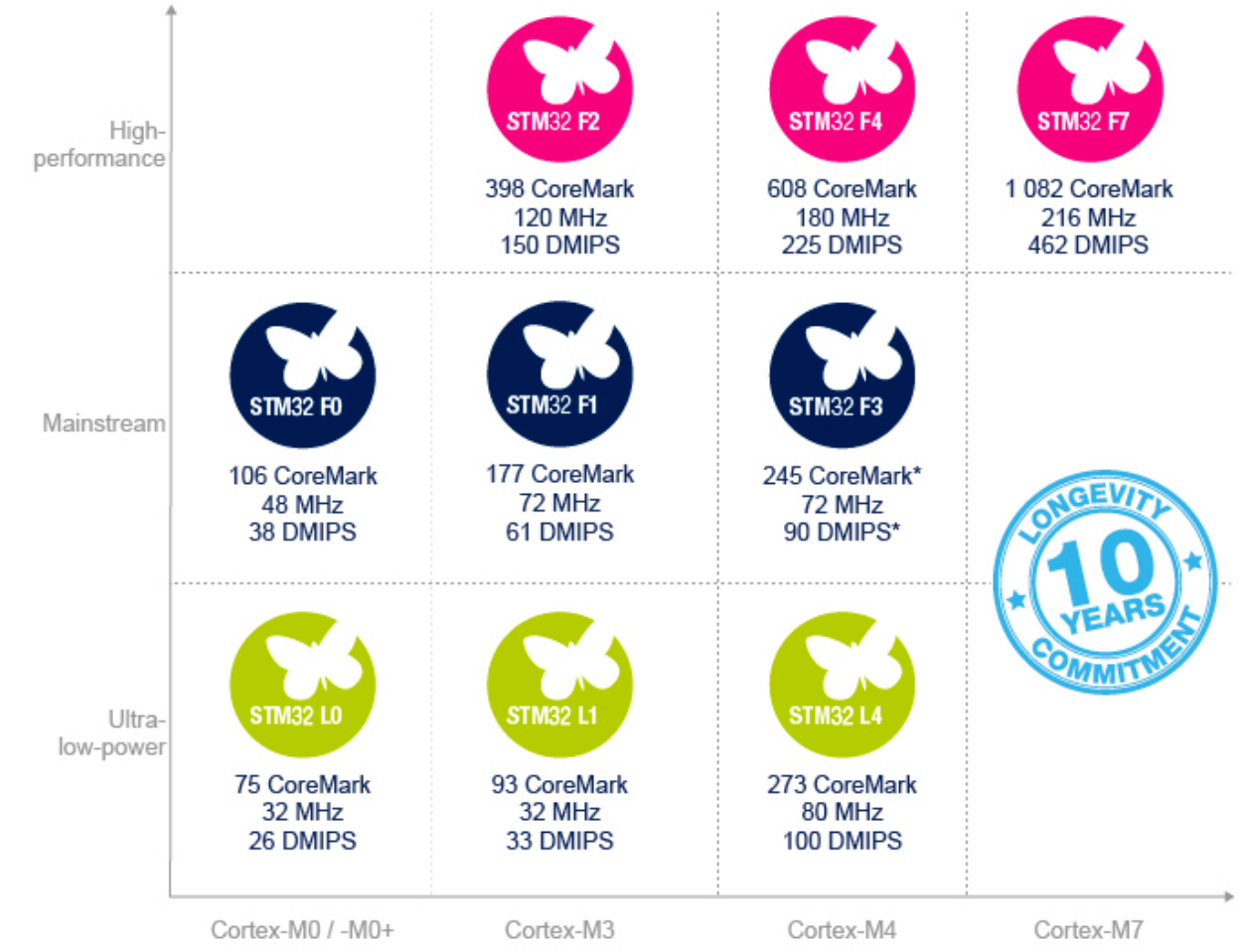
Un apperçu des outils de développement logiciel pour les micro-contrôleurs STM32-ARM-CORTEX de STMicroelectronics

Les outils, cartes et notes d'applications autour du STM32 sont développés en France à Rousset, Bouches du Rhône.

1 STM32 ARM-CORTEX

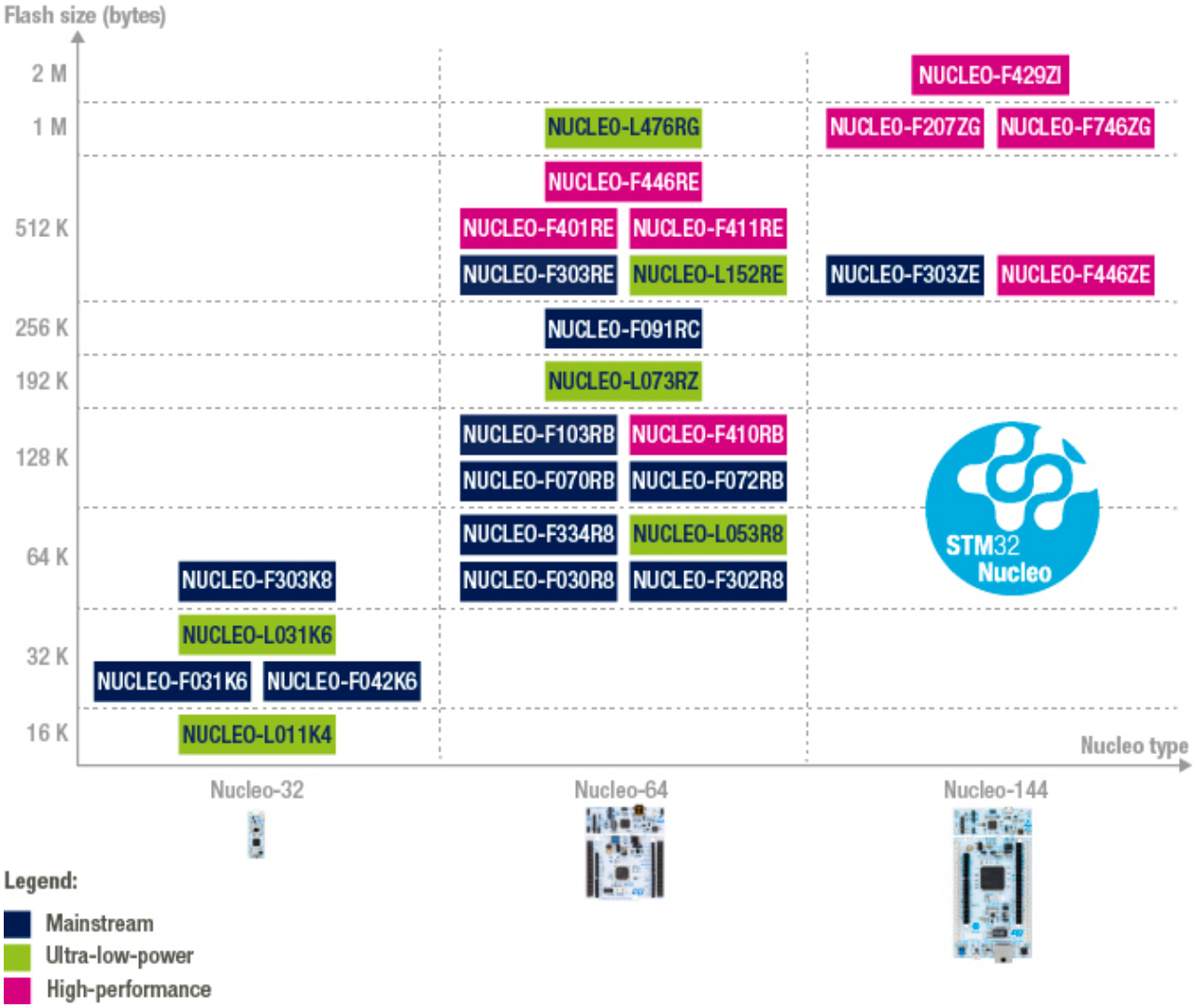
La technologie ARM-CORTEX est utilisée par de nombreux fondeurs dont ST, NXP, ATMEL, MAXIM, Silicon, WIZnet.
Le coeur ARM-CORTEX est également disponible pour des SoC comme Freescale, Infineon, NVIDIA, Apple, Samsung, TI... et de nombreux autres fondeurs.
ST propose toute une série de composants à coeur ARM-CORTEX 32 bits depuis la très faible consomation jusqu'à une puissance de calcul de 462 DMIPS (Dhrystone MIPS)
http://www.st.com/web/en/catalog/mmc/FM141/SC1169?sc=stm32



* from CCM-SRAM

STM32 Outils de développement

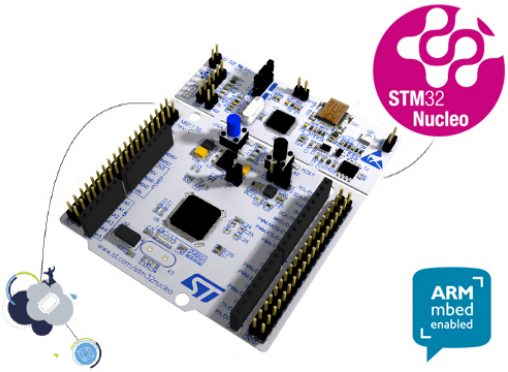
ST propose des cartes de développement à très bas cout (moins de 10€) intégrant un microcontrôleur, un interface de programmation ST-LINK/V2 détachabe, et une connectique de type Arduino.
Ces cartes sont tout particulièrement adaptées à l'étude de faisabilité d'un projet.



STM32 Outils de développement

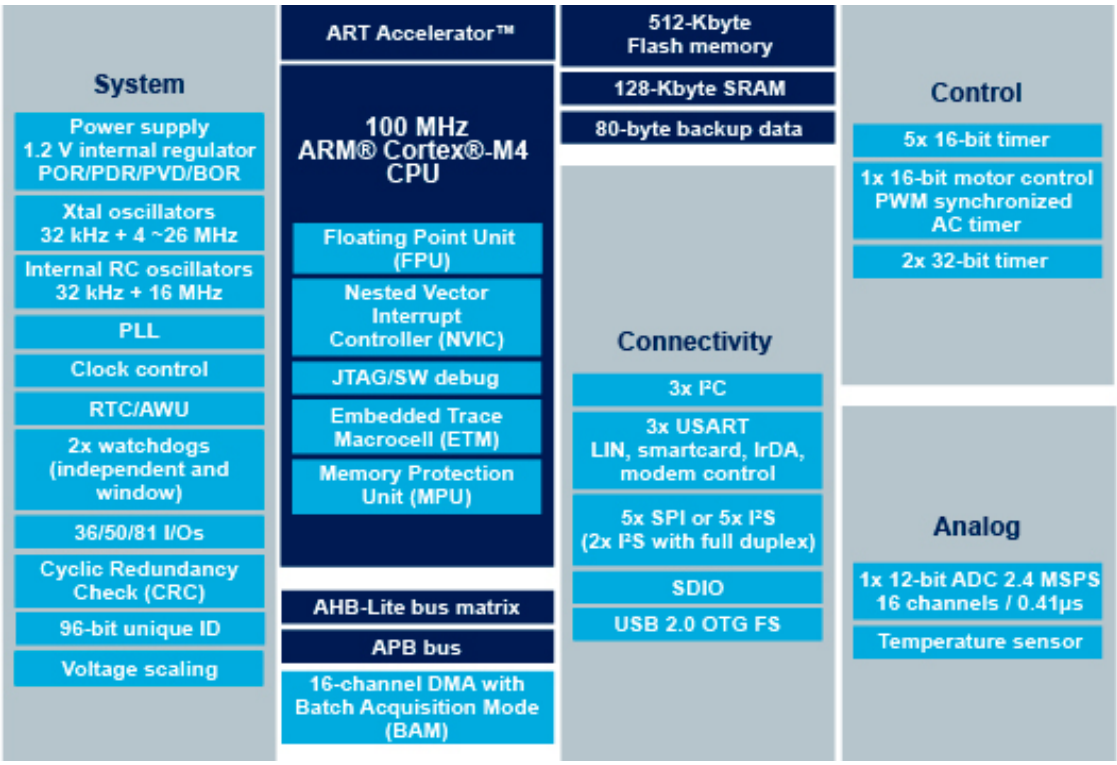
Ce document s'appuie sur les carte NUCLEO-F411RE et NUCLEO-F091RC
Le STM32F411 est un des microcontôleurs les plus puissants.
La licence KEIL µVision pour le STM32F091RC est gratuite (lire le paragraphe correspondant)

http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1847?sc=stm32nucleo



Il est recommandé d'avoir une bonne connaissance de la carte
http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1847/PF260320

et du microcontrôleur utilisé **STM32F411RE**
http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN1877/PF260049?s_searchtype=partnumber



