
# STM32CubeIDE : Prise en Main.

# 1 Introduction

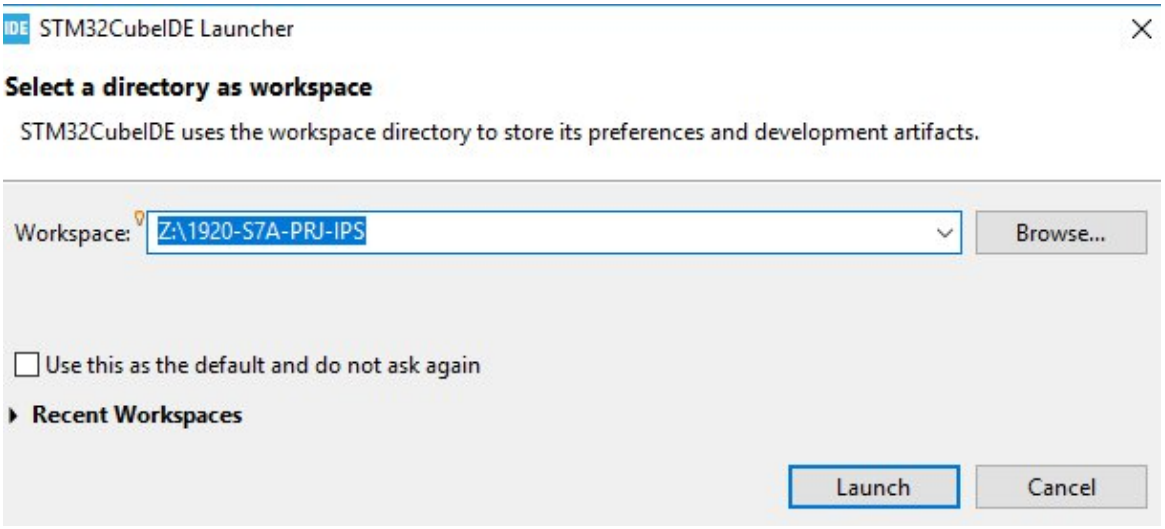
STM32CubeIDE est l'environnement de développement utilisé pour les cartes Nucléo. Ce logiciel multi plateformes (Linux, Windows, OSX) est disponible gratuitement sur le site de ST. Vous pouvez l'installer sur vos ordinateurs personnelles si nécessaire en suivant le lien suivant :

