

Table des matières

I) DEBUTER SIMPLEMENT AVEC LA CARTE NUCLEO-H755ZI-Q	2
II) EXEMPLE SIMPLE « GPIO-EXTI »	2
II-1) TELECHARGEMENT DU PACKAGE LOGICIEL.....	2
II-2) INSTALLATION DU PACKAGE LOGICIEL.....	3
II-3) POSITONNEMENT DANS LE REPERTOIRE DE L'EXEMPLE	3
II-4) COMPILATION ET CHARGEMENT DU CODE.....	3
II-5) EXECUTION DU CODE SUR LA CARTE	4
III) MEME EXEMPLE MAIS AVEC PERSONNALISATION(GAIN DE PLACE DISQUE)	4
III-1) PERSONNALISATION DE L'INSTALLATION DES EXEMPLES.....	4
III-2) COMPILATION DE L'EXEMPLE GPIO_EXTI	4
III-3) EXECUTION DU CODE SUR LA CARTE	8

I) Débuter simplement avec la carte Nucléo-H755ZI-Q

Dans la suite de ce document, je suppose que votre environnement de développement (IDE) est Keil μ Vision5 version 5.29.0.0 et qu'il est déjà installé avec les packages nécessaires au développement avec STM32H7.

Je considère également que votre carte NUCLEO-H755ZI-Q est configurée pour être raccordée à un port USB de votre ordinateur, via le connecteur CN1.

Au moment où j'écris ces lignes, il n'existe pas d'exemple tout fait pour la carte NUCLEO-H755ZI-Q. La littérature autour des cartes Nucléo et des microcontrôleurs STM32 est surabondante et déroutante, néanmoins en y consacrant beaucoup de temps, on finit par y trouver des indications. On trouve notamment ceci dans le document UM2204 :

The STM32CubeH7 features a rich set of examples and demonstrations at all levels making it easy to understand and use any HAL driver and/or middleware components. These examples can be run on any of the STMicroelectronics boards as listed in [Table 2. Evaluation, Discovery and Nucleo boards for STM32H7 Series](#) :

Table 2. Evaluation, Discovery and Nucleo boards for STM32H7 Series

Board	Supported STM32H7 lines
STM32H743I-EVAL	STM32H743xx, STM32H753xx and STM32H750xx ⁽¹⁾
NUCLEO-H743ZI	
NUCLEO-H745ZI-Q	STM32H745xx and STM32H755xx
STM32H745I-DISCO	STM32H745xx and STM32H755xx
STM32H747I-DISCO	STM32H747xx and STM32H757xx
STM32H747I-EVAL	STM32H747xx and STM32H757xx
STM32H750B-DK	STM32H750xx

Ce qui nous laisse penser que les exemples réalisés pour le STM32H745, fonctionneront sur le STM32H755. Nous partons donc sur cette hypothèse.

II) Exemple simple « GPIO-EXTI »

Cet exemple simple va consister à allumer ou éteindre les deux LED verte et jaune (LD1 et LD2) à chaque appui sur bouton B1 (bleu) qui va générer une interruption externe sur le GPIO PC13 affecté à ce bouton.

Cet exemple se trouve dans le répertoire « GPIO-EXTI.

II-1) Téléchargement du package logiciel

Il faut aller sur le site de www.st.com et rechercher le package « STM32CubeH7 STM32Cube MCU Package for STM32H7 series ».

Au moment de l'écriture de ce document, le dernier package en vigueur était :

« **en.stm32cubeh7-v1.7.0.zip** ».

Téléchargez ce package logiciel.

II-2) Installation du package logiciel

Décompressez l'énorme fichier « .zip » dans le répertoire de votre choix, par exemple dans :
« D:\Documents\Informatique\Programmes\STM32Cube\Repository\ »

II-3) Positionnement dans le répertoire de l'exemple

Dans le cas de cet exemple, il faut se rendre dans le répertoire :

« D:\Documents\Informatique\Programmes\STM32Cube\Repository\STM32Cube_FW_H7_V1.7.0\Projects\NUCLEO-H745ZI-Q\Examples\GPIO\GPIO_EXTI\MDK-ARM\ »

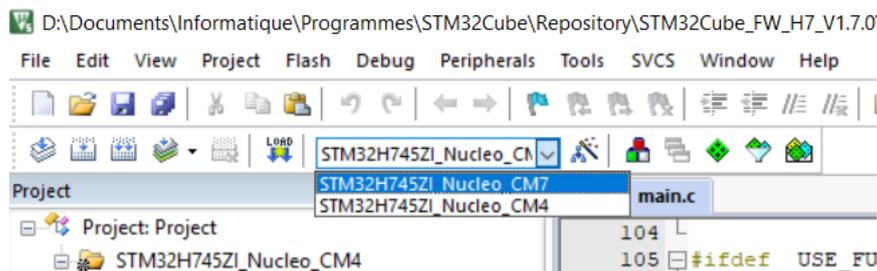
Nous y trouvons 3 fichiers :

