



## STM32 Outils de développement

### Un aperçu 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.

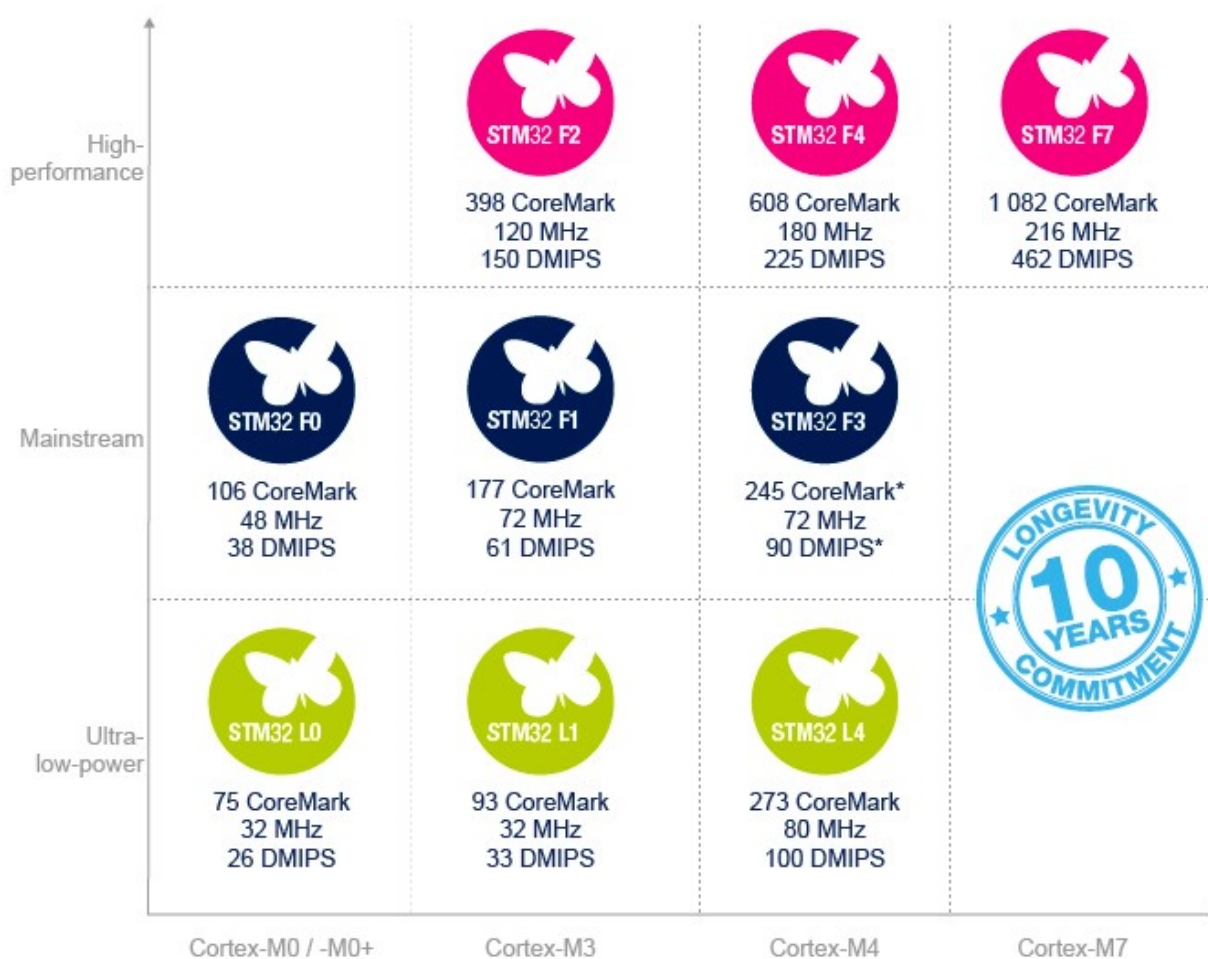
## 1 STM32 ARM-CORTEX

La technologie ARM-CORTEX est utilisée par de nombreux fondeurs dont ST, NXP, ATMEL, MAXIM, Silicon, WIZnet.

Le cœur 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 cœur ARM-CORTEX 32 bits depuis la très faible consommation jusqu'à une puissance de calcul de 462 DMIPS ( Dhrystone MIPS )

<http://www.st.com/web/en/catalog/mmc/FM141/SC1169?sc=stm32>

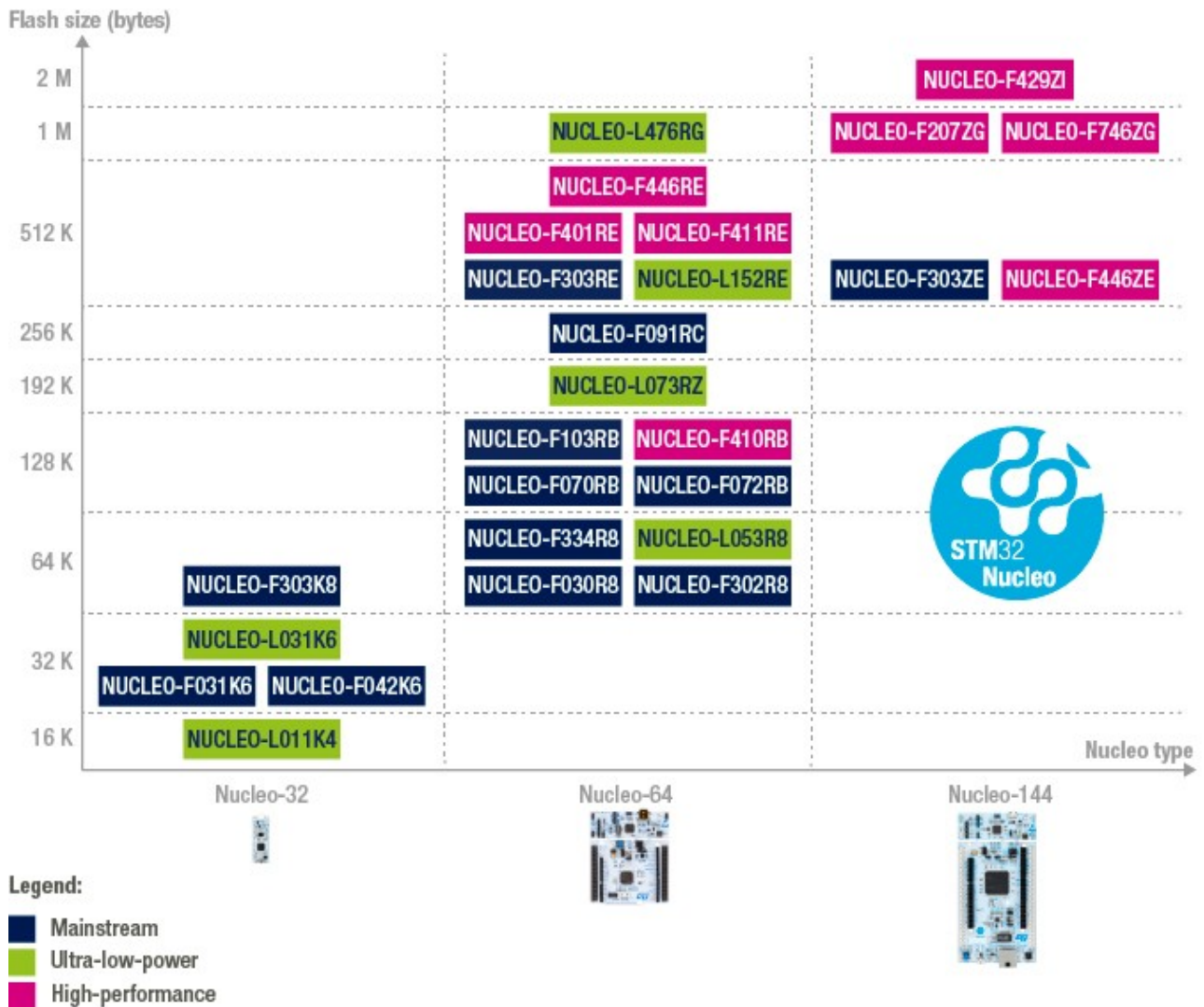


\* from CCM-SRAM



## 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 détachable, et une connectique de type Arduino. Ces cartes sont tout particulièrement adaptées à l'étude de faisabilité d'un projet.