`https://www.st.com/en/development-tools/stm32cubeide.html`

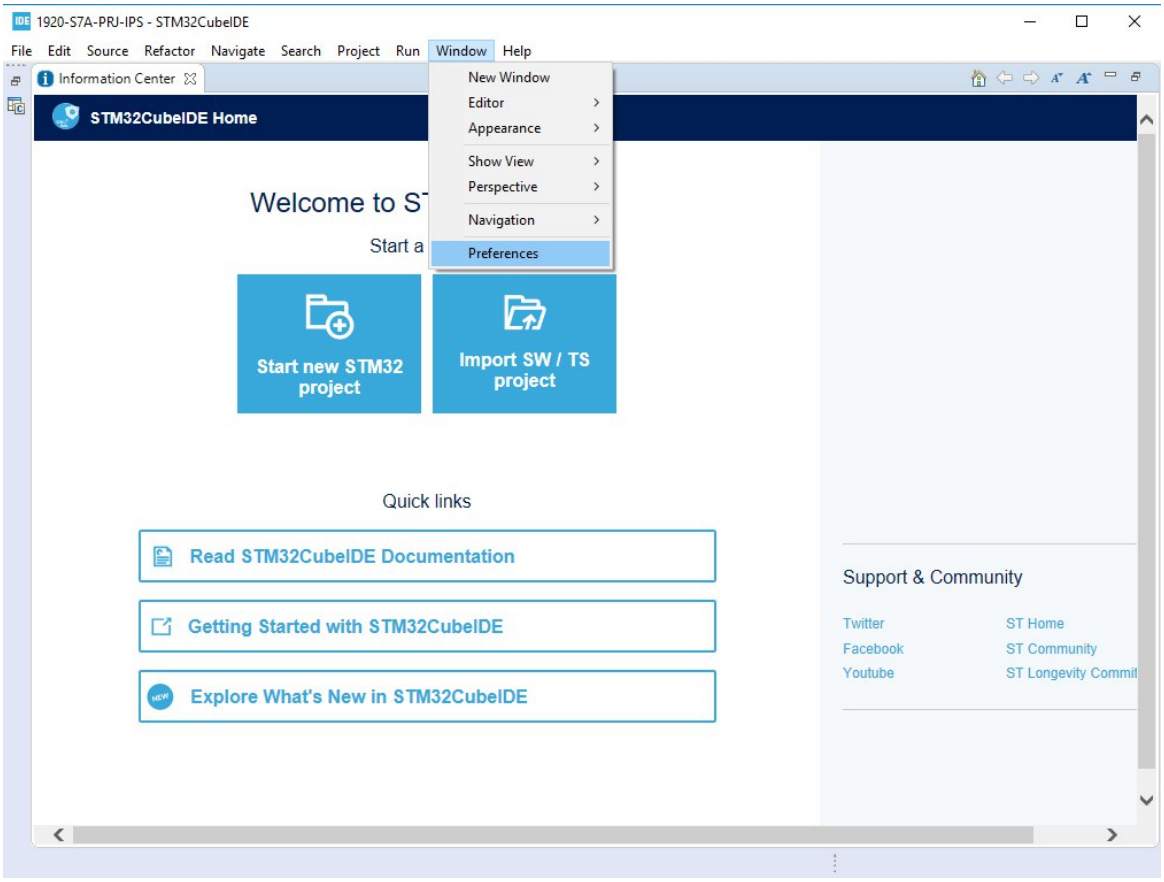
A chaque fois que vous utilisez une carte de développement différente le logiciel télécharge les kits de développement (SDK) liés un microcontrôleur utilisé. Lorsque vous utilisé les ordinateurs de la salle B002, par défaut ces SDK sont