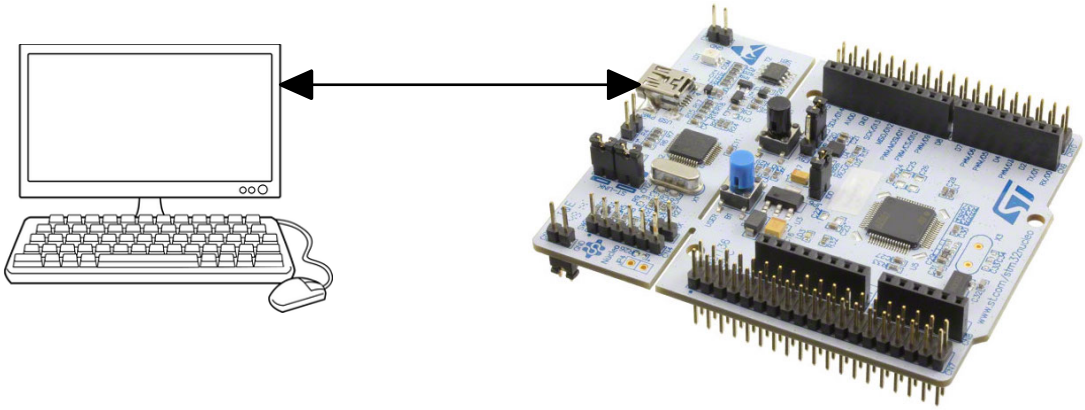
2 Environnement de développement intégré

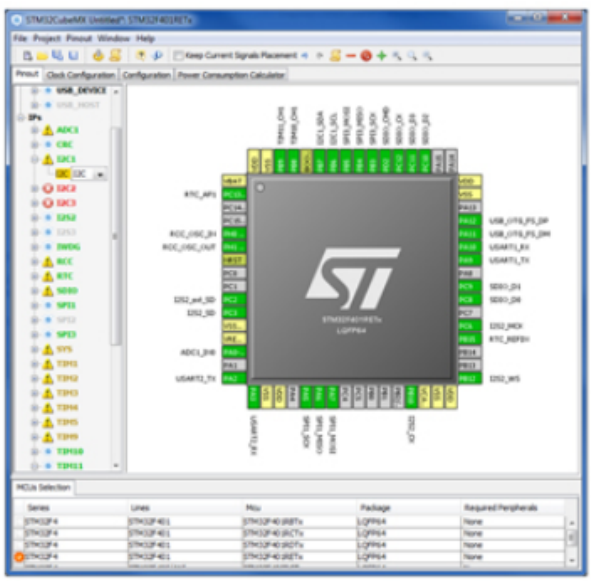
EDI en français et IDE in english

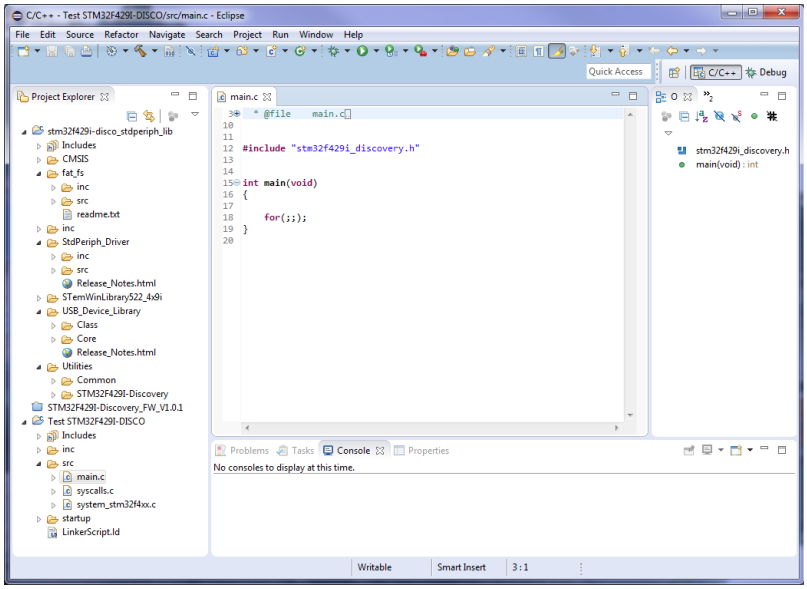
Remarque : Les cartes NUCLEO de ST intègrent l'équipement de programmation du microcontrôleur.
ST propose pour les développeurs de carte un interface ST-LINK/V2 extérieur pour environ 35€

2.1 EDI intégré à l'ordinateur.

Un générateur de code d'initialisation (pas de compilateur)
STM32CUBE est proposé par ST, il permet une configuration simple, rapide et graphique des périphériques et horloges du microcontroleur. STMCUBE génère les fichiers d'initialisation et une puissance bibliothèque appelée HAL.
L'EDI (ici **Eclipse AC6**) permet l'écriture du programme, le téléchargement dans le microcontrôleur et le debug complet. Il est à noter qu'éclipse fonctionne sur Windows, Linux et MACOS
ARM-KEIL propose un EDI sour Windows très performant mais gratuit uniquement pour les STM32F0.

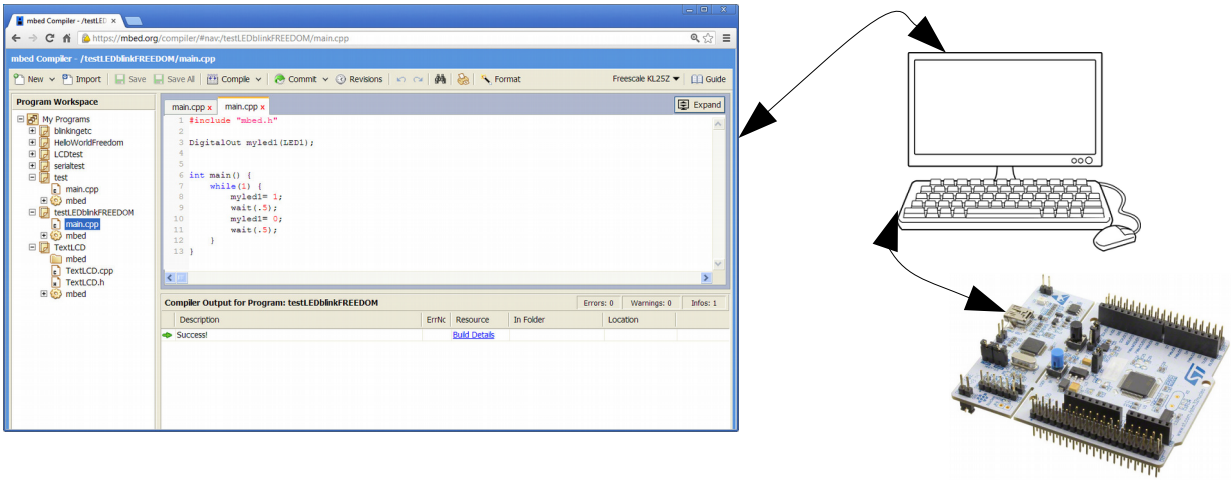






2.2 EDI en ligne (MBED)

La société ARM propose MBED un outil de développement logiciel C/C++ en ligne.
https://www.mbed.com
Cette solution a l'avantage de ne necessiter aucune installation dans l'ordinateur, une tablette et même un smartphone peuvent suffir. Une connexion internet est indispensable.
Par ailleur le debug classique, pas à pas, point d'arrêt, examen des registres et variables est impossible. On rédige le programme, on le compile, on l'essaye.



Nous verrons dans le TP suivant qu'il est possible d'exporter un projet MBED vers Elipse ou µVision.

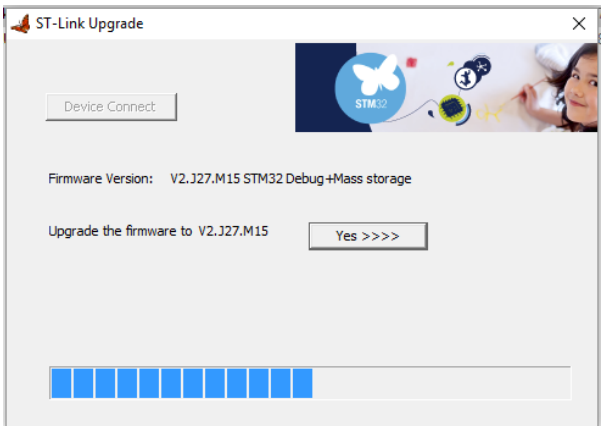
3 MBED, mise en oeuvre

MBED est un environnement de développement intégré gratuit de la société ARM.
MBED permet le développement d'application en C/C++ pour les processeurs/microcontroleurs ARM/ARM-CORTEX
MBED intègre un gestionnaire de projet, compilateur C/C++ ANSI, de très nombreuses bibliothèques de gestion de périphériques.
MBED est un application collaborative, l'objectif d'ARM étant de favoriser l'apprentissage du developpement sur plateforme ARM et les échanges entre développeurs.

Lors d'une première connexion vous devrez vous enregistrez pour accéder aux ressources MBED.
https://www.mbed.com puis cliquer "mbed classic developer site"

Vous êtes maintenant sur votre "mur" MBED.

Nous allons developper un petit projet sur NUCLEO-STM32F411
En haut de la page cliquer " Platforms", cocher ST Microelectronics, cliquer sur la "plateform" NUCLEO-F411RE, puis cliquer "add to your MBED compiler"
Visualiser la petite vidéo de présentation,



En bas de cette page, il est conseillé de mettre à jour le driver ST-LINK de votre PC, faites le
Mettez également à jour "Nucleo ST-LINK/V2 firmware "

Les plans des connecteurs sont donnés ensuite, puis une petite vidéo de prise en main de MBED (à visualiser)

Vous trouverez en bas de page les liens vers la documentation NUCLEO-STM32F411 et sur le microcontrôleur STM32F411

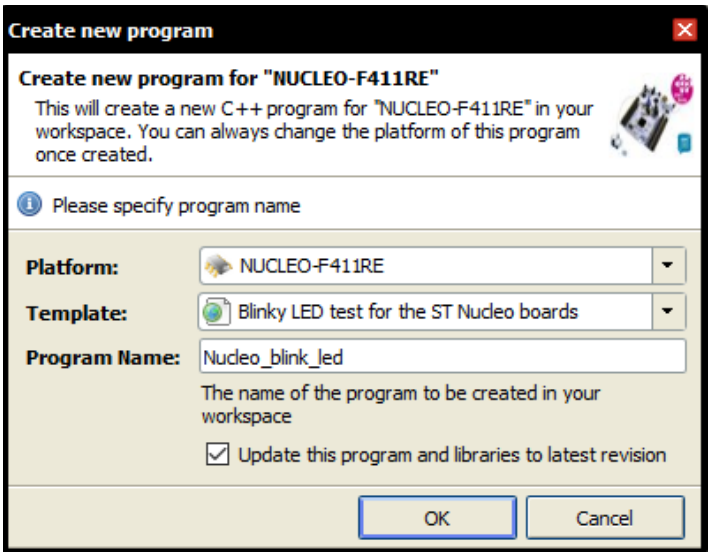
Technical references

For more information, please refer to: STM32F411RE microcontroller Nucleo board

L'onglet "Cookbook" procure de très nombreux exemples
L'onglet " Handbook" donne accès à la documentation complète de MBED

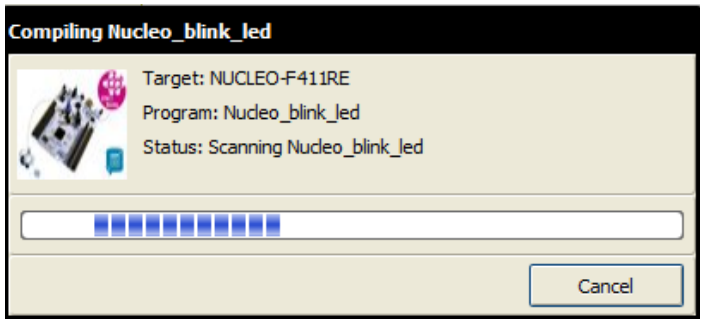
Cliquer maintenant en haut de page sur "Compiler" pour lancer l'EDI.

