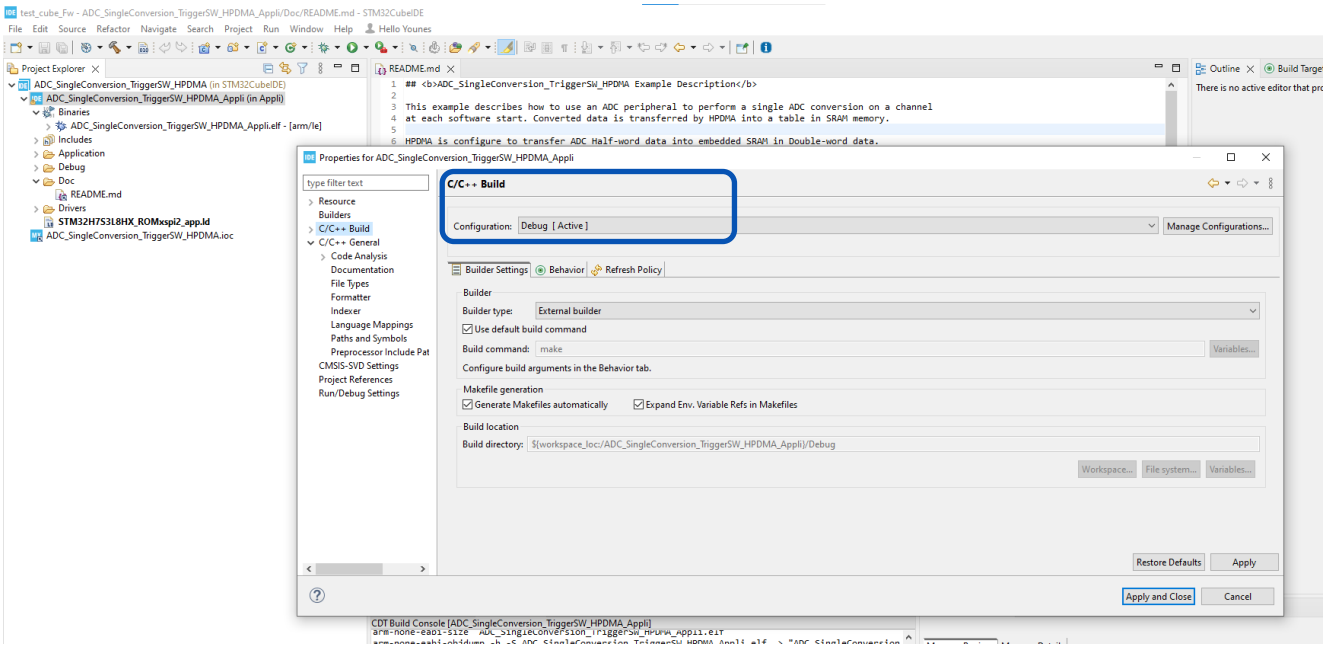
STM32CubeIDE: How to debug STM32H7Rx/Sx project without flashing boot every time

Introduction

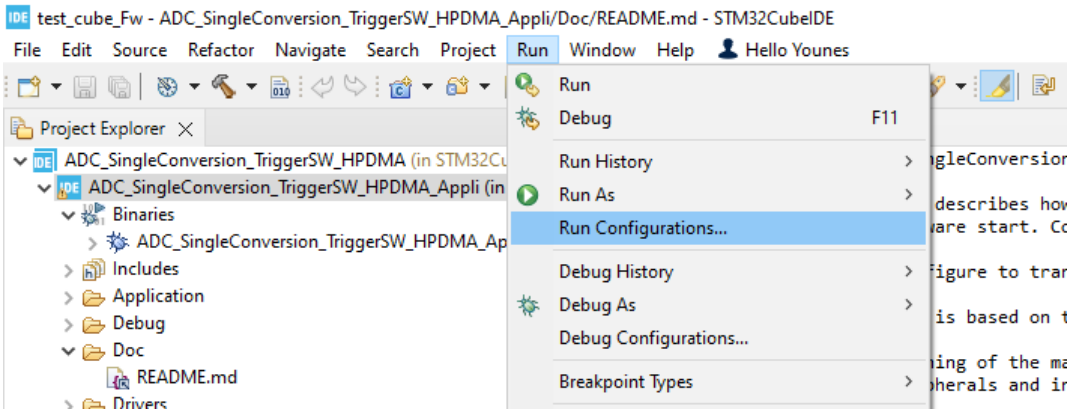
The STM32H7R/S devices come with 64k bytes of boot flash allowing applications to run from external memory. The STM32CubeH7RS Firmware package comes with a rich set of examples running on STM32H7S or STM32H7R boards. Each example requires at least 2 subprojects: boot and appli.