enregistrés dans votre profil sur le disque C: (profils effacés à chaque fois que vous vous déconnectez). Pour éviter d'avoir à re-télécharger (et perdre du temps) à chaque fois ces SDK, veuillez suivre la procédure suivante :

# 2 Procédure

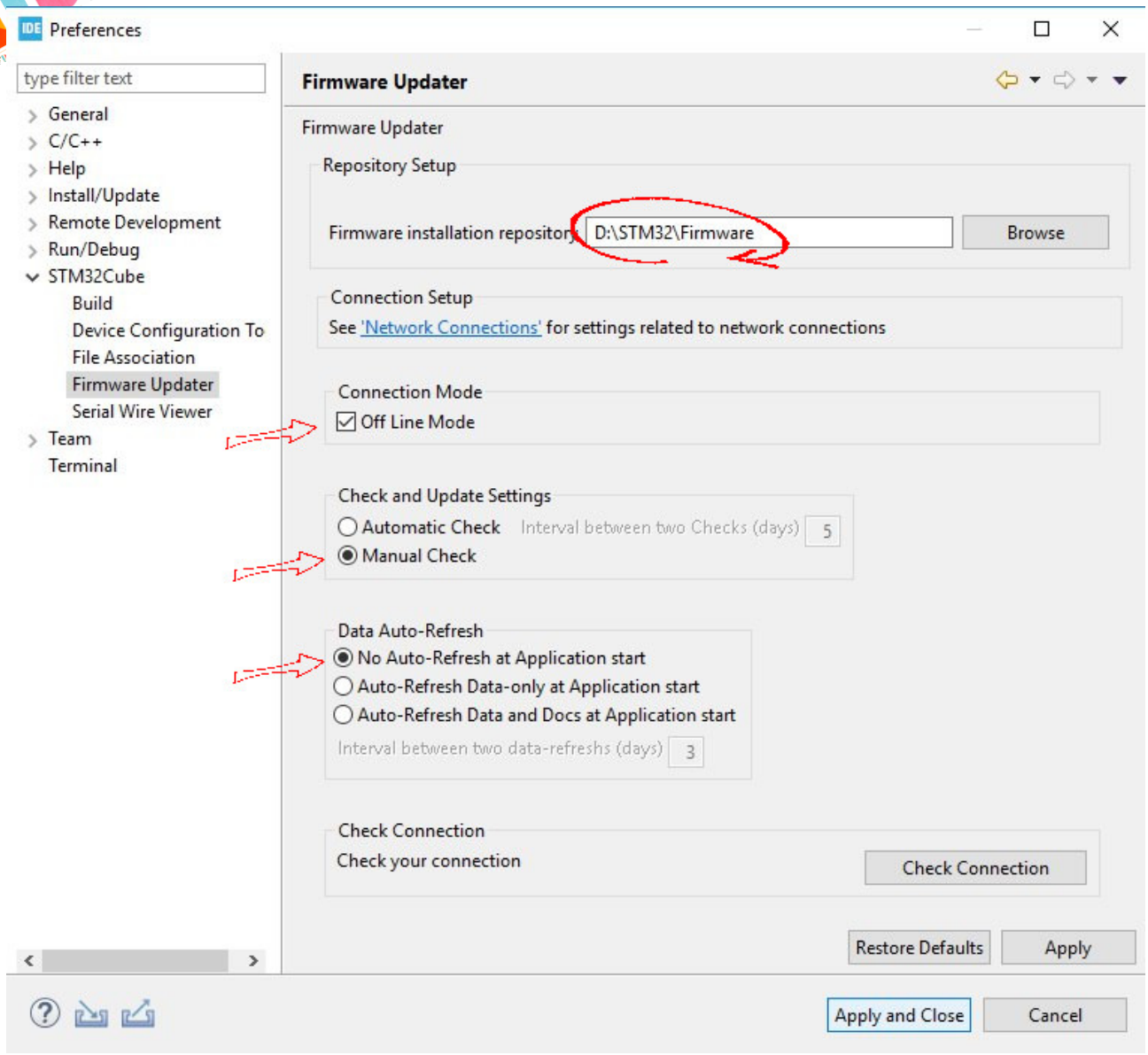
Lancer STM32CubeIDE sur un ordinateur de la salle B002.