La carte NUCLEO étant connectée.
Cliquer New-New Program puis selectionner comme suit :



Cliquer main.cpp

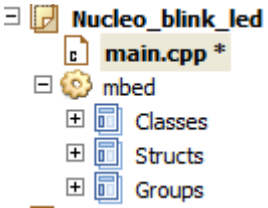
```
#include "mbed.h" // bibliothèque MBED (contient la class DigitalOut)

DigitalOut myled(LED1); // creation d'une instance de DigitalOut sur la LED1
// Le port de la LED1 est de la carte NUCLEO est connu de MBED

int main() {
    while(1)
{
        myled = 1; // LED is ON
        wait(0.2); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}
```



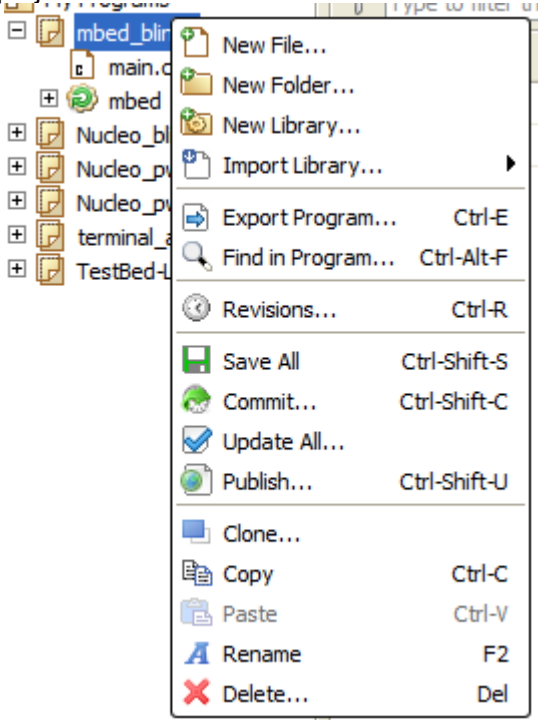
cliquer Compile

Après compilation, enregistrer le fichier Nucleo_blink_led_NUCLEO_F411RE.bin.
La caret NUCLEO apparait comme une clé USB, copier le fichier .bin dedans.
La LED du ST-LINK doit clignoter, ce qui indique un chargement du programme.
Le programme démarre automatiquement, la LED1 clignote, sinon presser le bouton RESET
Remarque: Il est simple d'accéder à un autre port par son nom en remplçant LED1 par PA_10 on accède au bit 10 du PORTA;

Dans la zone projet cliquer sur mbed, puis Classes, puis DigitalOut, ce qui ouvre la documentation de cette classe.



Un clic-droit sur un élément du projet ouvre le menu contextuel



Le bouton NUCLEO-F411RE en haut à droite permet de changer de cible

Créer et essayer un nouveau projet avec l'exmple Nucleo_read_button

```
#include "mbed.h"

DigitalIn mybutton(USER_BUTTON);
DigitalOut myled(LED1);

int main() {
  while(1) {
    if (mybutton == 0) { // Button is pressed
      myled = !myled; // Toggle the LED state
      wait(0.2); // 200 ms
    }
  }
}
```

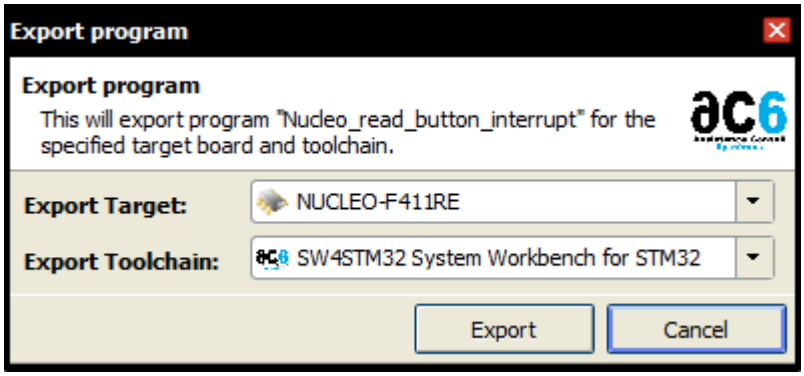
MBED ne permet pas le debug. Il est cependant possible d'exporter un projet MBED vers un EDI permettant le debug.
Clique-droit sur le projet puis Export Program
L'export peut se faire vers des EDI payant comme **Keil µVision** ou IAR, mais également vers des EDI sous Eclipse comme CooCox ou **AC6-Eclipse SW4STM32**

4 Eclipse AC6, mise en oeuvre

La société AC6 (http://www.ac6.fr/) propose des formations orientées développement logiciel.
Elle a développé pous ses supports de cours un EDI gratuit disponible sur le site OpenSTM32, http://www.openstm32.org

System Workbench for STM32 est un EDI développé sur JAVA, il est utilisable indiférent sur Windows et Linux. Sur le site OpenSTM32 aller dans "System Workbench for STM32". Suivez la procédure de téléchargement et d'installation de l'EDI en cliquant sur " Installing System Workbench for STM32 with istaller ".

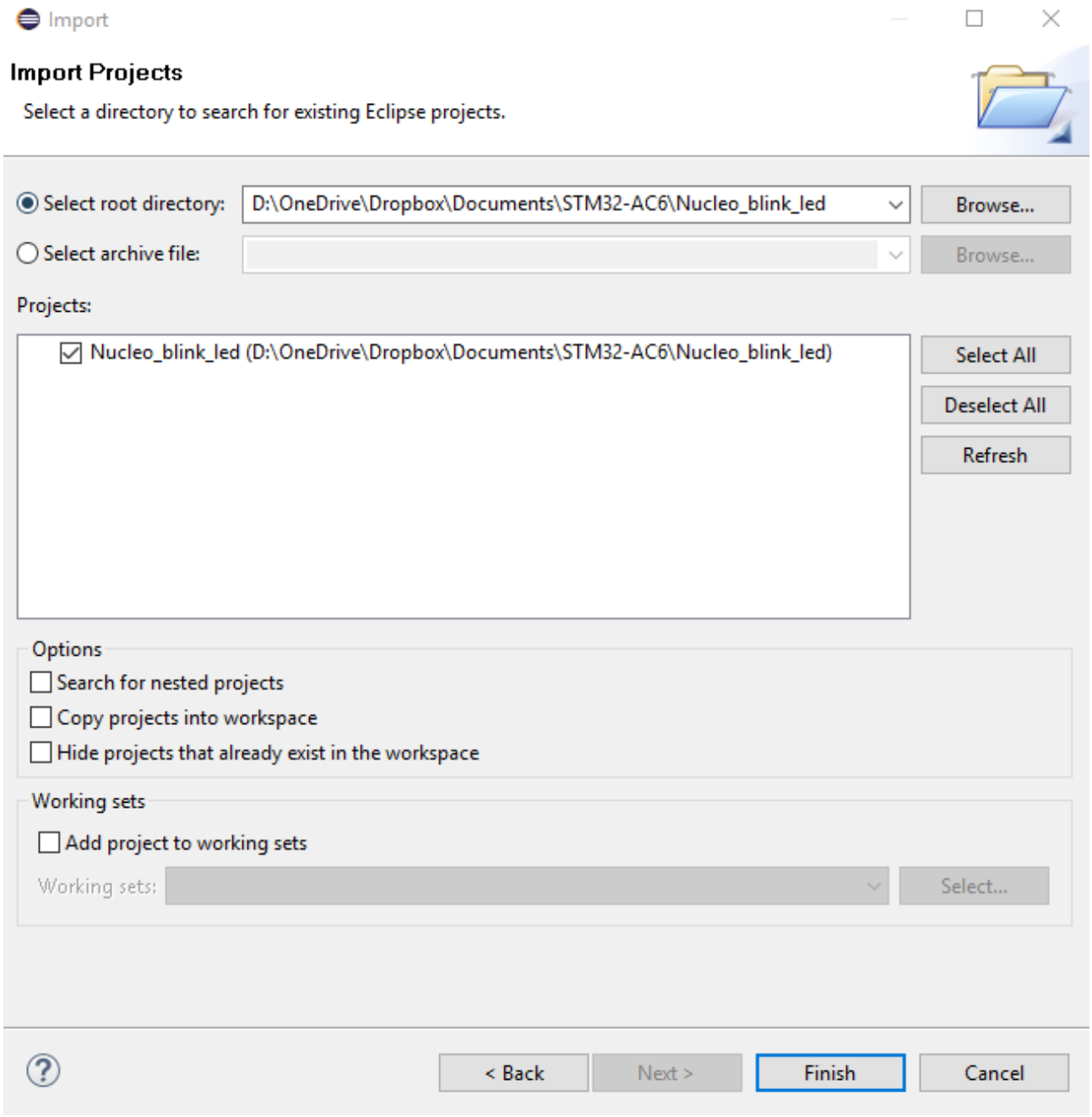
Une fois l'installation terminée revenir sur MBED et exporter le projet mbed_blinky vers AC6-Eclipse SW4STM32



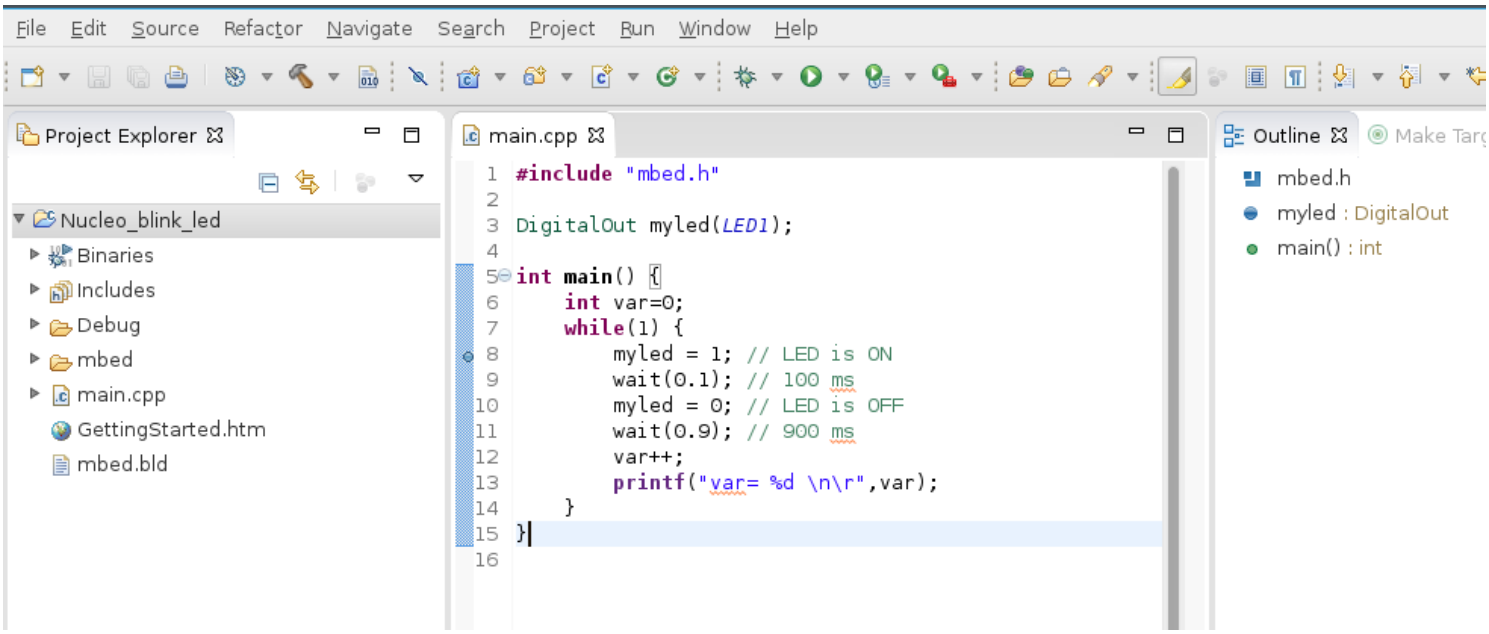
Dézipper le ficher d'export dans dans le dossier de votre choix (exemple: mbed_blinky_led)