The readme file describes how to configure the multi-projects for the following toolchains: IAR, MDK-ARM and STM32CubeIDE. However, STM32CubeIDE multi-project seems a bit complicated for non-familiar users. To better facilitate the steps mentioned in the readme file, this document illustrates each step with snapshot images for more clarifications.

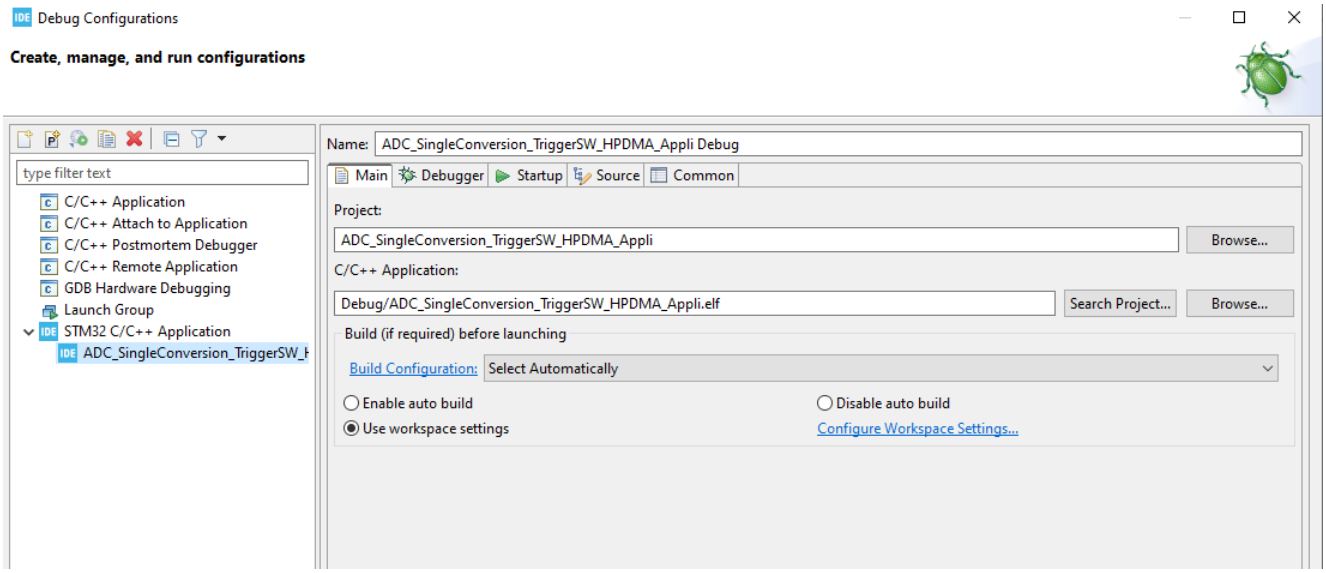
STM32CubeIDE multi-project configuration

- Compile the example/application.