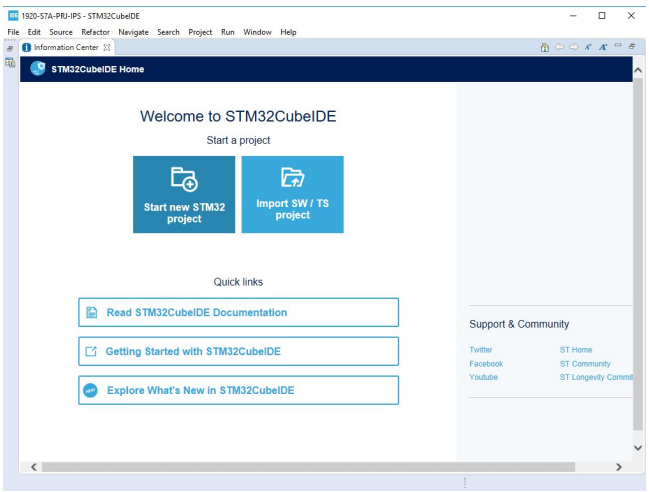

Choisir son "Workspace" sur un lecteur sauvegardé, dans mon cas c'est :
`Z:\1920-S7A-PRJ-IPS`



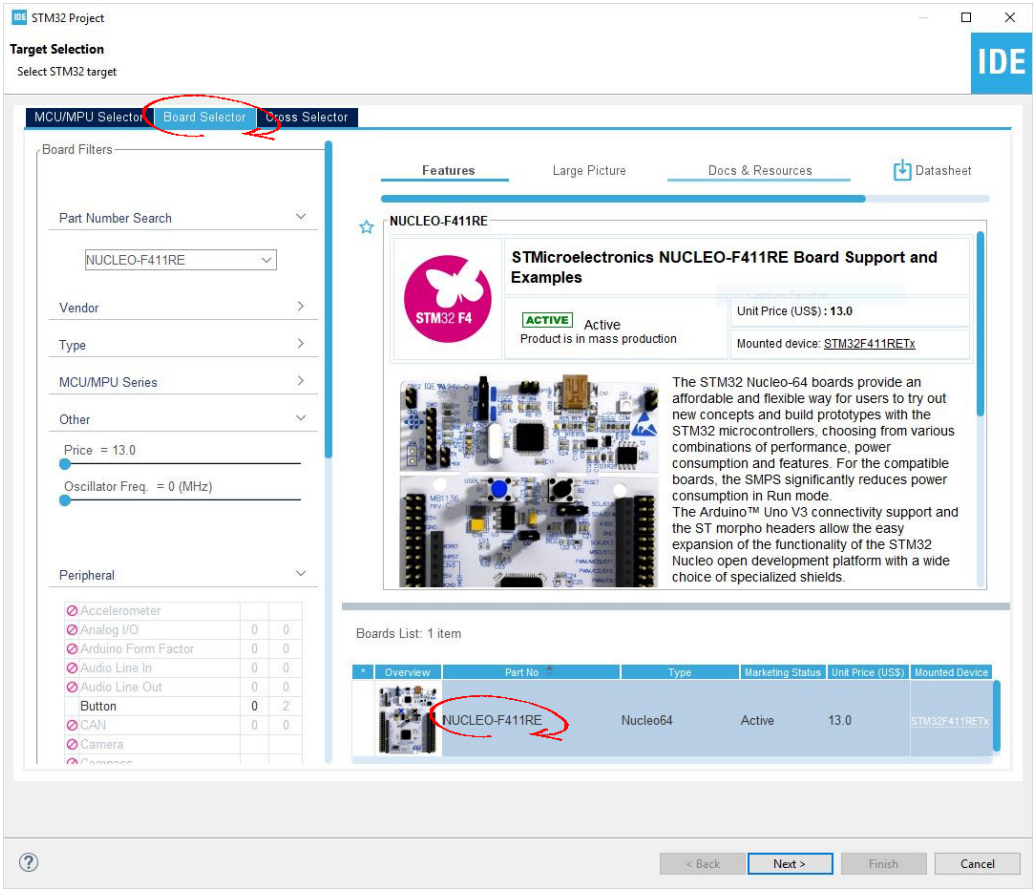

Avant tout, changer les préférences pour éviter d'avoir à recharger à chaque session les fichiers liés à votre carte STM sur les ordinateurs de la salle B002 (600Mo à télécharger à chaque fois). `Preferences -> STM32Cube -> Firmware Updater`



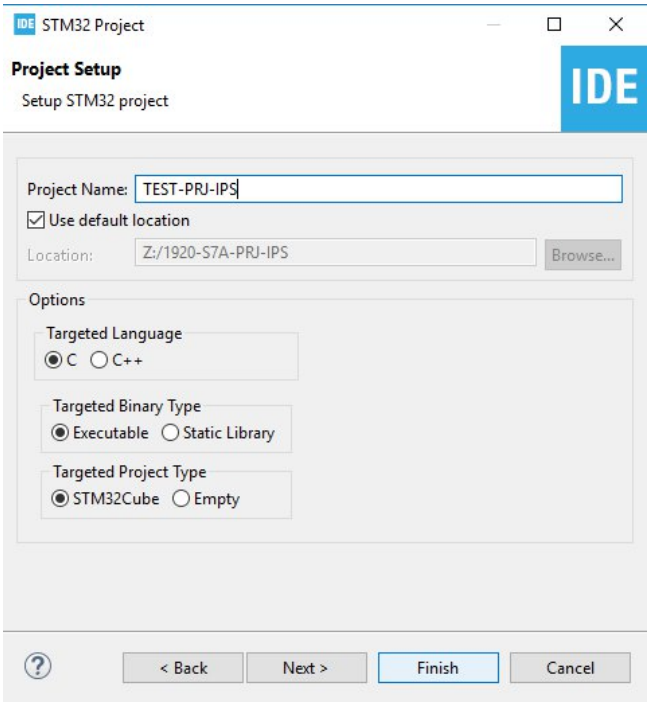

Choisir `D:\STM32\Firmware` pour le "Firmware installation repository"
Cocher "Offline mode"
Cocher Manual Check"
Cocher No auto-refresh at application start