Lancer "System Workbench for STM32"

File -> Import ->Existing Projetcts into Workspace -> Next
Selectionner le dossier contenant le projet exporté puis Finish



Double click sur main.cpp pour éditer le fichier du programme



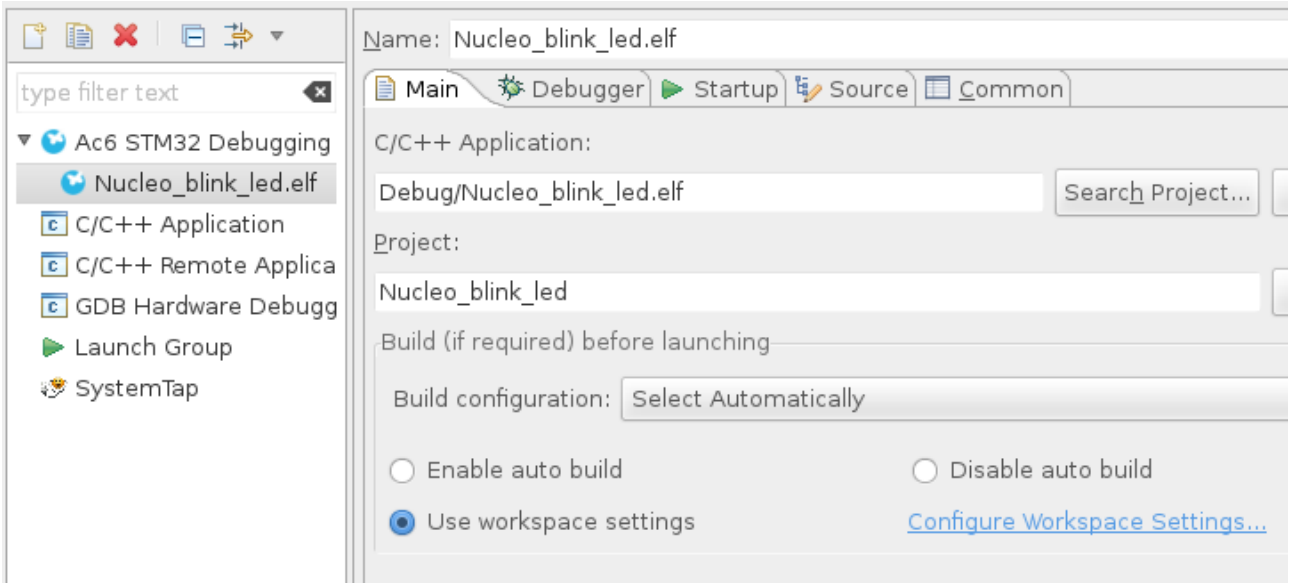
Il y a deux modes de compilation, debug (pour la mise au point) et release pour une version autonome, sans debug.
Pour activer le mode de compilation "debug", Project_>Build configuration -> Set active, valider "debug".

Pour tout recompiler , onglet "Project" -> Build All (ctrl-B), un fichier de code executable est crée (.elf)

Le projet est compilé... Il ne doit pas y avoir d'erreur indiquée dans la console en bas de l'EDI.
La prochaine compilation pourra se faire en cliquant sur le marteau (compilation partielle) ou sur le logo à droite, compilation complète.

Mode debug :
Run -> Run configurations

Créer une configuration Ac6 STM32 Debugging



Puis Run (en bas de la fenêtre), le prochain lancement pourra s'effectuer en cliquant sur le petit insecte...

Si une erreur apparait :

- Assurez vous que l'exportation depuis MBED a été faite pour la bonne cible et que la bibliothèque MBED était à jour.

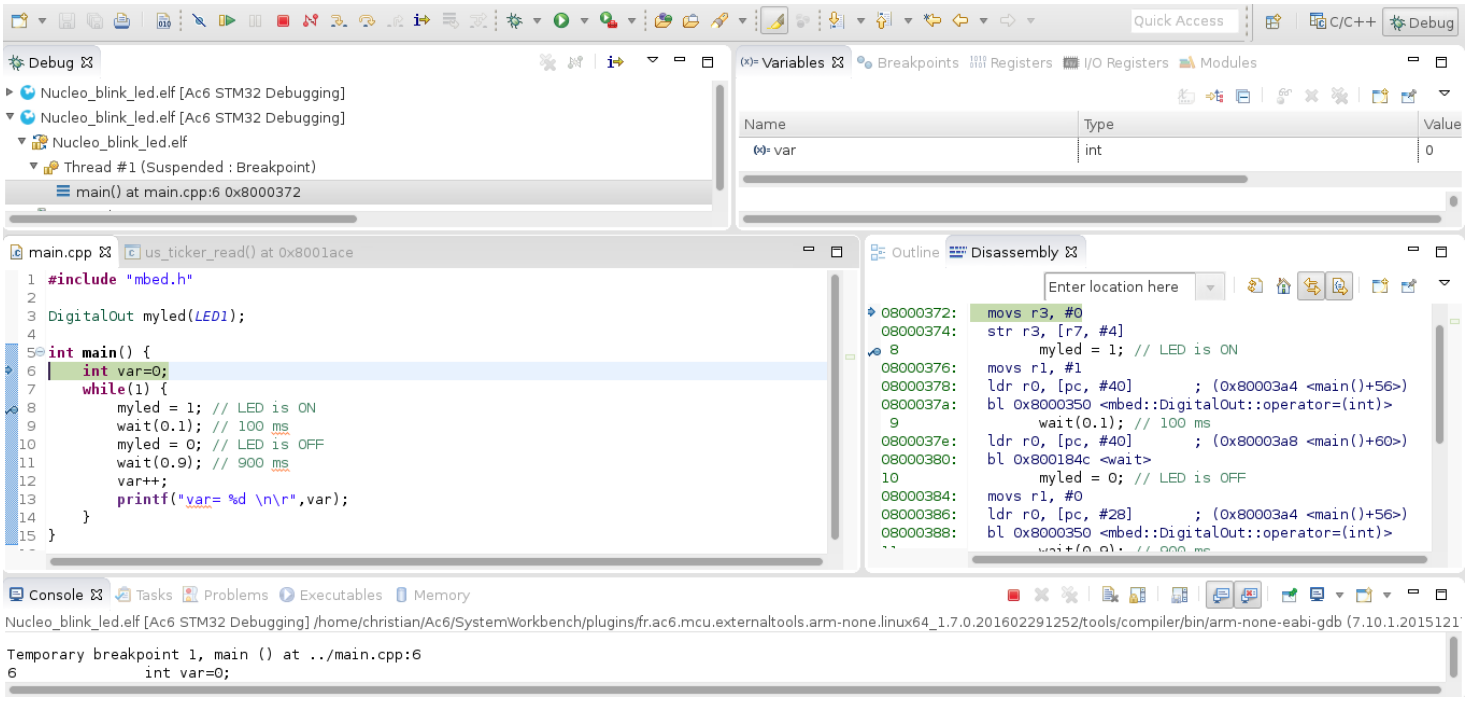
- Assurer vous que la carte cible est bien connectée.

Debugging:

Dans la perspective debug (en fait l'écran est réorganisé pour la mise au point), le programme est prêt à être lancé au début du"main".

Cliquer sur la flêche verte, le programme s'execute, cliquer à nouveau il s'arrête. Si le code source n'est pas connu (pour un arrêt dans une fonction MBED par exemple, il est possible de suivre le déroulement en langage assembleur.
Le debugger offre les fonctionnalités, point d'arrêt, pas à pas into et over, visulation des variables, registres du STM32, entrée sorties etc...

Pour placer un point d'arrêt, double clic dans la zone bleue à gauche le ligne C++ du point d'arrêt.



Avant de quitter le debugger, cliquer sur le carré rouge pour l'arrêter.

5 ARM-KEIL mise en oeuvre, STMF0

KEIL µVision est un EDI professionnel pour ARM/CORTEX. La version gratuite est limité à 32KO de code, ce qui sera suffisant pour découvrir les fonctionnalité du STM32 mais insufisant pour l'utilisation de certaines bibliothèques.
ST offre une licence gratuite µVision pour les microcontrôleurs ARM-CORTEXM0.
Le logiciel pour une carte Nucleo-F091RC peut être développé avec µVision.

Avec MBEB:
Effectuer simplement une exportation du projet MBED vers µVision4 (la mise à niveau vers la version 5 est automatique). Ouvrir le projet ainsi crée avec µVision.

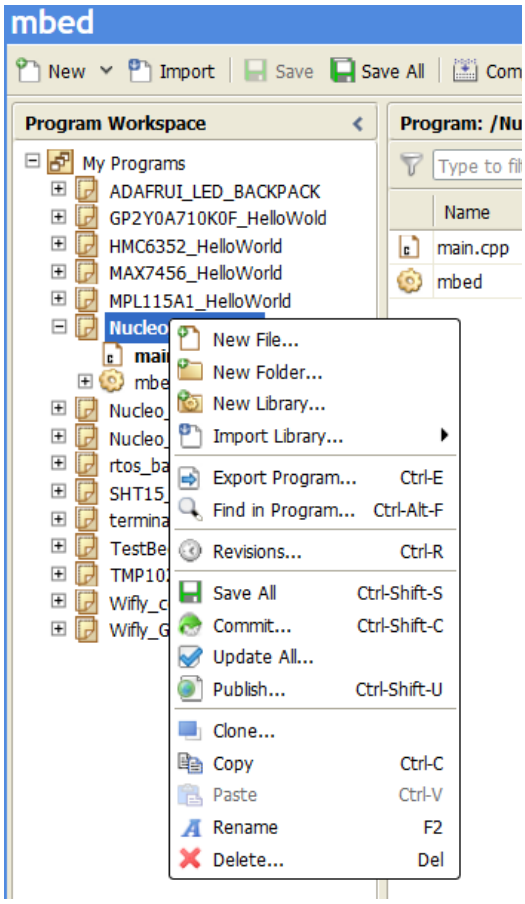
KEIL µVision est téléchargeable gratuitement (version 32KO)http://www2.keil.com/mdk5/uvision/
La licence gratuite pour le STM32 M0 : http://www2.keil.com/stmicroelectronics-stm32/mdk