    - the elf file is required to configure the debug profile (the “active configuration” must be “debug”, else only assembly debug is available)

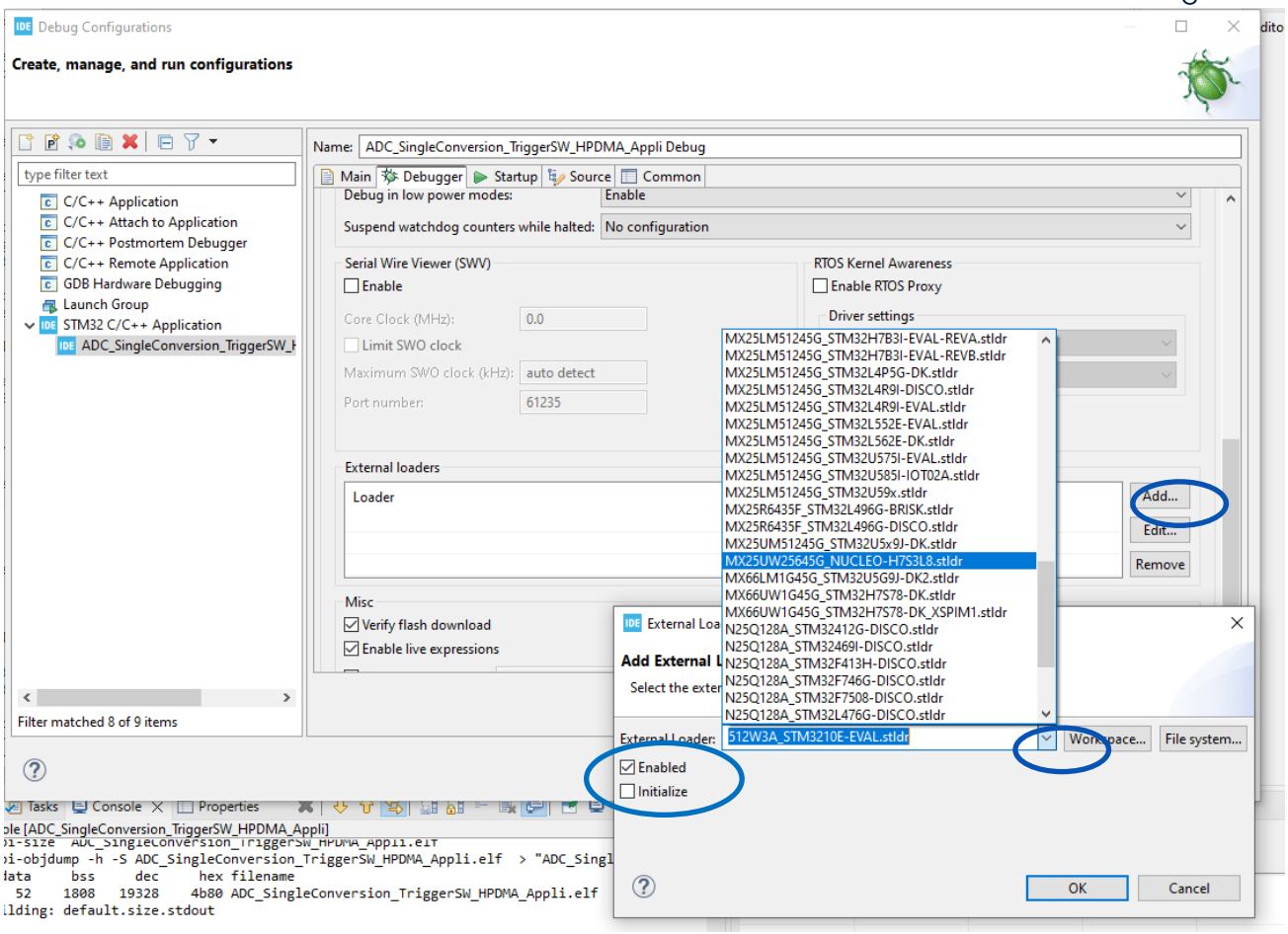


- Open the menu [Run]->[Debug configuration] and double click on [STM32 C/C++ Application]