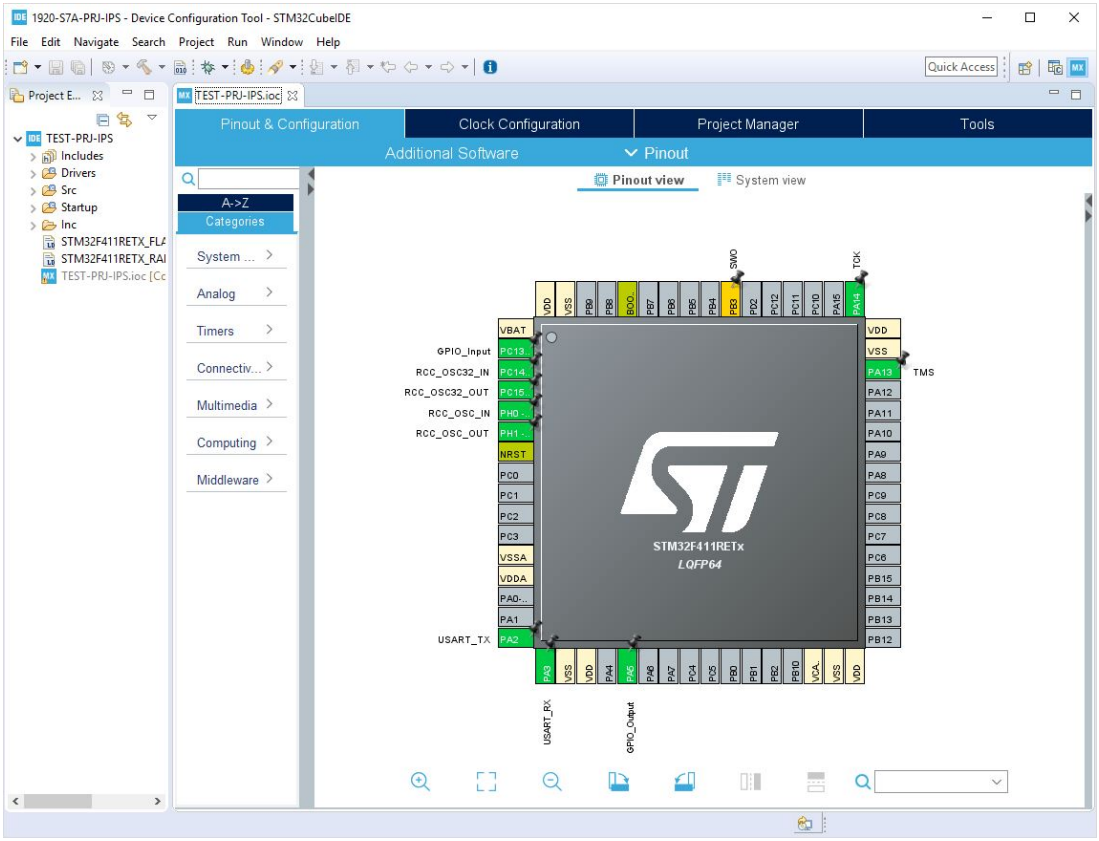

Start new STM32 project



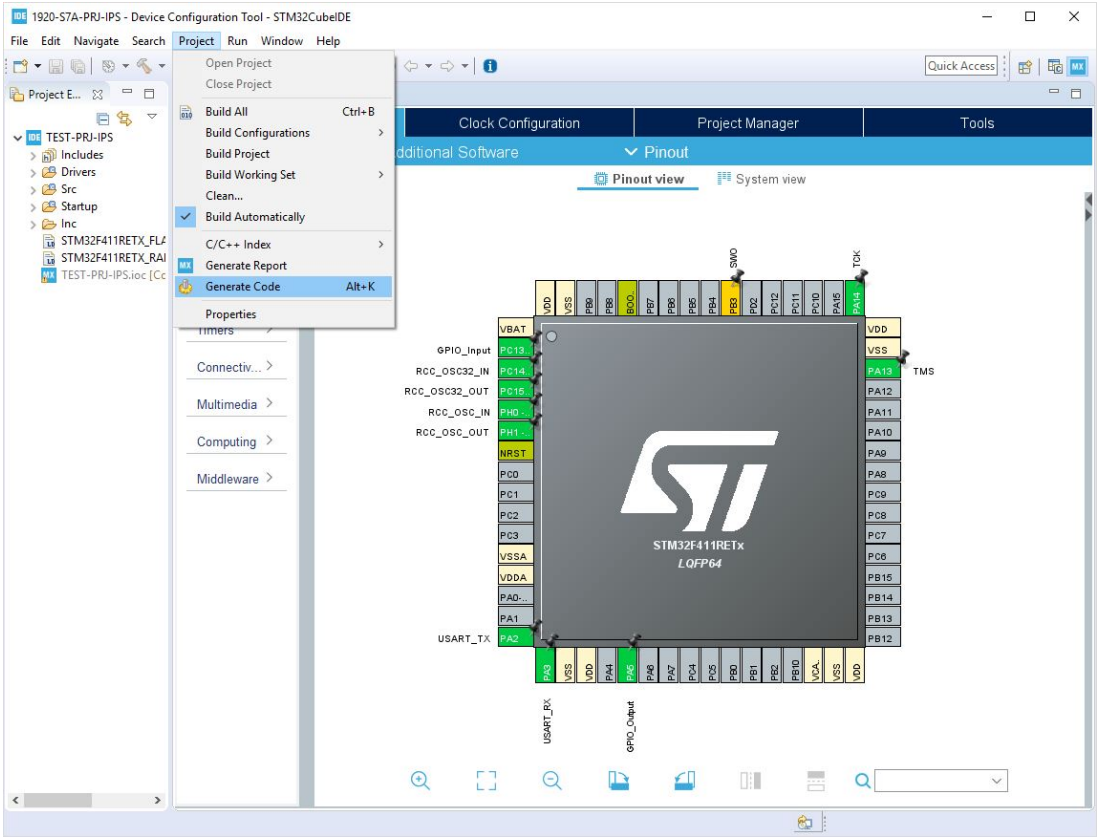

Dans l'onglet "Board Selector", choisir la carte nucleo utilisée : `NUCLEO-F411RE -> Next`



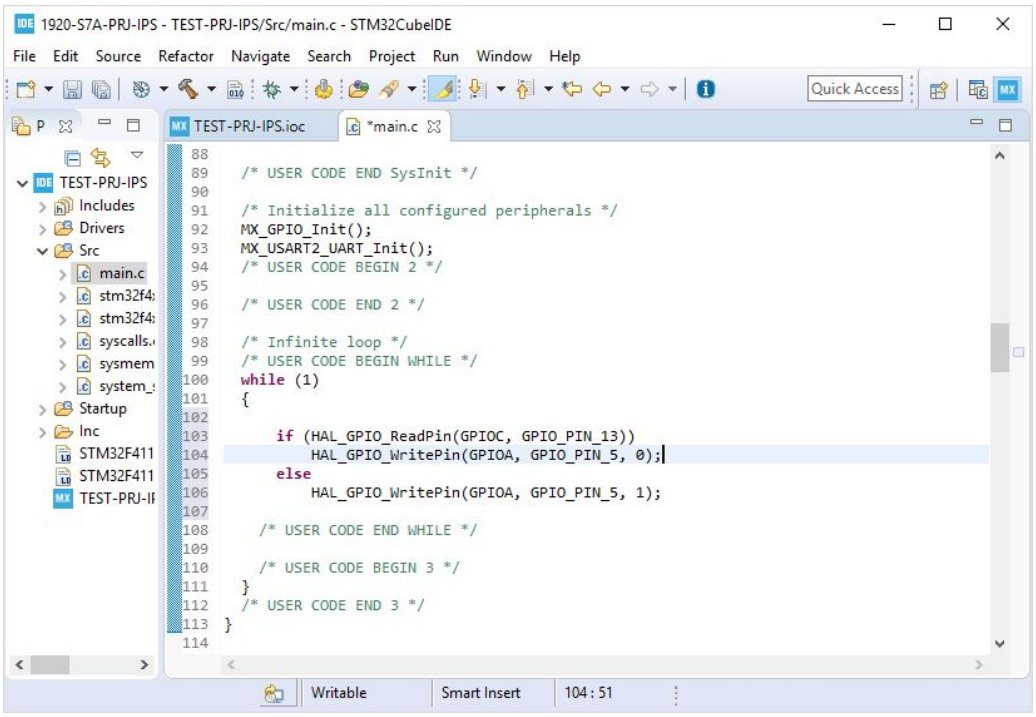

Donner un nom a votre projet (ici `TEST-PRJ-IPS`) `-> Finish`




Affecter les broches du composant en fonction de votre application, ici on fixe la broche PC13 en entrée numerique (c'est le bouton poussoir bleu), la broche PA5 en sortie numerique (c'est la LED LD2)



Générer le code de configuration, dans "src" vous trouverez le code généré, qui correspond à ce que vous avez configuré dans CubeMX.