5.1 Un exemple de debug avec µVision

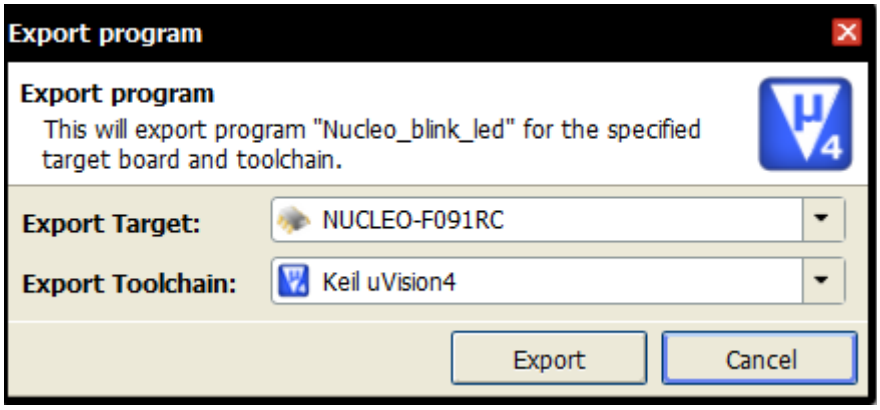
Modifier l'exemple MBED Nucleo_Blink_Led en ajoutant une variable comme suit :

```
#include "mbed.h"

DigitalOut myled(LED1);

int main()
{
    int mavar=0;
    while(1) {
        myled = 1; // LED is ON
        wait(0.1); // 100 ms
        myled = 0; // LED is OFF
        wait(0.5); // 0.5 sec
        mavar++;   // un simple compteur
    }
}
```

Essayer le programme sur une carte Nucleo comme indiqué précédement.
Pour exporter le programme et la bibliothèque MBED (version compilé)
Clic-droit sur le nom du projet, puis Export Program



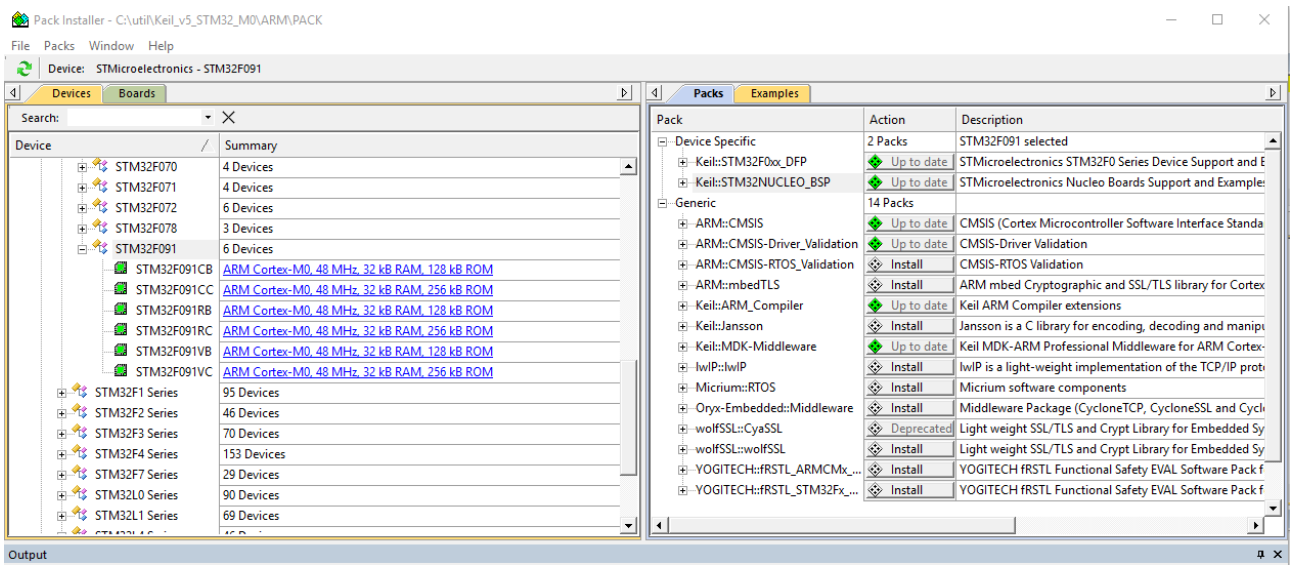
L'export se fait ici vers une carte NUCLEO-F091RC pour bénéficier de la version complète de µVision, vers une cible autre que le CORTEXM0 le code sera limité à 32KO (ce qui est largement siffisant ici).
La licence Keil est gratuite et sans limitation pour les µContrôleurs CORTEX M0



Selectionner Keil uVision, puis Export
Decompresser le fichier exporté : Nucleo_blink_led_uvision_nucleo_f091rc.zip

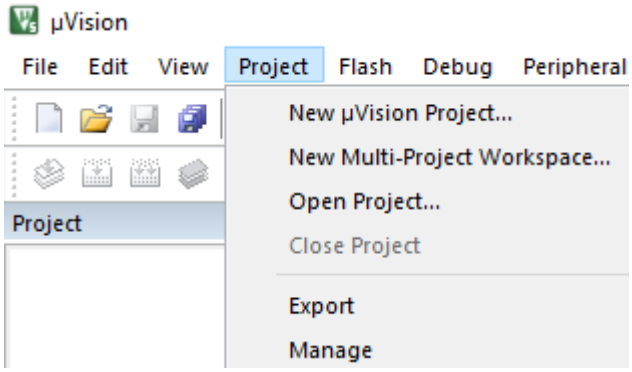
Ouvir KEIL µVision (V4 ou V5)

Installation de la configuration de la carte NUCLEO F091rc.
Cliquer sur Pack Intaller
Onglet Devices, Selectionner STM32F091, s'assurer que les deux premiers packs sont à jour

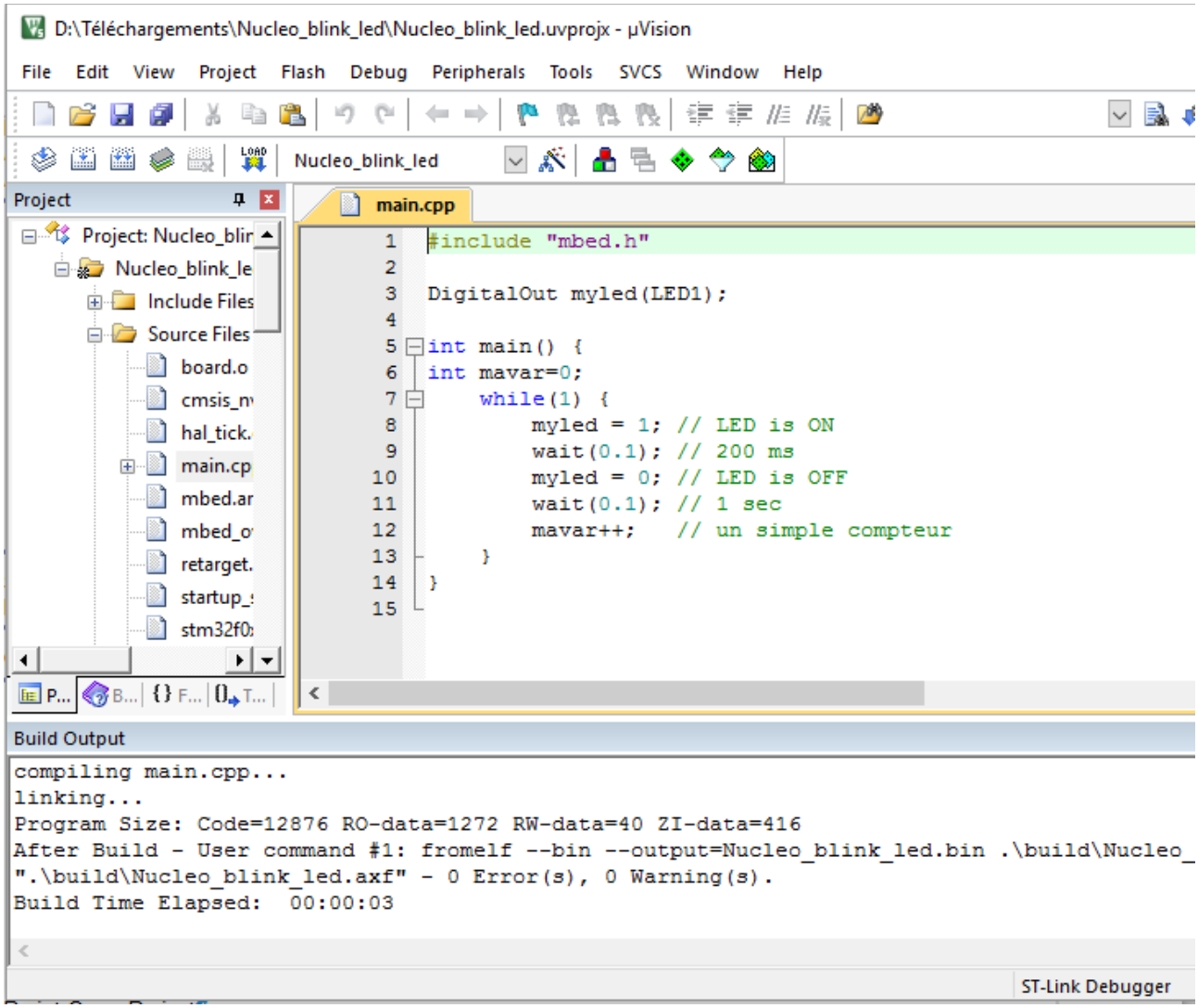


Il est également possible de télécharger des bibliothèques logicielles, d'exemples ou de fonctionnalités.

Chargement du projet exporté depuis MBED :
Projet-Open Project



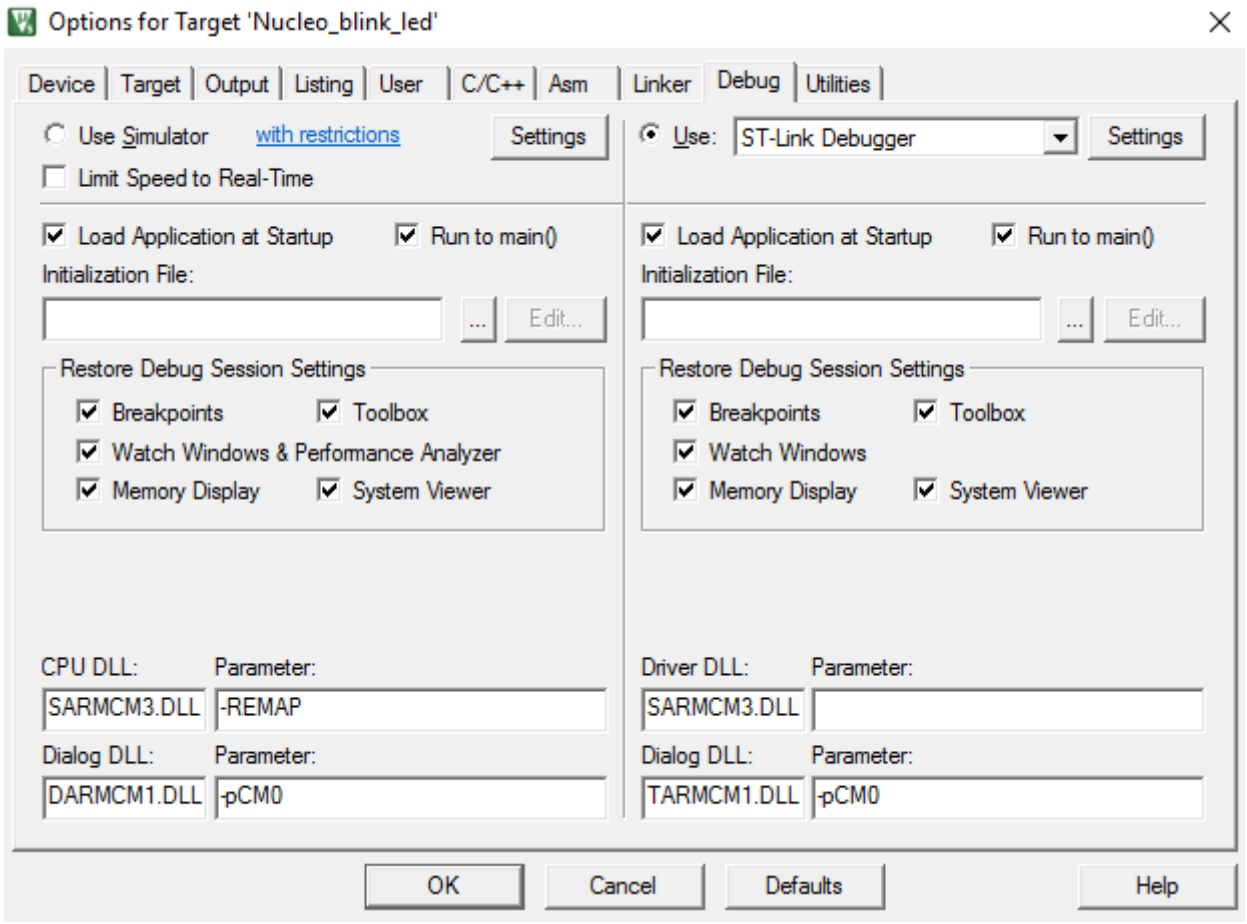
Naviguer jusqu'au fichier importé : Nucleo_blink_led.uvproj
Sous µVision 5 lors de l'ouverture, si il est proposé une migration, validez là.