- « Project.uvoptx »
- « Project.uvprojx »
- « startup_stm32h745xx.s »

II-4) Compilation et chargement du code

1) Double cliquez sur « **Project.uvprojx** »

Grâce à cette boîte déroulante, sélectionnez la cible « **STM32H745ZI_Nucleo_CM7** »



Rappelez-vous que nous avons émis comme hypothèse que ce qui a été écrit pour le H745 doit fonctionner sur le H755...

2) Appuyez sur la touche de fonction « **F7** »

→ ceci permet de compiler le code du *core M7*.

3) Appuyez sur la touche de fonction « **F8** »

→ ceci permet de déverser le code compilé pour le *core M7* dans la carte NUCLEO.

4) Avec la même boîte déroulante que précédemment, choisir la cible « **STM32H745ZI_Nucleo_CM4** ».

5) Appuyez sur la touche de fonction « **F7** »

→ ceci permet de compiler le code du *core M4*.

6) Appuyez sur la touche de fonction « **F8** »

→ ceci permet de déverser le code compilé pour le *core M4* dans la carte NUCLEO.

II-5) Exécution du code sur la carte

Une fois les deux codes déversés dans la carte, vous devez faire un *reset* en appuyant sur son bouton noir.

Constatez maintenant qu'en appuyant sur le bouton bleu, les 2 leds s'allument ou s'éteignent simultanément.

Dans cet exemple, le core M7 se charge d'allumer ou d'éteindre LD1 tandis que le core M4 se charge d'allumer ou d'éteindre LD2.

III) Même exemple mais avec personnalisation(gain de place disque)

Vous n'avez pas été sans remarquer que le téléchargement du package contenant l'ensemble des exemples de la famille STM32H7 est très consommateur en place disque. Vous aimeriez certainement pouvoir copier un seul exemple du package ailleurs sur votre disque et ne garder que le code nécessaire au fonctionnement de la famille STM32H7 en général.

III-1) Personnalisation de l'installation des exemples

Pour gagner de la place en supprimant les exemples des modèles qui ne me concernaient pas, voici comment j'ai procédé :

- 1) J'ai téléchargé et installé dans le répertoire
D:\Documents\Informatique\Programmes\STM32Cube\Repository » l'ensemble du package (le même que dans le chapitre [II-1-a\) Téléchargement du package logiciel](#)).
- 2) J'ai déplacé les exemples qui concernent notre carte dans un répertoire personnalisé. Pour l'exemple qui nous concerne «
D:\Documents\Informatique\Programmes\STM32Cube\Repository\STM32Cube_FW_H7_V1.7.0\Projects\NUCLEO-H745ZI-Q\Examples\GPIO\GPIO_EXTI» a été déplacé vers
D:\Documents\Informatique\Programmes\STM32\NUCLEO-H755ZI-Q »
- 3) J'ai supprimé le sous-répertoire « Project » du répertoire
D:\Documents\Informatique\Programmes\STM32Cube\Repository\STM32Cube_FW_H7_V1.7.0», gardant ainsi uniquement tout ce qui est commun aux divers exemples (les drivers BSP, CMSI, HAL, Middlewares, Utilities, Documentation).