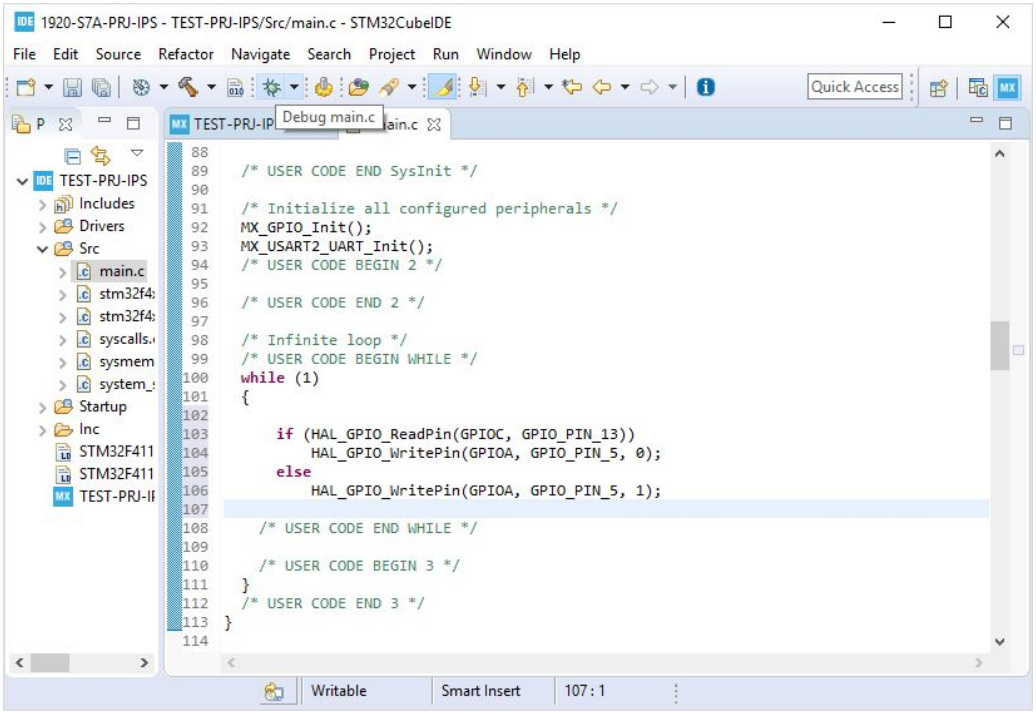



Vous pouvez compléter le squelette du code de configuration avec votre code applicatif.

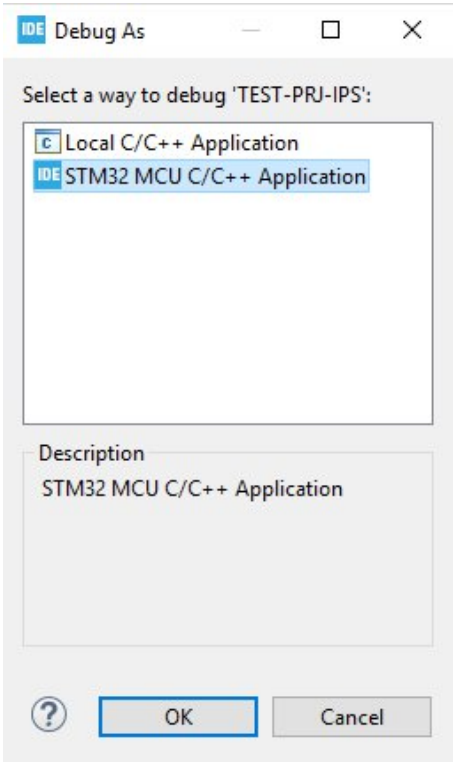
**Attention !** pour que votre code ne soit pas effacé si vous re-générez le code depuis CubeMX (si vous changez la config de votre microcontrôleur par exemple), il faut placer votre code perso **ENTRE** les `USER CODE BEGIN` et `USER CODE END`.