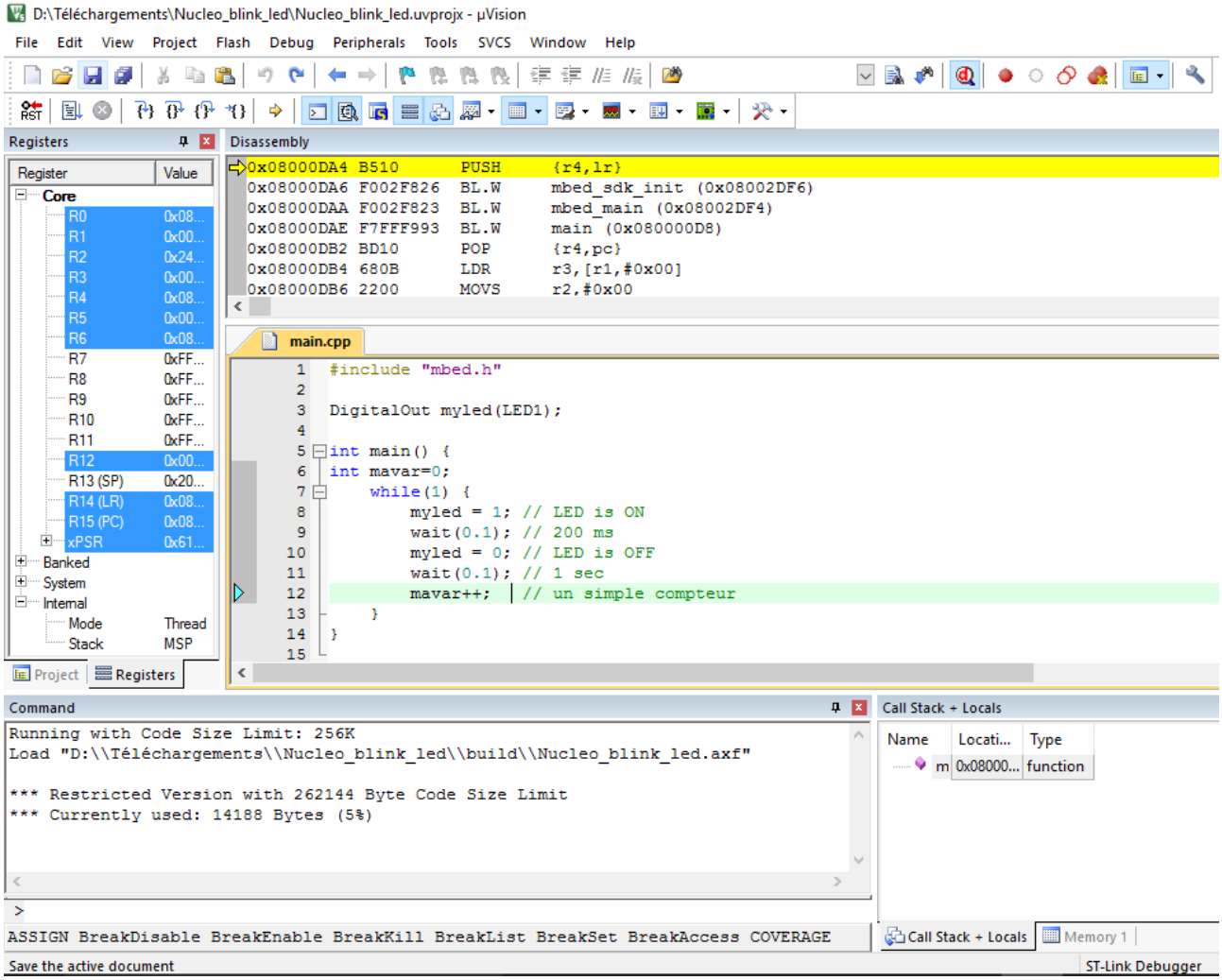
Pour compiler le programme F7 ou clic sur deuxième icone dernière ligne
Le résultat doit être **0 Errors, 0 Warning.**

Remarque: *un grand nombre de fichiers objet (.o) accompagne le projet, ce sont les fichiers précompilés de la bibliothèque MBED, seuls les fonctions utiles seront liée au programme. Les fichiers sources de la bibliothèque ne sont pas présent, la mise au point ne s'effectuera que sur le programme main.cpp.*
Il est tout à fait possible d'exporter avec le projet la bibliothèque sour MBED, le .zip est alors beaucoup plus important mais le degub est total.

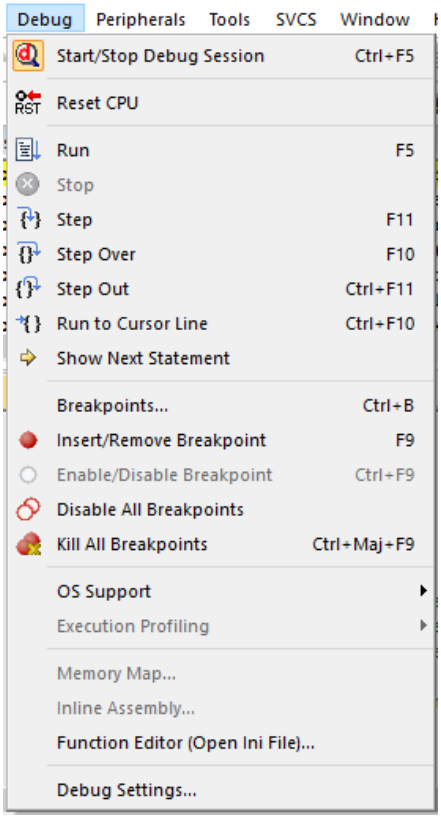
Si le debugger est ST-Link et s'il n'apparait pas en bas de la fenêtre µVision, cliquer sur la baguette magique ou ALT-F7. Dans l'onglet Debug selectioner dans la partie droire ST-Link Debugger



Quitter la fenêtre d'Edition/Compilation en cliquant sur l'icone loupe avec 'd' rouge ou ctrl-F5

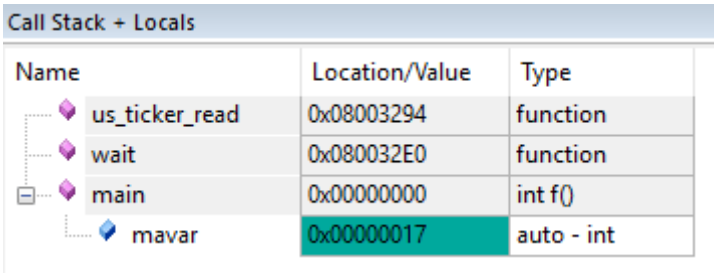


Lancer le programme Ctrl-F5
Pas à pas (Step) F11
Pas à pas par dessus les appel de sous programme F10
Etc...

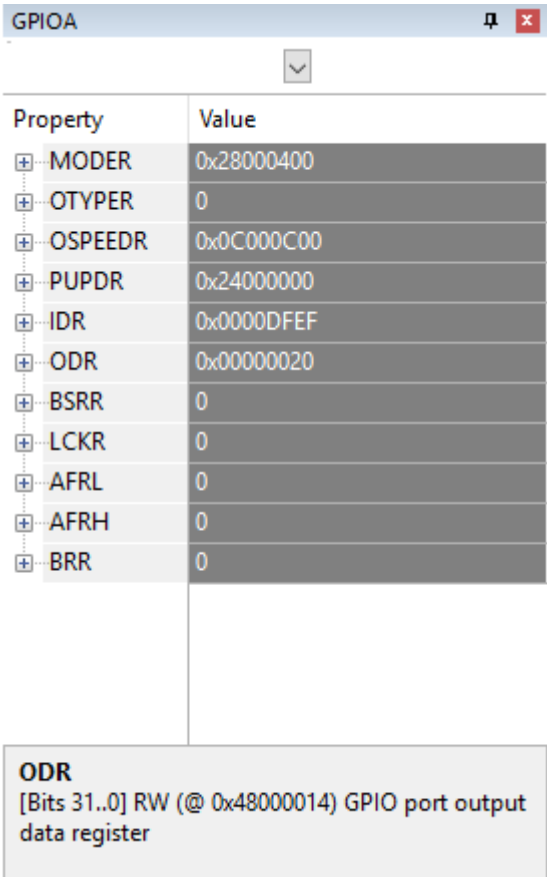


Pour placer un point d'arrêt : double-clic à gauche du numéro de ligne, de même pour le retirer

Les variables sont observables **lorsque le programme est arrêté**



Il est possible de visualiser les registres des périphériques
Peripherals – System Viewver, selectionner GPIOA sur lequel est la LED (bit ORD5)



La documentation de µVision pour ARM est disponible sur
https://armkeil.blob.core.windows.net/product/mdk5-getting-started.pdf
et
http://www.keil.com/support/man/docs/uv4/

6 STM-CUBE , configurateur de périphériques

STM CUBE est un utilitaire de ST facilitant la configuration de microcontrôleurs STM32.
Il génère une fichier de configuration ainsi que le fichier main.cpp.
L'appication doit ensuite être développé ave un EDI (exemple µVision).