    - It creates a default debug configuration for the current project selected

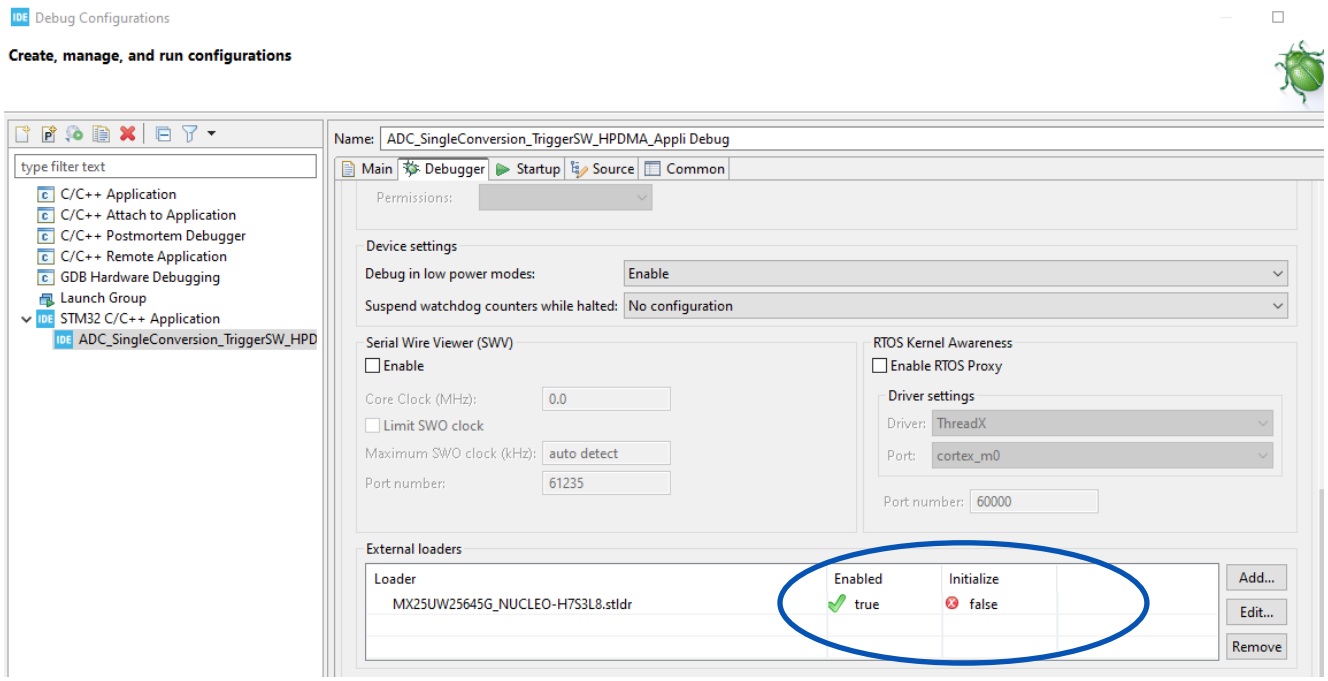




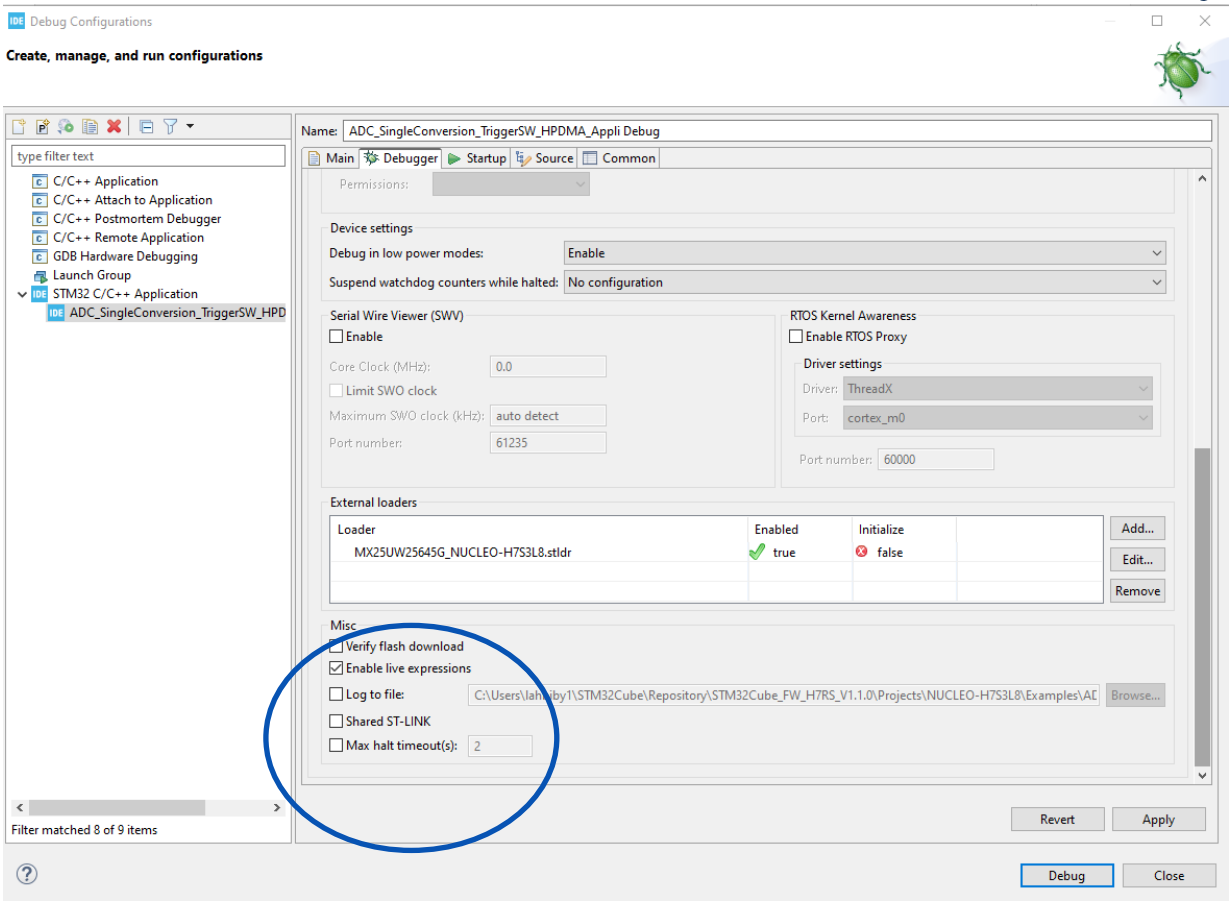
- In [Debugger] tab, section “External loaders” add the external loader corresponding to your Board/Memory as described below:
  - In “External loaders” section, click on [Add]
  - Select the loader among the available list (**MX25UW25645G_NUCLEO-H7S3L8.stldr** or **MX66UW1G45G_STM32H7S78-DK.stldr**)



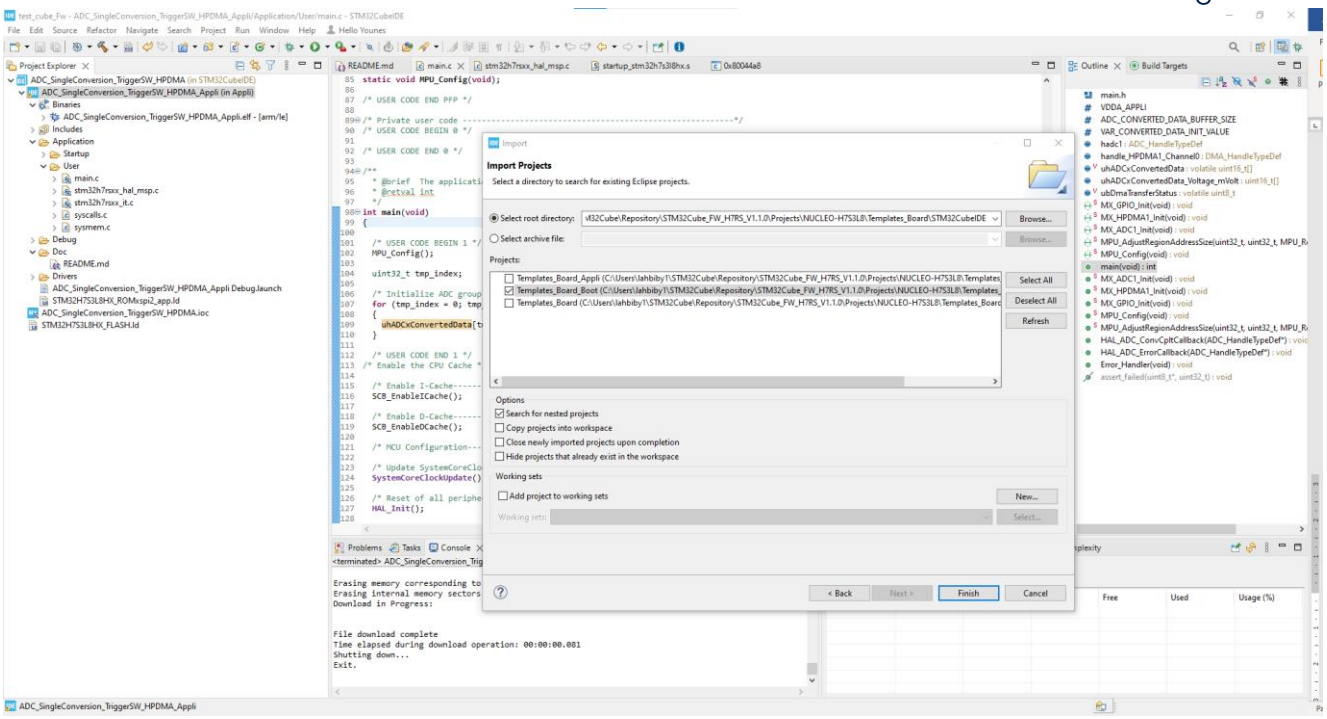
- Option “Enabled” checked and Option “Initialize” unchecked.