On peut placer le code suivant dans la boucle `while(1)` du `main()`

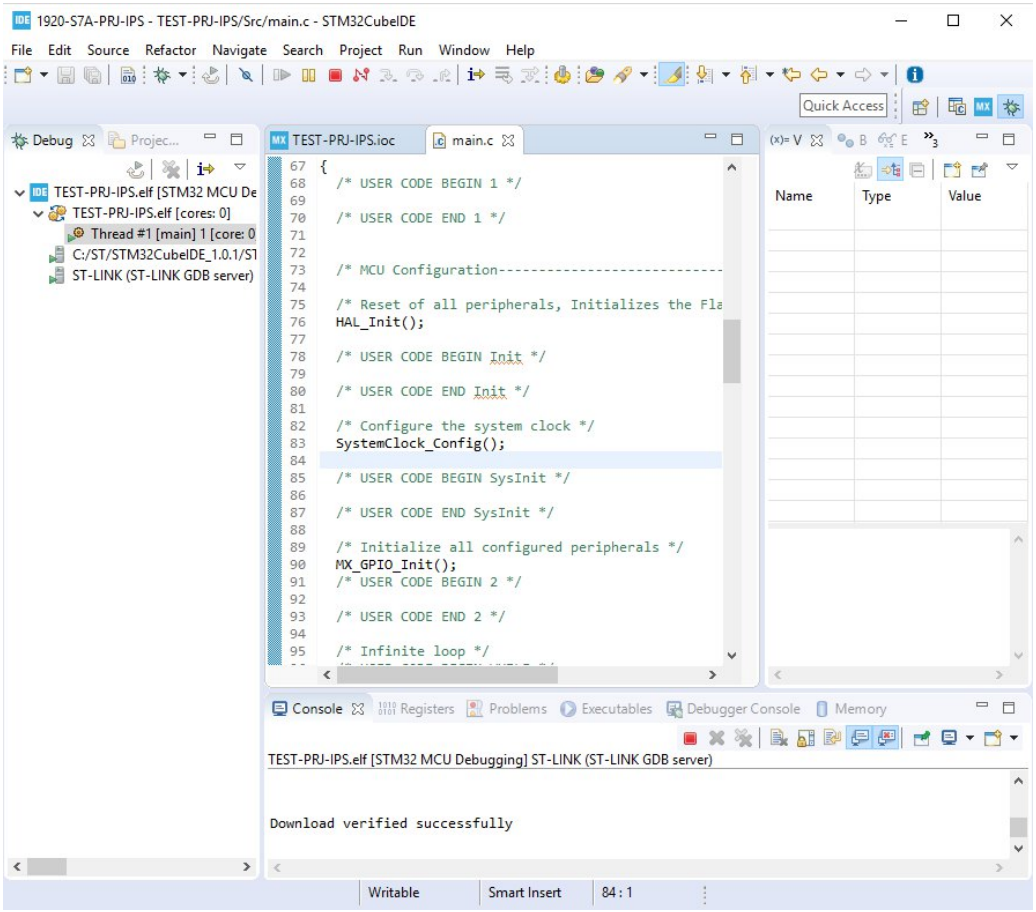
```
if  (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13))
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
else
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
```




Cliquer sur `debug`




choisir `STM32 MCU C/C+` Application+ puis basculer en mode `debug`




La touche F8 permet de lancer le programme

Un appui sur le bouton bleu permet d'allumer la led LD2

# 3 Conclusion

Vous avez pris en main la carte nucleo, maintenant à vous de développer un code plus conséquent répondant aux attentes de votre projet.

Voici quelques exercices que vous pouvez faire qui vous serons très utile pour la réalisation de votre projet.

- Utiliser une interruption (contrôler la LED avec le bouton poussoir, mais en utilisant une interruption sur le bouton poussoir).
- Définir un Timer toute les secondes générant une interruption, et utiiser cette interruption pour faire clignoter la LED.
- Utiliser une sortie PWM pour faire varier l'intensité lumineuse d'une LED.
- Envoyer "Hello World" sur le port série (via le port USB de la carte nucleo).
- Connecter un potentiomètre sur une entrée analogique, renvoyer la tension sur le port série.