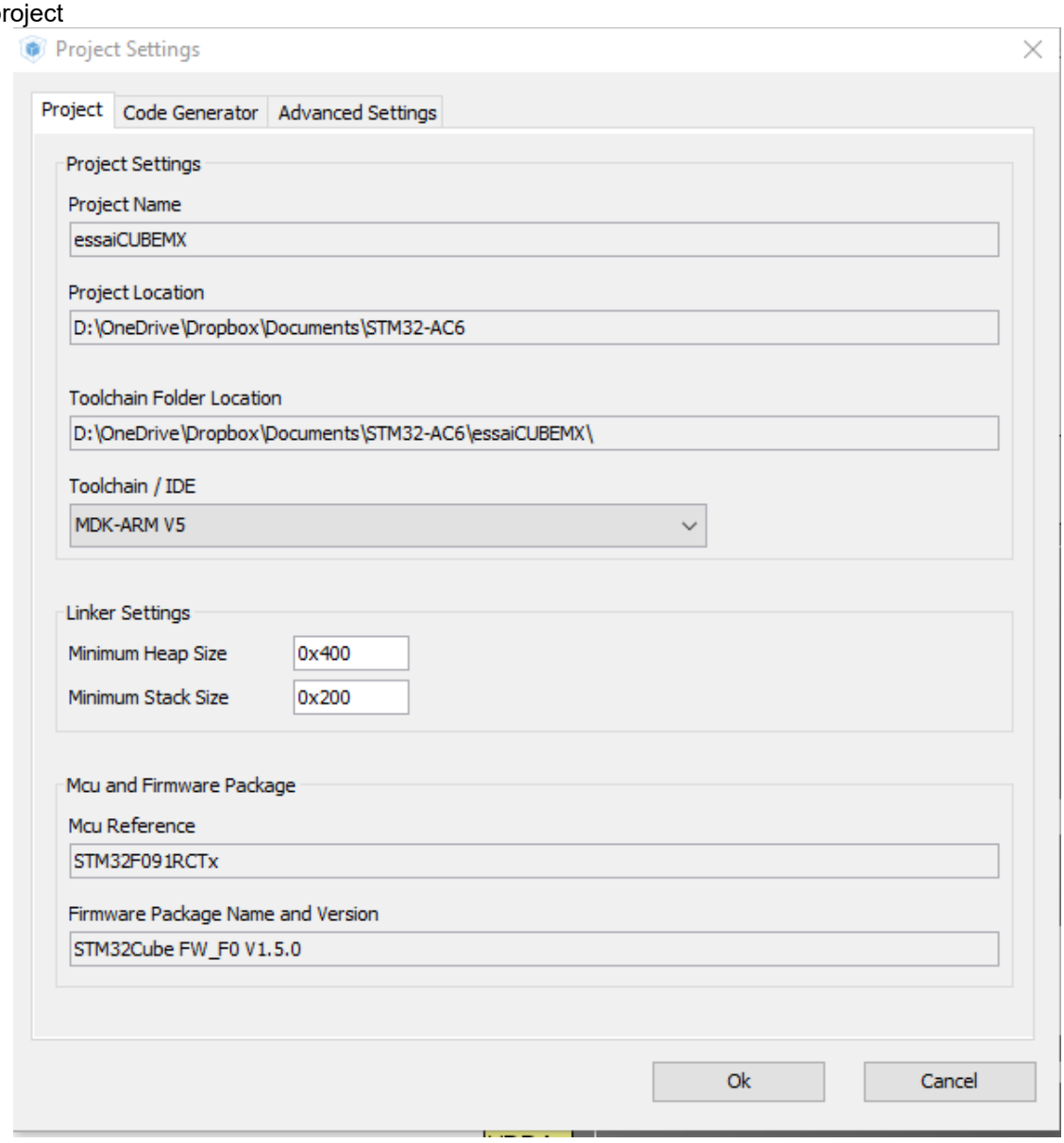
Télécharger CUBE-MX sur www.st.com et l'installer (sous Windows).
STM32CubeMX contients l'ensemble des configurateurs CUBE MX.

Le manuel de prise en main : http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/user_manual/DM00107720.pdf

Un exemple simple pour une carte NUCLEO F091RC

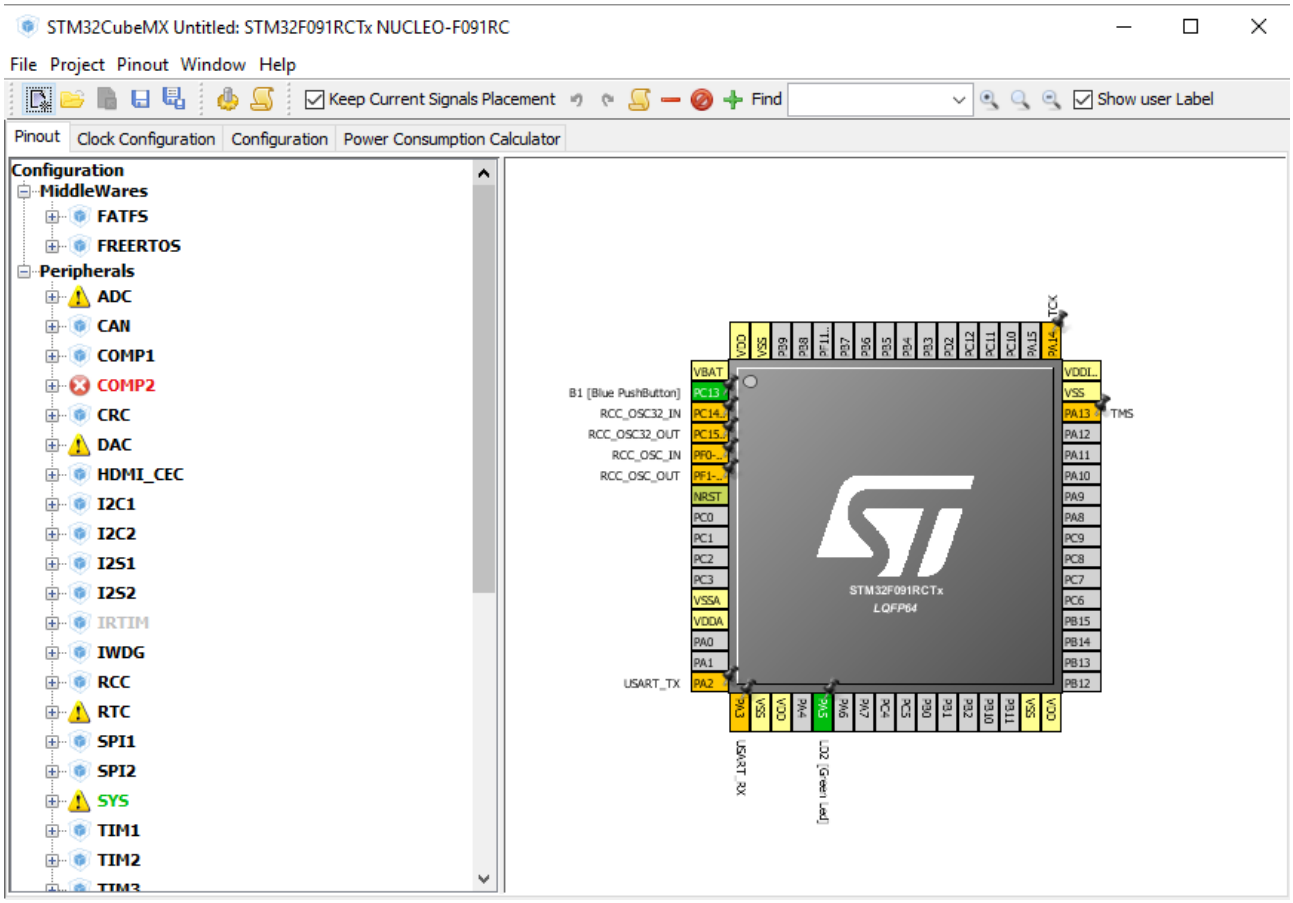
Lancer STM32CubeMX

New project

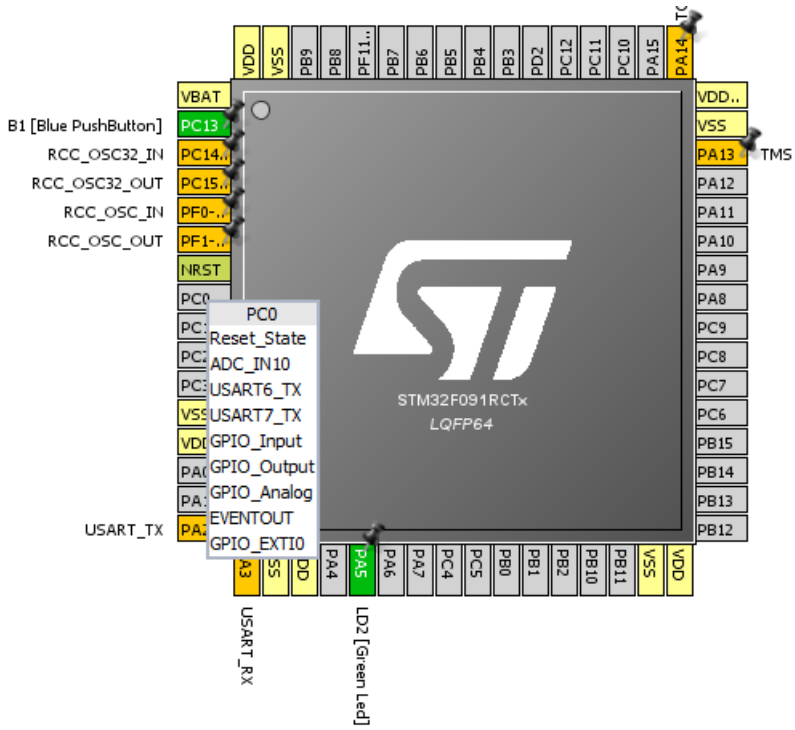


Ongelt "Board Selector", selectionner la carte NUCLEO F091RCSelectionner le debugger , par exemple

KEIL µVision.

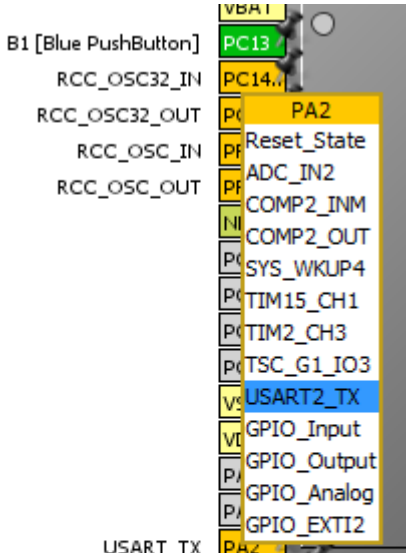


Certains périphériques ne sont pas utilisables comme PC13 et PC5 respectivement le bouton bleu et la LED rouge, on peut le vérifier en survolant la broche avec le pointeur de la souris.



Configuration de PC0, PC1, PC2, PC3 en GPIO sortie
Cliquer sur la broche correspondante puis GPIO_OUTPUT
Configurer ainsi les quatre GPIO

Vérifier que PA2 PA3 sont configurés pour les communications sur le port serie asynchrone virtuel (over USB).



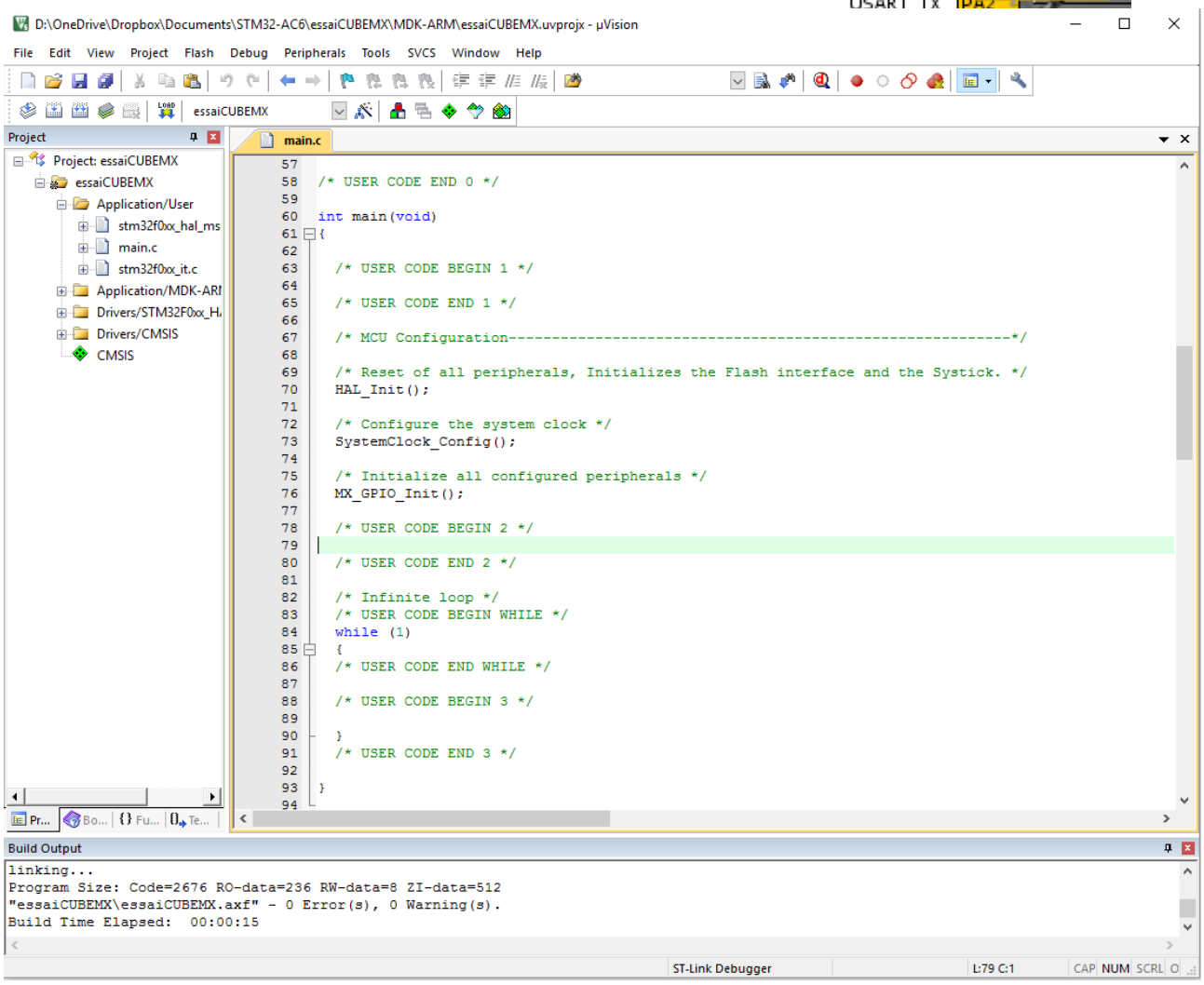
Il est également possible de configurer l'horloge (onglet clock configuration). Nous la laisserons dans la configuration proposée.