III-2) Compilation de l'exemple GPIO_EXTI

Allons dans le répertoire

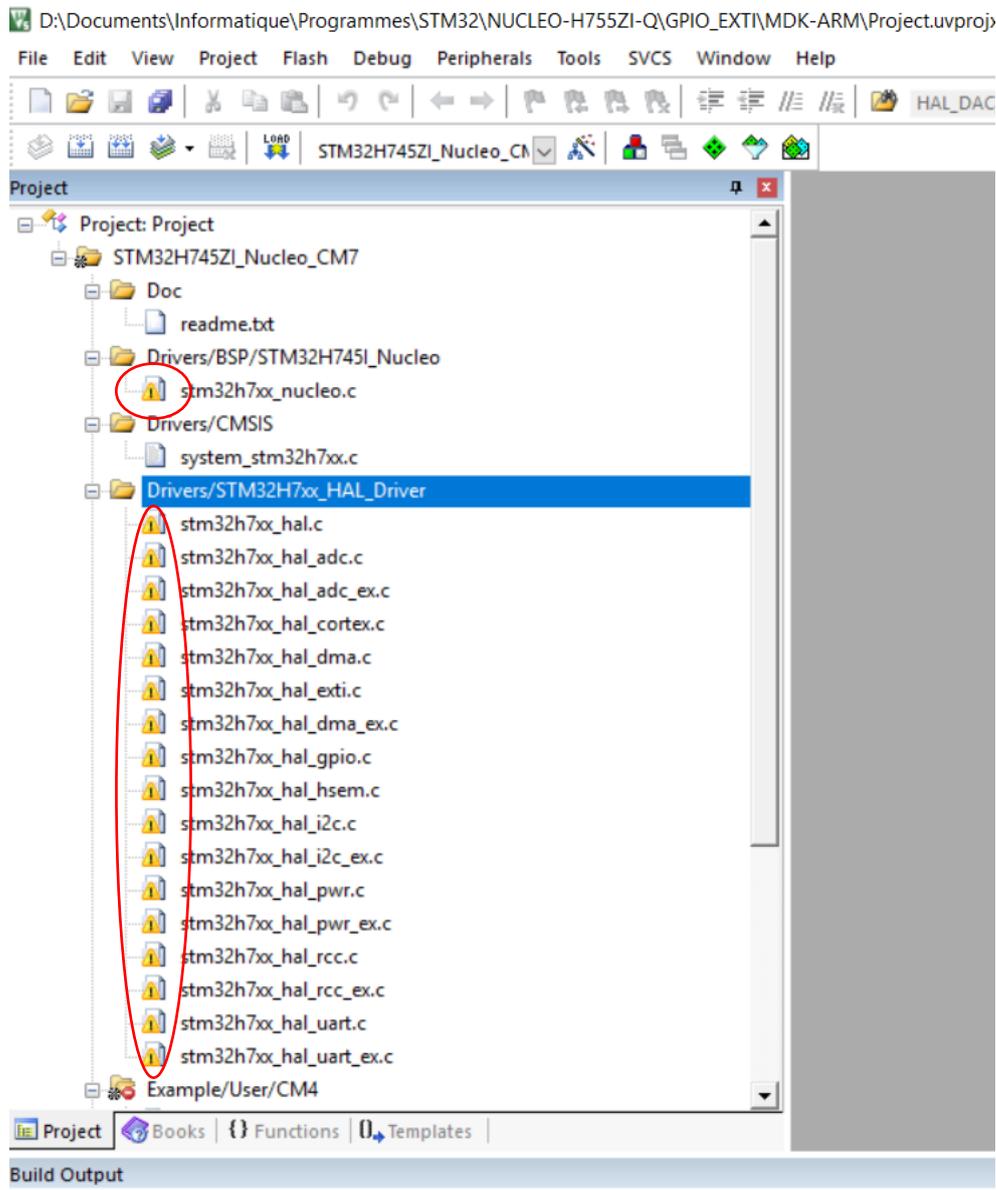
D:\Documents\Informatique\Programmes\STM32\NUCLEO-H755ZI-Q\GPIO_EXTI\MDK-ARM»

Nous y trouvons 3 fichiers :

- « Project.uvoptx »
- « Project.uvprojx »
- « startup_stm32h745xx.s »

1) Double cliquons sur « **Project.uvprojx** »

On peut voir qu'il y a un problème, car beaucoup de fichiers ne sont pas trouvés par Keil µVision :



C'est lié au fait que nous avons déplacé l'arborescence de l'exemple en dehors de son package d'origine.

- Et dans Keil µVision, il n'y a aucune facilité pour remédier à cela simplement, la seule façon de faire est de prendre le fichier « **Project.uvprojx** », de l'ouvrir dans un éditeur de texte puissant comme « Sublime Text » et de corriger les chemins incorrects.

Si on laisse la souris au dessus d'un des fichiers non trouvés, on voit ceci :



En l'occurrence, le chemin qui pose problème est relatif au répertoire « Drivers » qui en effet ne se situe pas dans D:\Documents, mais dans

« **D:\Documents\Informatique\Programmes\STM32Cube\Repository\STM32Cube_FW_H7_V1.7.0** »

2) Fermons Keil μ Vision

3) Editons « **Project.uvprojx** » dans notre éditeur préféré

En balayant ce fichier, on finit par tomber sur les fichiers inaccessibles, par exemple :

```
<File>
  <FileName>stm32h7xx_hal.c</FileName>
  <FileType>1</FileType>
  <FilePath>../../../../../../../../Drivers/STM32H7xx_HAL_Driver/Src/stm32h7xx_hal.c</FilePath>
</File>
<File>
  <FileName>stm32h7xx_hal_adc.c</FileName>
  <FileType>1</FileType>
  <FilePath>../../../../../../../../Drivers/STM32H7xx_HAL_Driver/Src/stm32h7xx_hal_adc.c</FilePath>
</File>
<File>
  <FileName>stm32h7xx_hal_adc_ex.c</FileName>
```