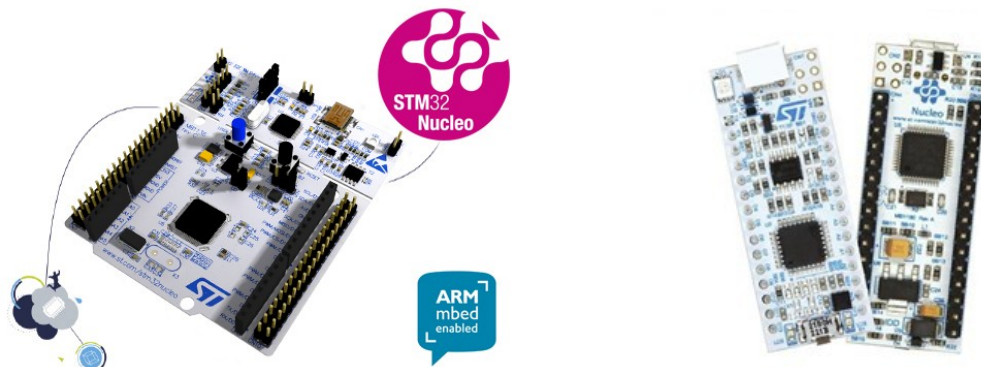
## STM32 Outils de développement

Ce document s'appuie sur les carte NUCLEO-F411RE et NUCLEO-F091RC

Le STM32F411 est un des microcontrôleurs 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](http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN1877/PF260049?s_searchtype=partnumber)

System		Control	
<b>Power supply</b> 1.2 V internal regulator POR/PDR/PVD/BOR  <b>Xtal oscillators</b> 32 kHz + 4 ~26 MHz  <b>Internal RC oscillators</b> 32 kHz + 16 MHz  <b>PLL</b>  <b>Clock control</b>  <b>RTC/AWU</b>  <b>2x watchdogs</b> (independent and window)  <b>36/50/81 I/Os</b>  <b>Cyclic Redundancy Check (CRC)</b>  <b>96-bit unique ID</b>  <b>Voltage scaling</b>	<b>ART Accelerator™</b>  <b>100 MHz ARM® Cortex®-M4 CPU</b>  <b>Floating Point Unit (FPU)</b>  <b>Nested Vector Interrupt Controller (NVIC)</b>  <b>JTAG/SW debug</b>  <b>Embedded Trace Macrocell (ETM)</b>  <b>Memory Protection Unit (MPU)</b>  <b>AHB-Lite bus matrix</b>  <b>APB bus</b>  <b>16-channel DMA with Batch Acquisition Mode (BAM)</b>	<b>512-Kbyte Flash memory</b>  <b>128-Kbyte SRAM</b>  <b>80-byte backup data</b>	<b>5x 16-bit timer</b>  <b>1x 16-bit motor control PWM synchronized AC timer</b>  <b>2x 32-bit timer</b>
		<b>Connectivity</b>  <b>3x PC</b>  <b>3x USART LIN, smartcard, IrDA, modem control</b>  <b>5x SPI or 5x I²S (2x I²S with full duplex)</b>  <b>SDIO</b>  <b>USB 2.0 OTG FS</b>	<b>Analog</b>  <b>1x 12-bit ADC 2.4 MSPS 16 channels / 0.41µs</b>  <b>Temperature sensor</b>



## 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€



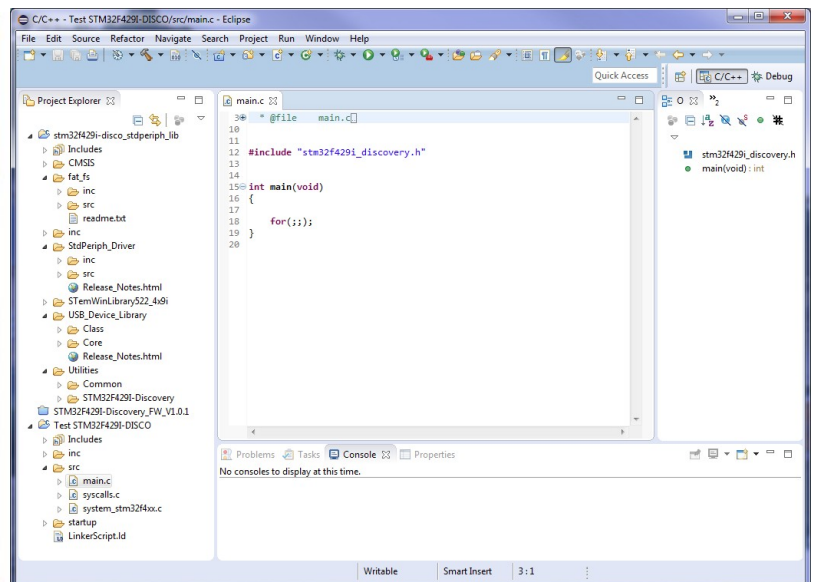
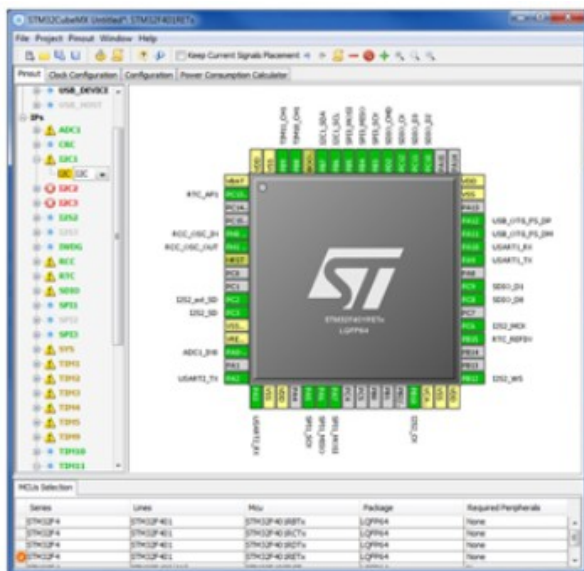
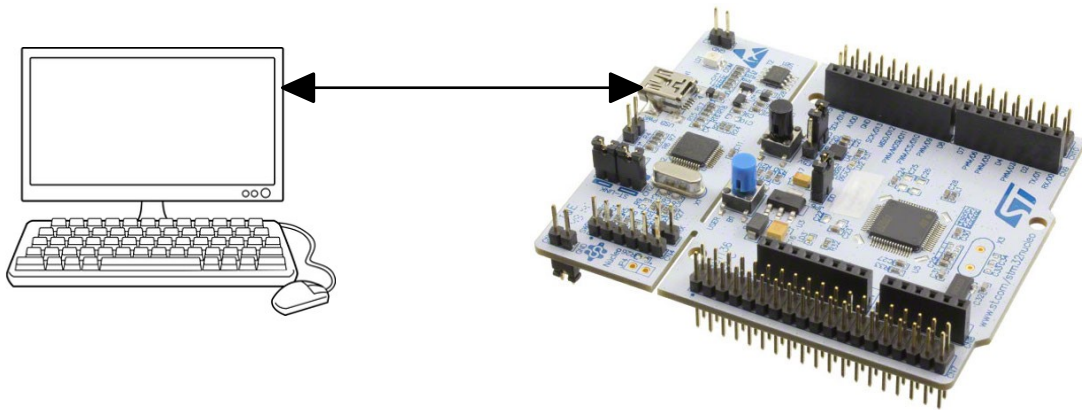
### 2.1 EDI intégré à l'ordinateur.