Pour créer le projet

Project – Generate Code
Entrer un nom de projet et un chemin
OK

Il est possible que des bibliothèques manques, CubeMX proposera de les télécharger

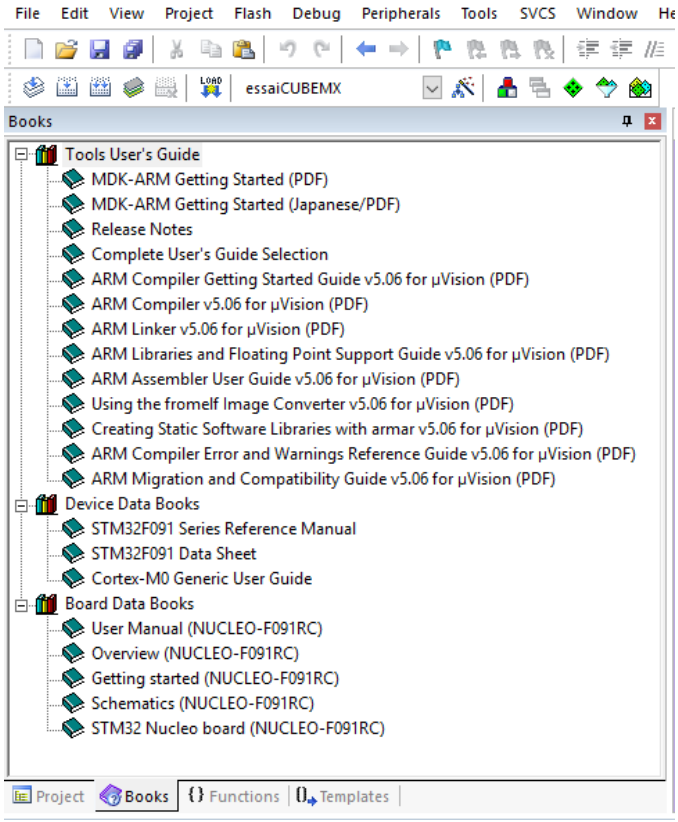
Après génération, CubeMXpropose d'ouvrir le dossier du projet ou directement de lancer le debbuger avec le projet.



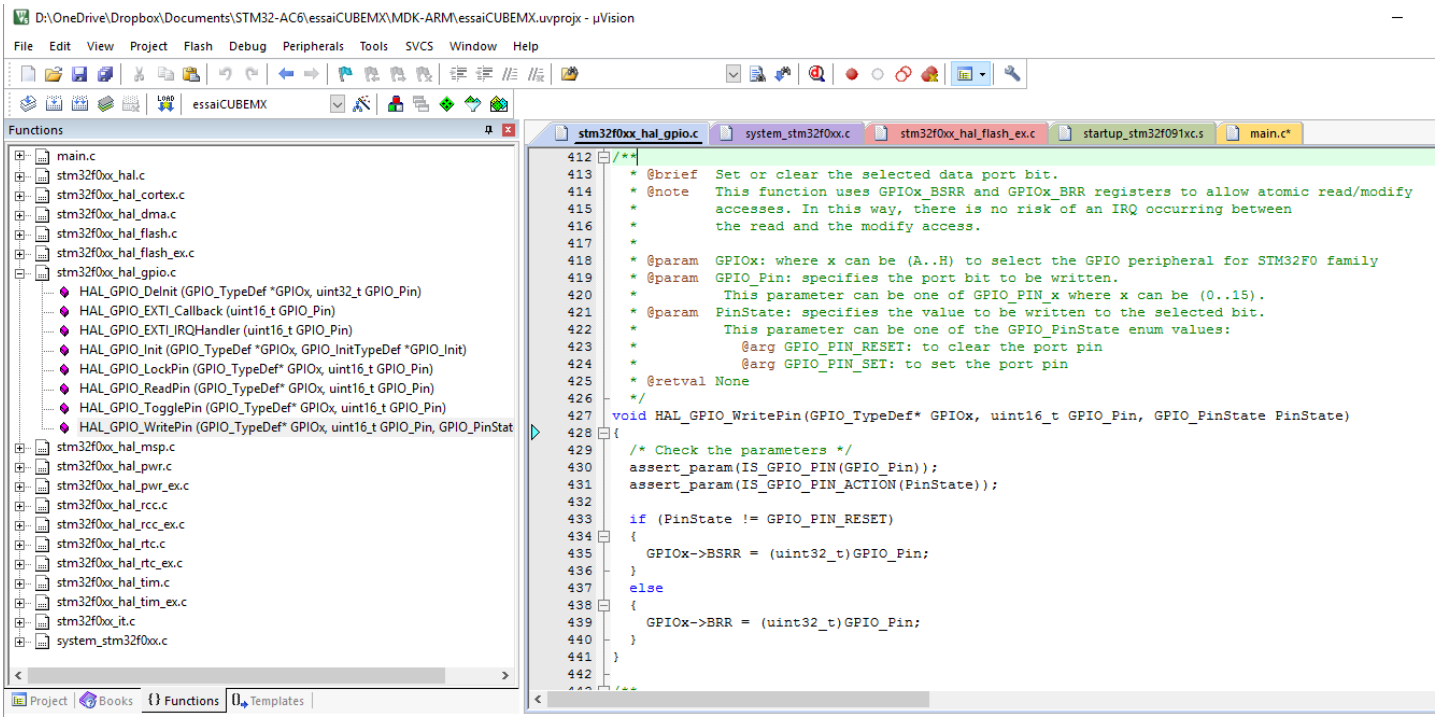
Le code utilisateur doit être écrit entre les balises /* USER CODE .....

L'onglet "Book" permet d'accéder à la documentation

La bibliothèque ST HAL est automatiquement inclue au projet, elle contient à peu prêt toutes les fonctions utiles pour programmer d'un STM32. L'onglet Function permet d'acceder aux fonctions de HAL



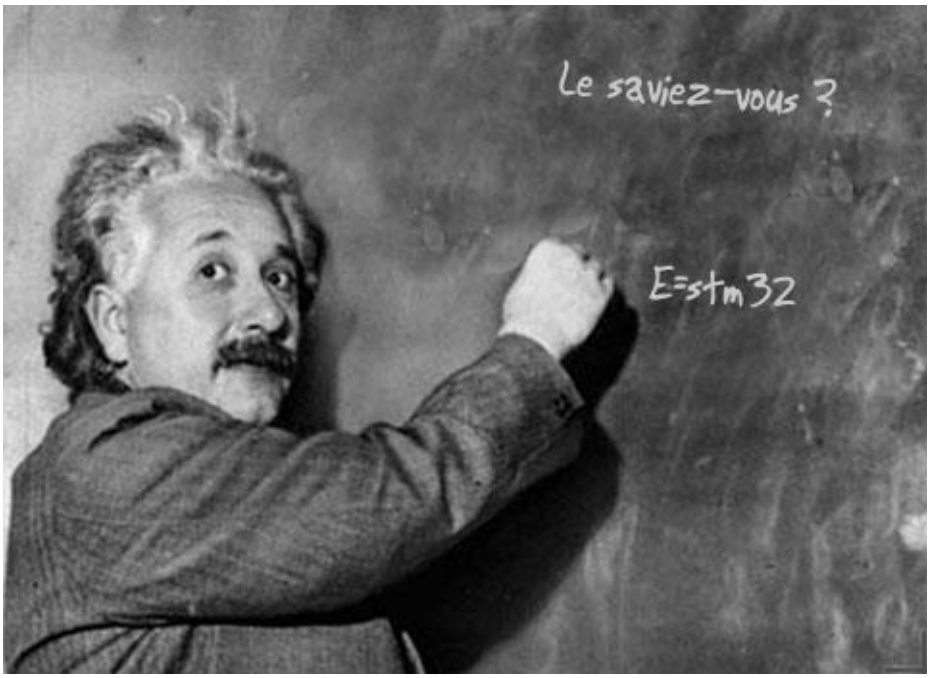
La documentation de HAL :
http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/DM00105879.pdf



Ici les fonctions de contrôle des GPIO

Pour faire simplement clignoter le LED rouge entrerdans main.c le code :

```
/* Infinite loop */
  /* USER CODE BEGIN WHILE */
  HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5,GPIO_PIN_RESET);
  while (1)
  {
  /* USER CODE END WHILE */
      int i;
      for(i=5;i>=0;i--)
      {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); //bascule PC5 (LED rouge)
        HAL_Delay(100); //delay 100ms
      }
      HAL_Delay(500); //delay 500ms
  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */
```



7 DOCUMENTATIONS EN LIGNE :

7.1 CARTES NUCLEO

Cartes NUCLEO – Présentation
http://www.st.com/content/ccc/resource/sales_and_marketing/promotional_material/flyer/d2/06/5c/20/25/28/42/ff/flstm32nucleo.pdf/files/flstm32nucleo.pdf/jcr:content/translations/en.flstm32nucleo.pdf
Getting started with STM32 Nucleo board software development tools
http://www.st.com/content/ccc/resource/technical/document/user_manual/1b/03/1b/b4/88/20/4e/cd/DM00105928.pdf/files/DM00105928.pdf/jcr:content/translations/en.DM00105928.pdf
STM32 NUCLEO64 User manual
http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf

7.2 DOCUMENTATION SUR LES MICROCONTRÖLEURS ARM-STM32

STM32M4 programming manual
http://www.st.com/content/ccc/resource/technical/document/programming_manual/6c/3a/cb/e7/e4/ea/44/9b/DM00046982.pdf/files/DM00046982.pdf/jcr:content/translations/en.DM00046982.pdf
STM32 CORTEX M0 programming manual
http://www.st.com/content/ccc/resource/technical/document/programming_manual/fc/90/c7/17/a1/44/43/89/DM00051352.pdf/files/DM00051352.pdf/jcr:content/translations/en.DM00051352.pdf
STM32 CORTEX F0 Reference maual
http://www.st.com/content/ccc/resource/technical/document/reference_manual/c2/f8/8a/f2/18/e6/43/96/DM00031936.pdf/files/DM00031936.pdf/jcr:content/translations/en.DM00031936.pdf
STM32F411RE Reference manual
http://www.st.com/content/ccc/resource/technical/document/reference_manual/9b/53/39/1c/f7/01/4a/79/DM00119316.pdf/files/DM00119316.pdf/jcr:content/translations/en.DM00119316.pdf

7.3 EDI

Prendre en main l'EDI Keil µVision pour ARM-STM32
http://www2.keil.com/docs/default-source/default-document-library/mdk5-getting-started.pdf?sfvrsn=2[NC,L]
Prendre en main l'EDI en ligne MBED
https://developer.mbed.org/explore/