4) Faisons la substitution, remplaçons d'un seul coup tout ce qui commence par

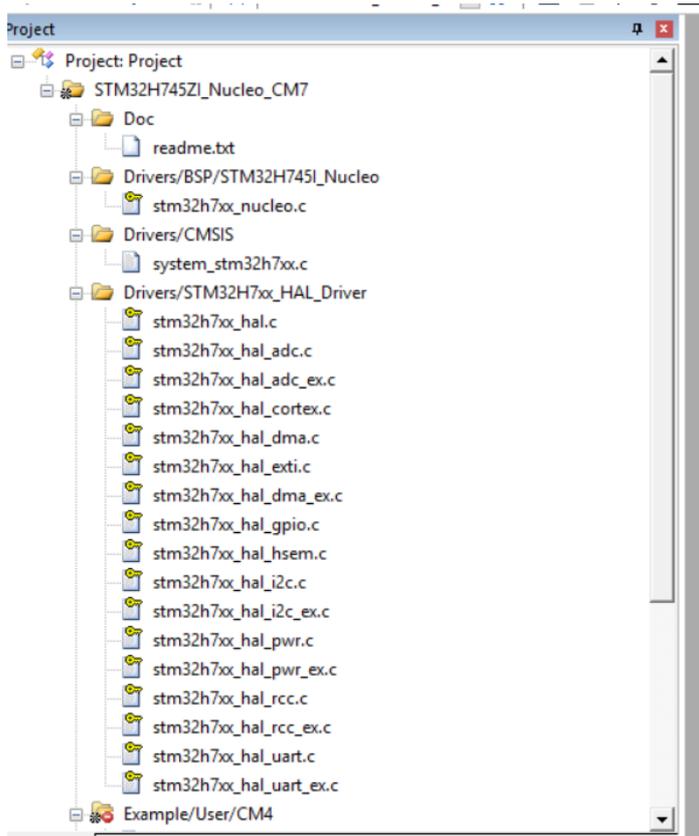
« **../../../../../../../../Drivers** » par

«**D:\Documents\Informatique\Programmes\STM32Cube\Repository\STM32Cube_FW_H7_V1.7.0\Drivers** »

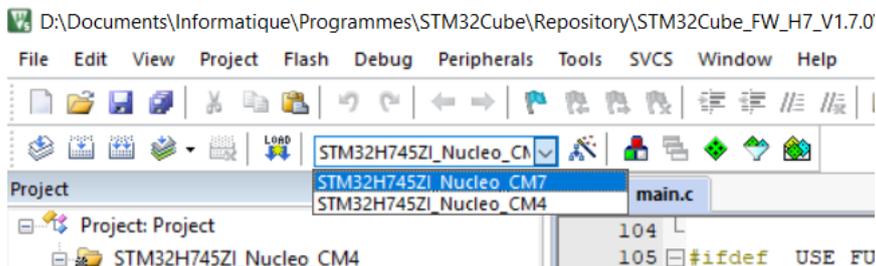
5) Sauvegardons le fichier modifié

6) Double cliquons de nouveau dessus.

Nous voyons que désormais Keil μ Vision sait trouver les fichiers du projet.



7) Sélectionnons la cible « **STM32H745ZI_Nucleo_CM7** »



8) Appuyons sur la touche de fonction « **F7** »

→ ceci permet de compiler le code du *core M7*.

9) Appuyons sur la touche de fonction « **F8** »

→ ceci permet de déverser le code compilé pour le *core M7* dans la carte NUCLEO.

10) Avec la même boîte déroulante que précédemment, choisissons la cible « **STM32H745ZI_Nucleo_CM4** ».

5) Appuyons sur la touche de fonction « **F7** »

→ ceci permet de compiler le code du *core M4*.

6) Appuyons sur la touche de fonction « **F8** »

→ ceci permet de déverser le code compilé pour le *core M4* dans la carte NUCLEO.

III-3) Exécution du code sur la carte

Une fois les deux codes déversés dans la carte, vous devez faire un *reset* en appuyant sur son bouton noir.

Constatez maintenant qu'en appuyant sur le bouton bleu, les 2 leds s'allument ou s'éteignent simultanément.

Dans cet exemple, le core M7 se charge d'allumer ou d'éteindre LD1 tandis que le core M4 se charge d'allumer ou d'éteindre LD2.