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 microcontrôleur. 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'eclipse fonctionne sur Windows, Linux et MACOS

**ARM-KEIL** propose un EDI sous Windows très performant mais gratuit uniquement pour les STM32F0.







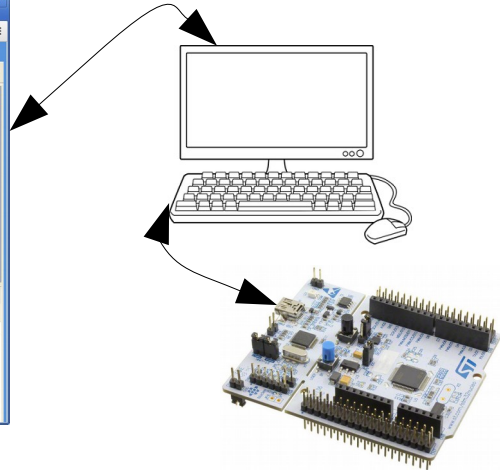
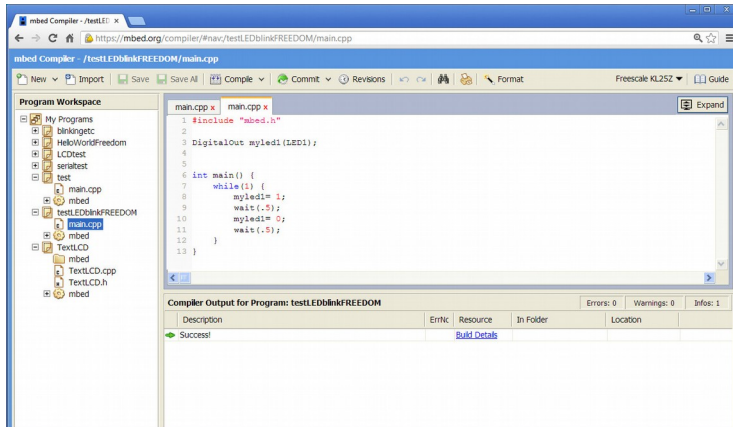
### 2.2 EDI en ligne (MBED)

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 nécessiter aucune installation dans l'ordinateur, une tablette et même un smartphone peuvent suffire. Une connexion internet est indispensable.

Par ailleurs 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/microcontrôleurs 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 développement sur plateforme ARM et les échanges entre développeurs.

Lors d'une première connexion vous devrez vous enregistrer pour accéder aux ressources MBED.

<https://www.mbed.com> puis cliquer "mbed classic developer site"

Vous êtes maintenant sur votre "mur" MBED.

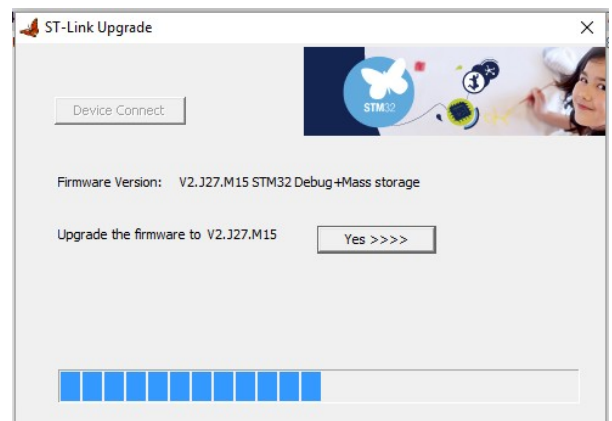
Nous allons développer un petit projet sur NUCLEO-STM32F411

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](#)

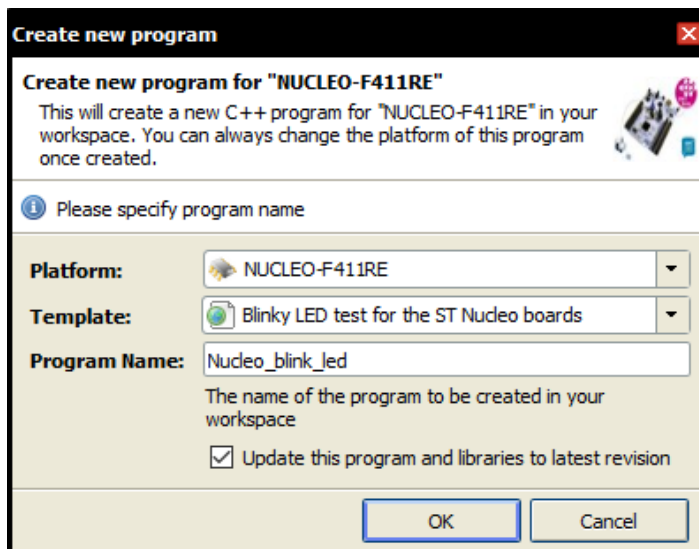
L'onglet "Cookbook" procure de très nombreux exemples

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 sélectionner 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

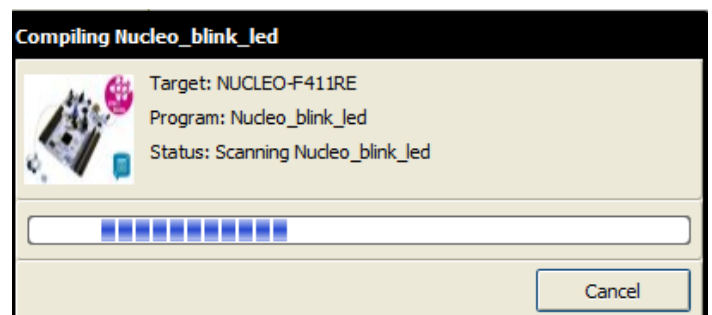
Après compilation, enregistrer le fichier  
Nucleo\_blink\_led\_NUCLEO\_F411RE.bin.

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

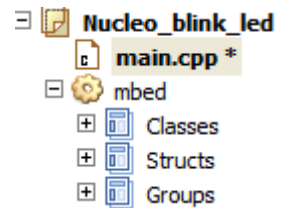
*Remarque:* Il est simple d'accéder à un autre port par son nom en remplaçant LED1 par PA\_10 on accède au bit 10 du PORTA;



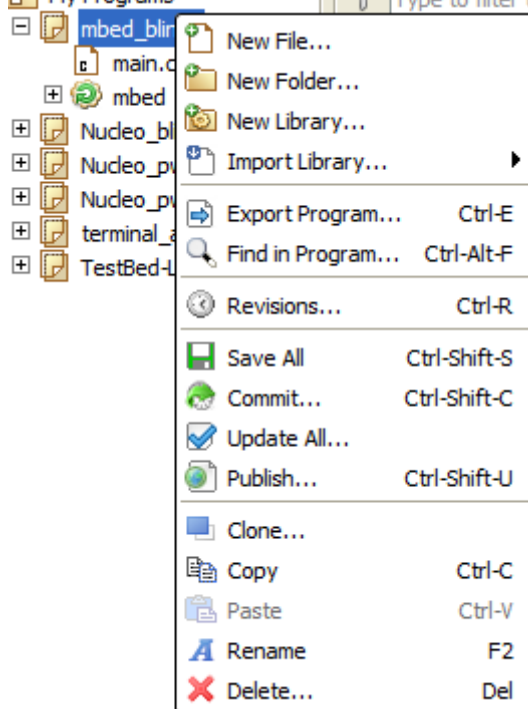
## STM32 Outils de développement



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**

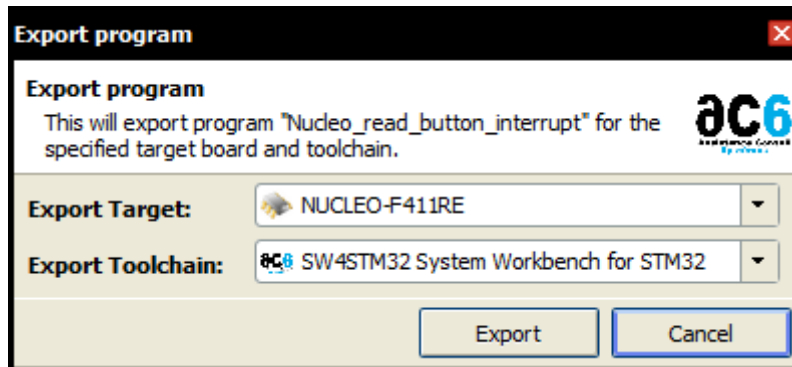


### 4 Eclipse AC6, mise en oeuvre

La société AC6 ( <http://www.ac6.fr/> ) propose des formations orientées développement logiciel. Elle a développé pour 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 indifférent 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 installer ".

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



Dézipper le fichier d'export dans dans le dossier de votre choix ( exemple: mbed\_blinky\_led)


Lancer "System Workbench for STM32"

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






## STM32 Outils de développement

 Import — □ ×

### Import Projects

Select a directory to search for existing Eclipse projects.



☒ Select root directory:

☐ Select archive file:

Projects:

<input checked="" type="checkbox"/> Nucleo_blink_led (D:\OneDrive\Dropbox\Documents\STM32-AC6\Nucleo_blink_led)	<input type="button" value="Select All"/>
	<input type="button" value="Deselect All"/>
	<input type="button" value="Refresh"/>

Options

☐ Search for nested projects


☐ Copy projects into workspace

☐ Hide projects that already exist in the workspace

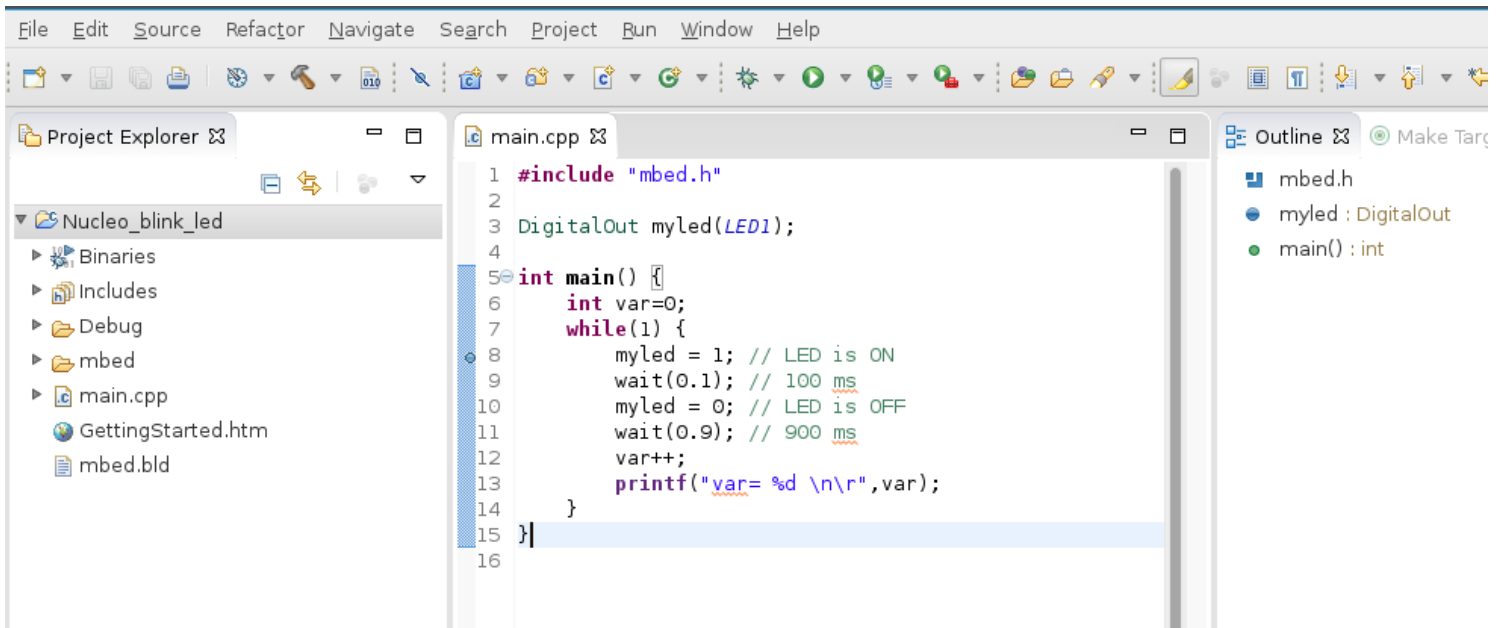
Working sets

☐ Add project to working sets

Working sets:



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)

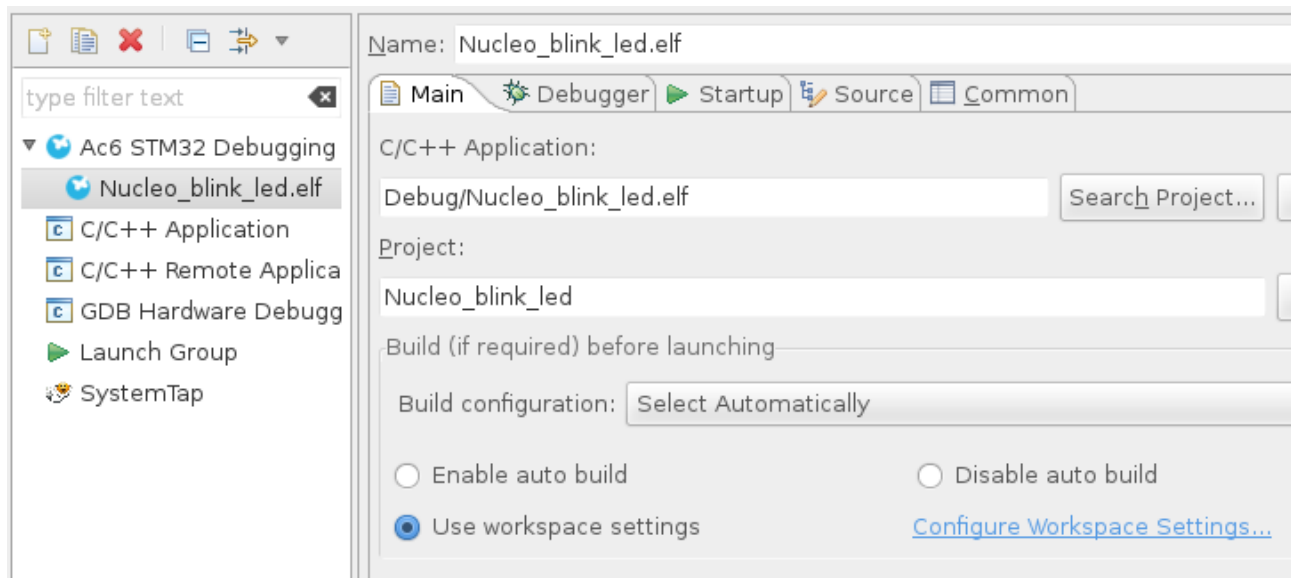
Le projet est compilé... Il ne doit pas y avoir d'erreur indiquée dans la console en bas de l'EDI.

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...



## STM32 Outils de développement

### Si une erreur apparaît :

- 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.

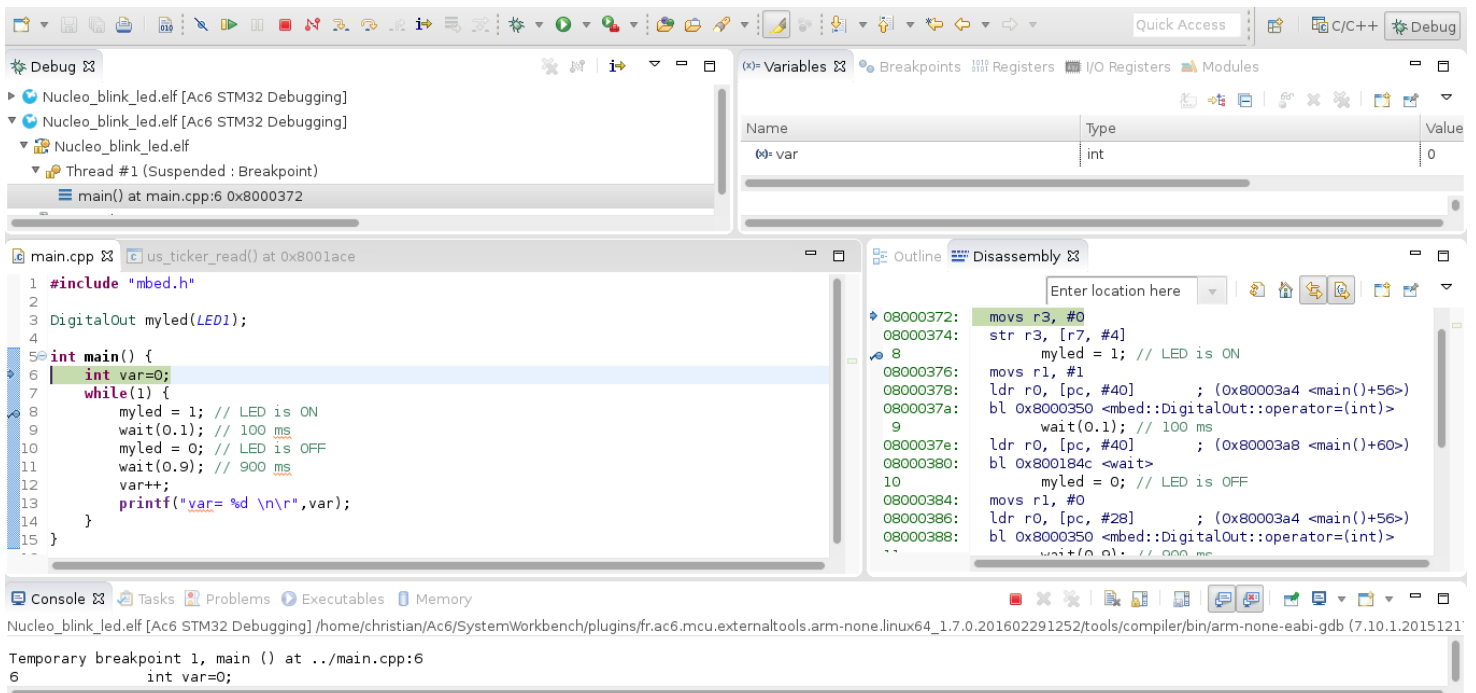
### Debugging:

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 flèche verte, le programme s'exécute, 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, visualisation 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  $\mu$ Vision est un EDI professionnel pour ARM/CORTEX. La version gratuite est limitée à 32KO de code, ce qui sera suffisant pour découvrir les fonctionnalités du STM32 mais insuffisant pour l'utilisation de certaines bibliothèques.

ST offre une licence gratuite  $\mu$ Vision pour les microcontrôleurs ARM-CORTEXM0.

Le logiciel pour une carte Nucleo-F091RC peut être développé avec  $\mu$ Vision.

### Avec MBED:

Effectuer simplement une exportation du projet MBED vers  $\mu$ Vision4 (la mise à niveau vers la version 5 est automatique). Ouvrir le projet ainsi créé avec  $\mu$ Vision.

KEIL  $\mu$ Vision 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 $\mu$ Vision

Modifier l'exemple MBED Nucleo\_Blink\_Led en ajoutant une variable comme suit :

```
#include "mbed.h"

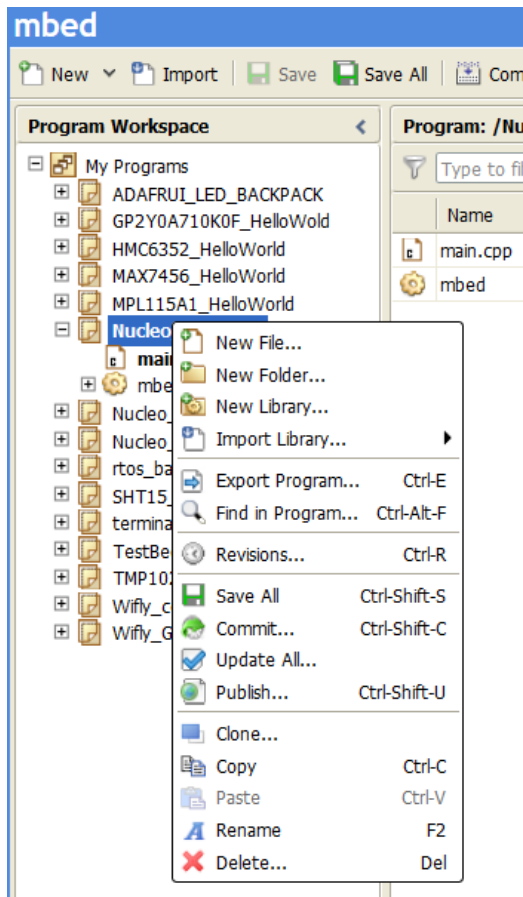
DigitalOut myled(LED1);

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
    }
}
```

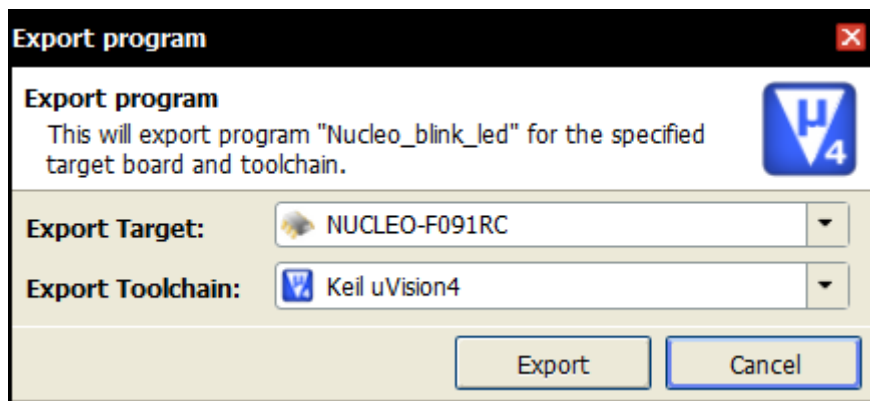
Essayer le programme sur une carte Nucleo comme indiqué précédemment.

Pour exporter le programme et la bibliothèque MBED (version compilée)

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  $\mu$ Vision, vers une cible autre que le CORTEXM0 le code sera limité à 32KO (ce qui est largement suffisant ici ).  
*La licence Keil est gratuite et sans limitation pour les  $\mu$ Contrôleurs CORTEX M0*



Selectionner Keil uVision, puis Export  
 Décompresser le fichier exporté : Nucleo\_blink\_led\_uvision\_nucleo\_f091rc.zip

Ouvrir KEIL  $\mu$ Vision (V4 ou V5)

### Installation de la configuration de la carte NUCLEO F091rc.

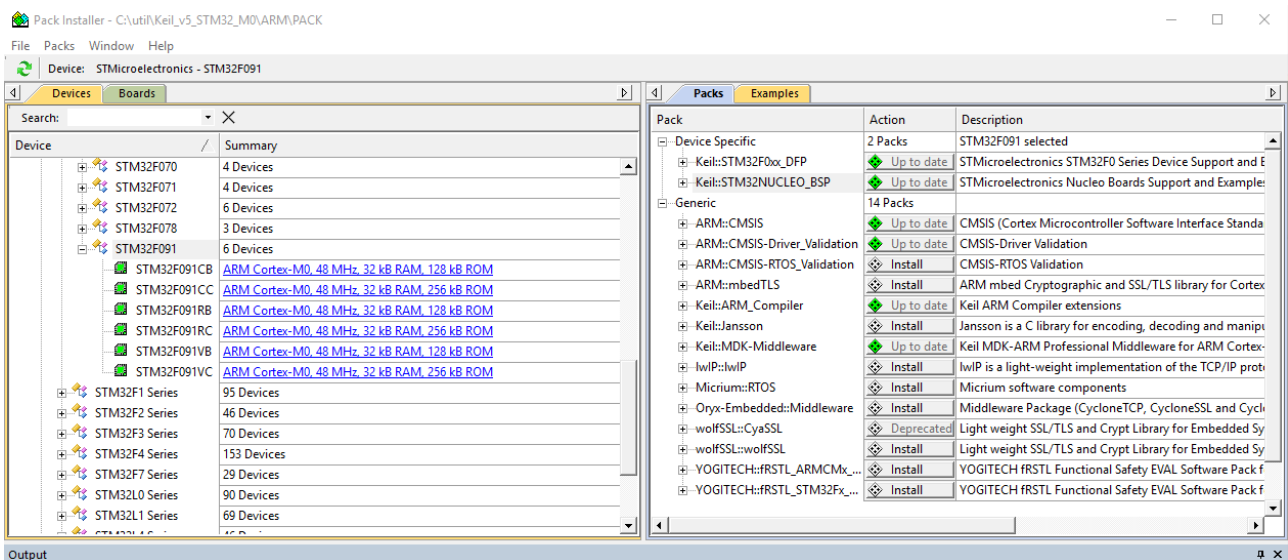
Cliquer sur Pack Installer

Onglet Devices, Selectionner STM32F091, s'assurer que les deux premiers packs sont à jour



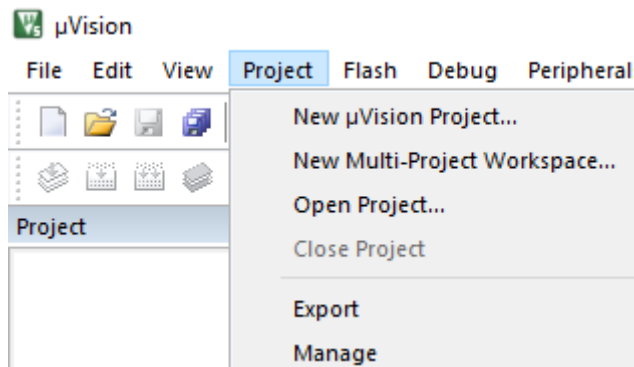


## STM32 Outils de développement

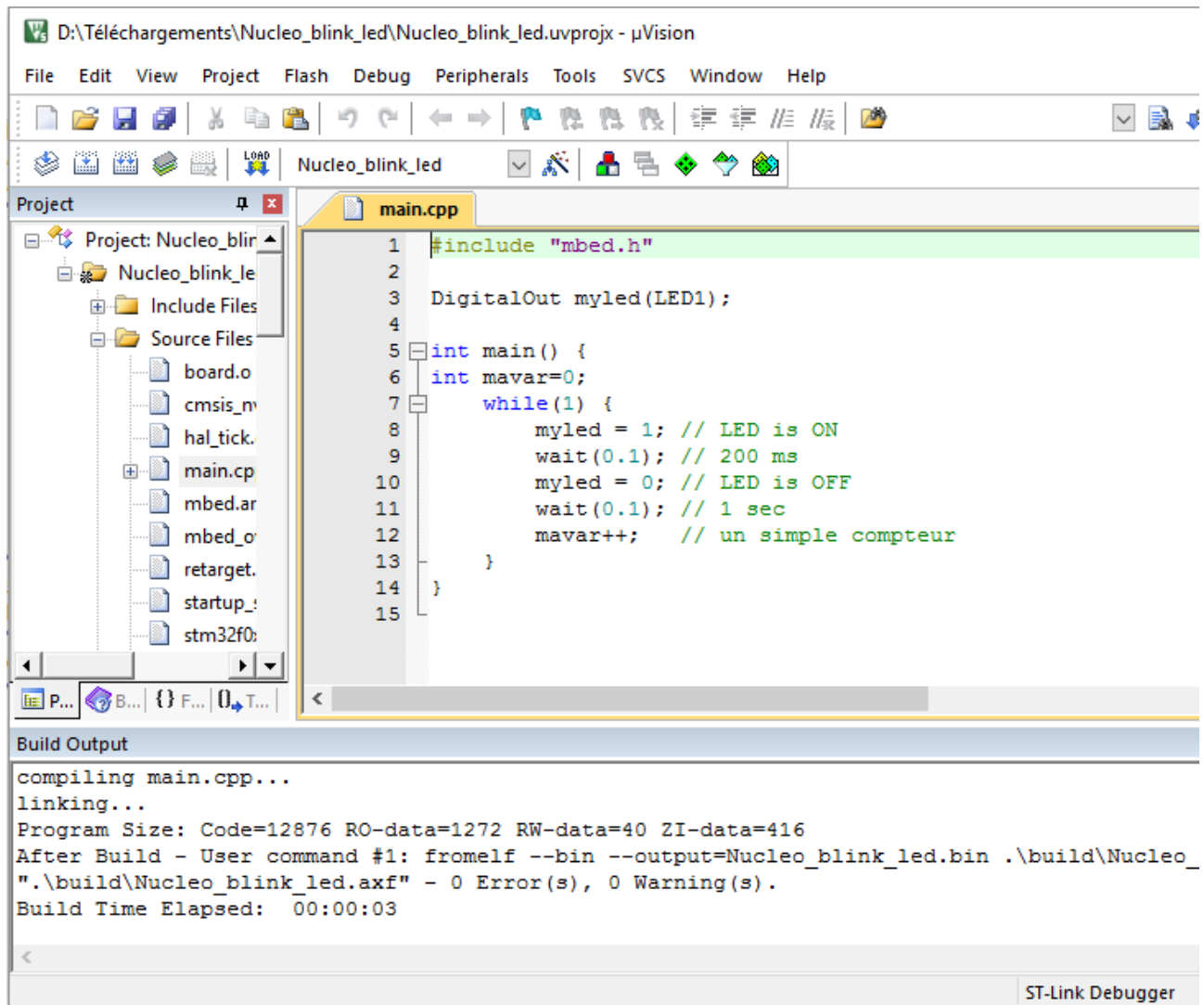


Il est également possible de télécharger des bibliothèques logicielles, d'exemples ou de fonctionnalités.

### 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 icône 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.

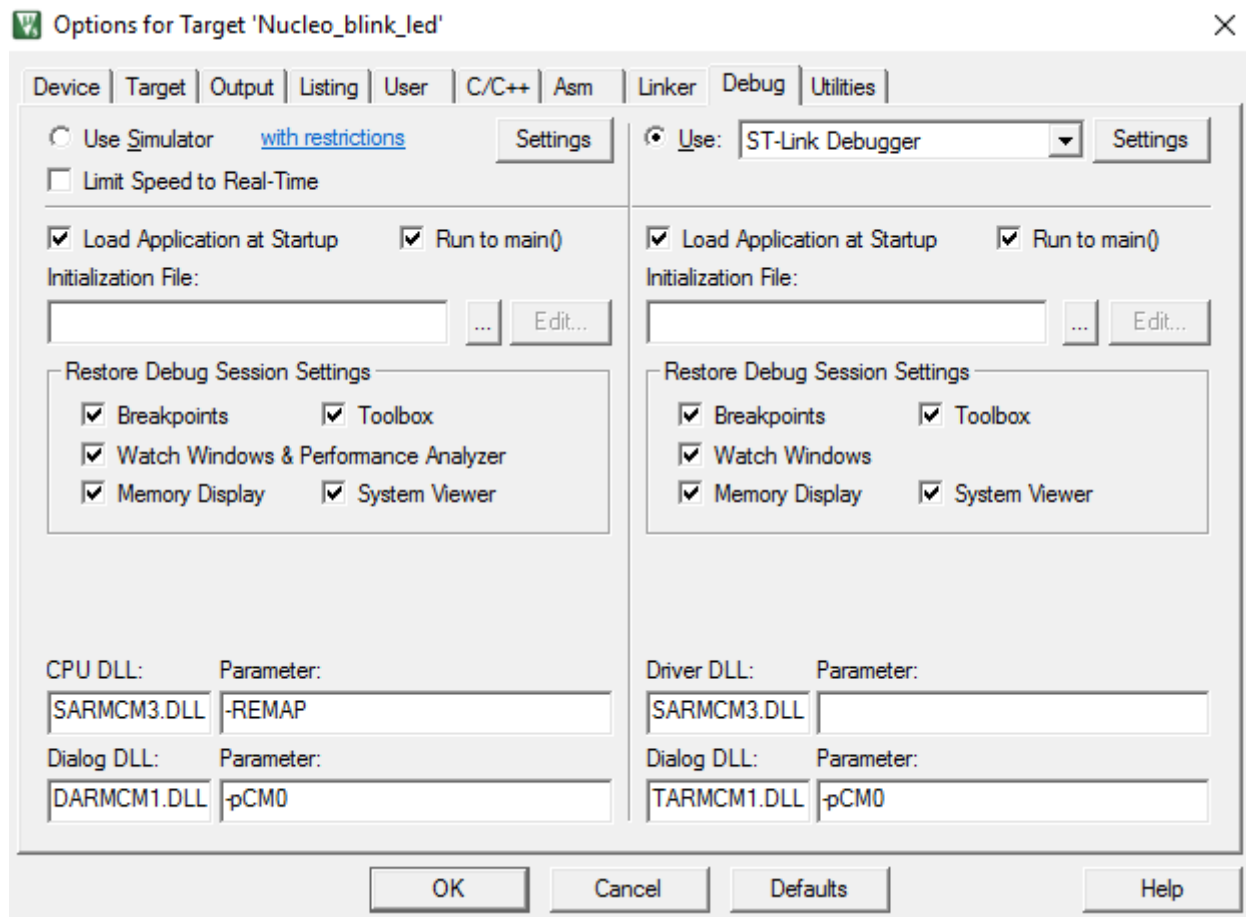
Il est tout à fait possible d'exporter avec le projet la bibliothèque sour MBED, le .zip est alors beaucoup plus important mais le debug est total.

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





## STM32 Outils de développement



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



## STM32 Outils de développement

D:\Téléchargements\Nucleo\_blink\_led\Nucleo\_blink\_led.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers Disassembly

Register	Value
R0	0x08...
R1	0x00...
R2	0x24...
R3	0x00...
R4	0x08...
R5	0x00...
R6	0x08...
R7	0xFF...
R8	0xFF...
R9	0xFF...
R10	0xFF...
R11	0xFF...
R12	0x00...
R13 (SP)	0x20...
R14 (LR)	0x08...
R15 (PC)	0x08...
xPSR	0x61...

Disassembly

```

0x08000DA4 B510 PUSH {r4,lr}
0x08000DA6 F002F826 BL.W mbed_sdk_init (0x08002DF6)
0x08000DAA F002F823 BL.W mbed_main (0x08002DF4)
0x08000DAE F7FFF993 BL.W main (0x080000D8)
0x08000DB2 BD10 POP {r4,pc}
0x08000DB4 680B LDR r3,[r1,#0x00]
0x08000DB6 2200 MOVS r2,#0x00
  
```

main.cpp

```

1 #include "mbed.h"
2
3 DigitalOut myled(LED1);
4
5 int main() {
6     int mavar=0;
7     while(1) {
8         myled = 1; // LED is ON
9         wait(0.1); // 200 ms
10        myled = 0; // LED is OFF
11        wait(0.1); // 1 sec
12        mavar++; // un simple compteur
13    }
14 }
15
  
```

Command

```

Running with Code Size Limit: 256K
Load "D:\\Téléchargements\\Nucleo_blink_led\\build\\Nucleo_blink_led.axf"

*** Restricted Version with 262144 Byte Code Size Limit
*** Currently used: 14188 Bytes (5%)
  
```

Call Stack + Locals

Name	Locati...	Type
m	0x08000...	function

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE

Save the active document

ST-Link Debugger

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

Debug Peripherals Tools SVCS Window

Start/Stop Debug Session Ctrl+F5

Reset CPU

Run F5

Stop

Step F11

Step Over F10

Step Out Ctrl+F11

Run to Cursor Line Ctrl+F10

Show Next Statement

Breakpoints... Ctrl+B

Insert/Remove Breakpoint F9

Enable/Disable Breakpoint Ctrl+F9

Disable All Breakpoints

Kill All Breakpoints Ctrl+Maj+F9

OS Support

Execution Profiling

Memory Map...

Inline Assembly...

Function Editor (Open Ini File)...

Debug Settings...



## STM32 Outils de développement

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**

Call Stack + Locals		
Name	Location/Value	Type
us_ticker_read	0x08003294	function
wait	0x080032E0	function
main	0x00000000	int f()
mavar	0x00000017	auto - int

Il est possible de visualiser les registres des périphériques

Peripherals – System Viewver, sélectionner GPIOA sur lequel est la LED (bit ORD5)

GPIOA	
▼	
Property	Value
+ MODER	0x28000400
+ OTYPER	0
+ OSPEEDR	0x0C000C00
+ PUPDR	0x24000000
+ IDR	0x0000DFEF
+ ODR	0x00000020
+ BSRR	0
+ LCKR	0
+ AFRL	0
+ AFRH	0
+ BRR	0

**ODR**  
[Bits 31..0] RW (@ 0x48000014) GPIO port output data register

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 un fichier de configuration ainsi que le fichier main.cpp. L'application doit ensuite être développée avec un EDI (exemple µVision).

Télécharger CUBE-MX sur [www.st.com](http://www.st.com) et l'installer (sous Windows).  
*STM32CubeMX contient 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](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

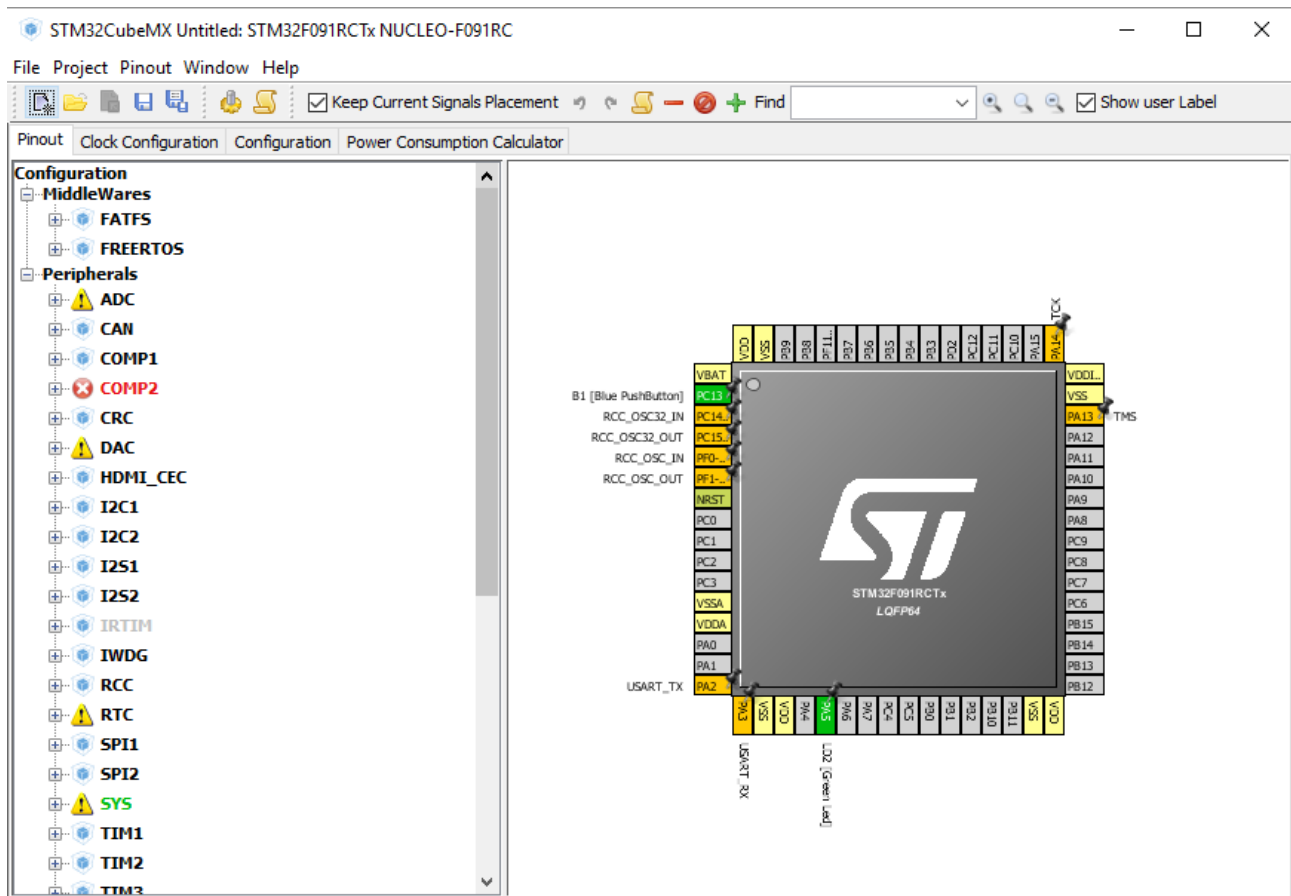
Lancer STM32CubeMX

New project

Onglet "Board Selector", sélectionner la carte NUCLEO F091RC Sélectionner le debugger , par exemple

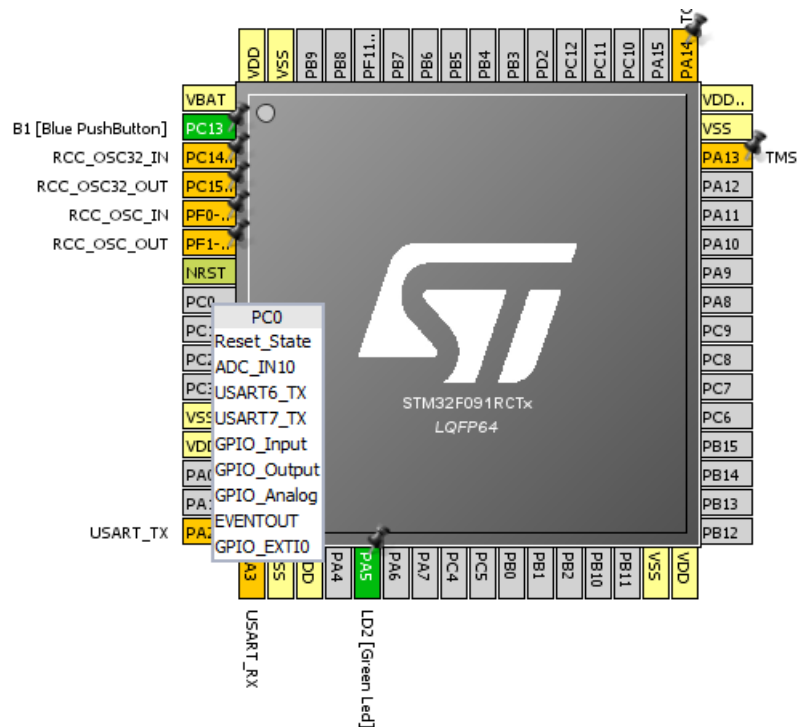
## STM32 Outils de développement

KEIL  $\mu$ Vision.



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



## STM32 Outils de développement



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

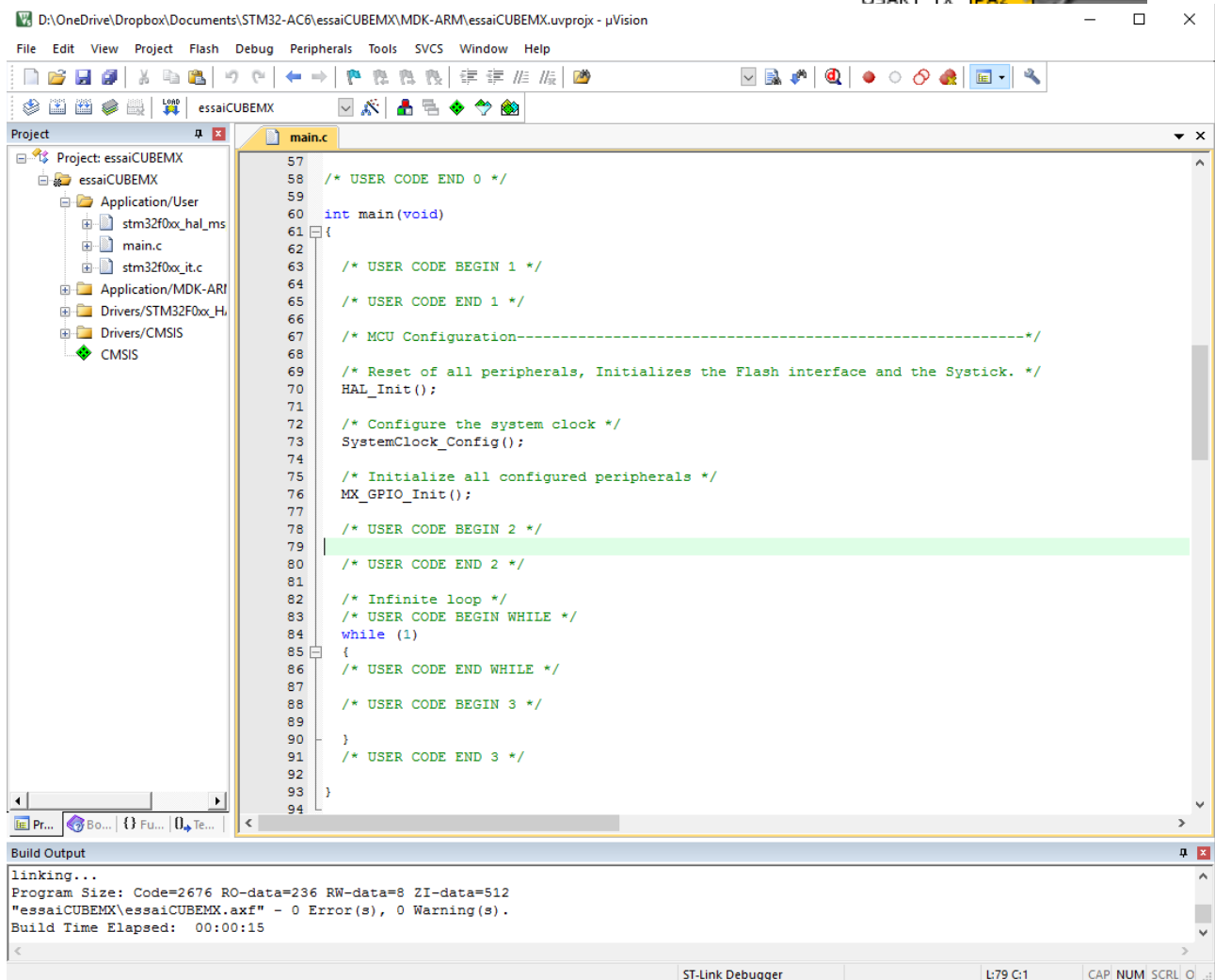
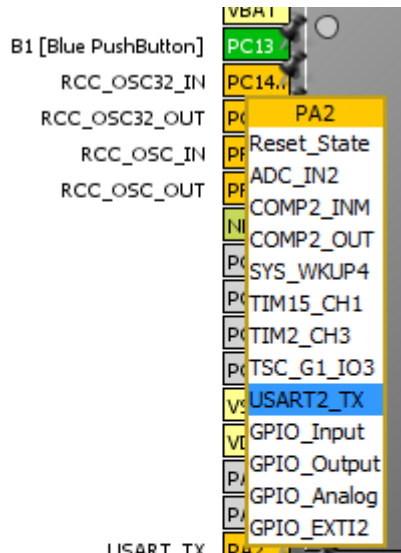
Project – Generate Code

Entrer un nom de projet et un chemin

OK

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

Après génération, CubeMX propose d'ouvrir le dossier du projet ou directement de lancer le débbuger avec le projet.



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

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





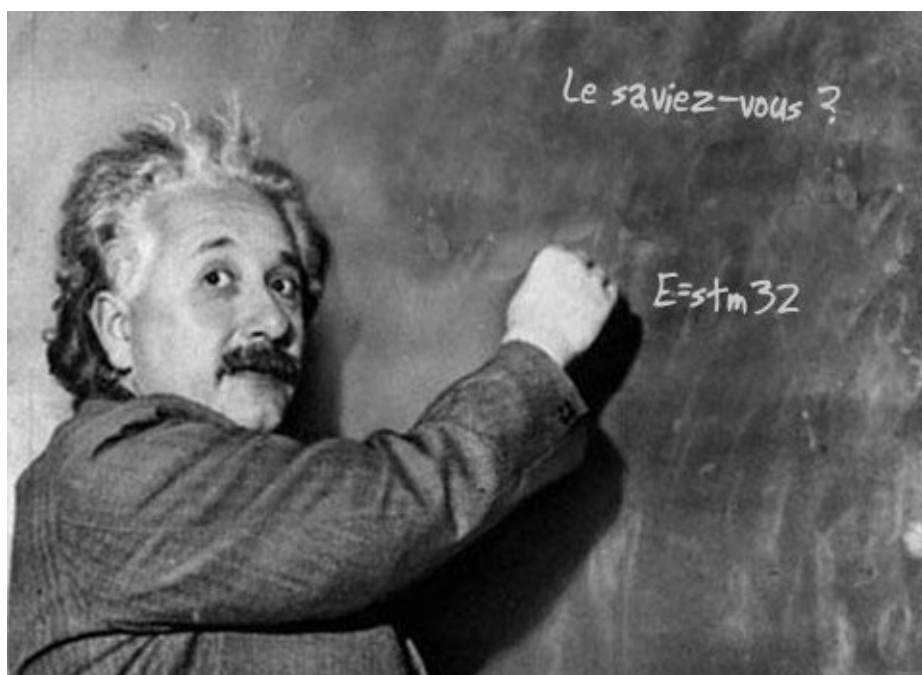
## STM32 Outils de développement

### 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](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](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](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 MICROCONTRÔLEURS 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](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](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](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](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\[NC,L\]](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/>