  - If not double click to change the settings



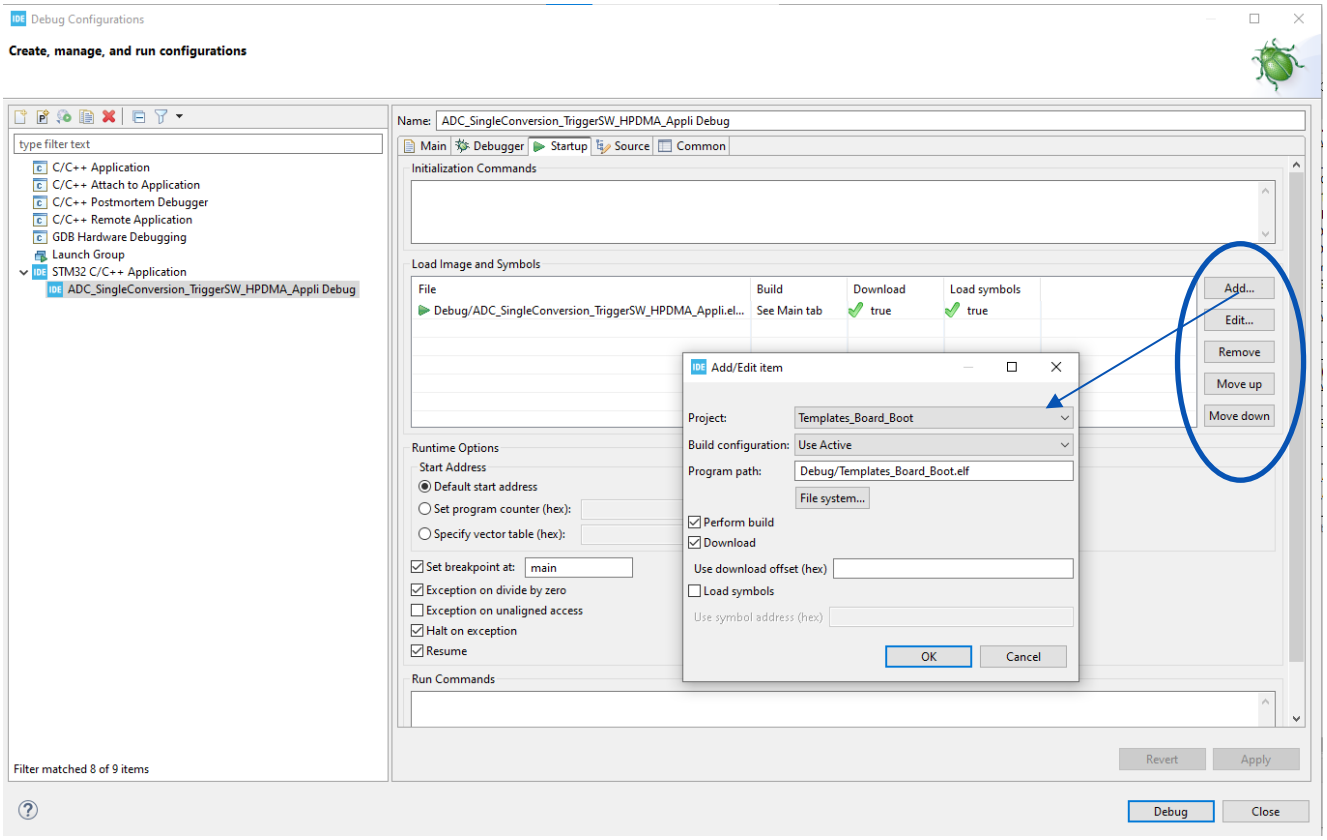
- In “Misc” section, uncheck the option “Verify flash download.”

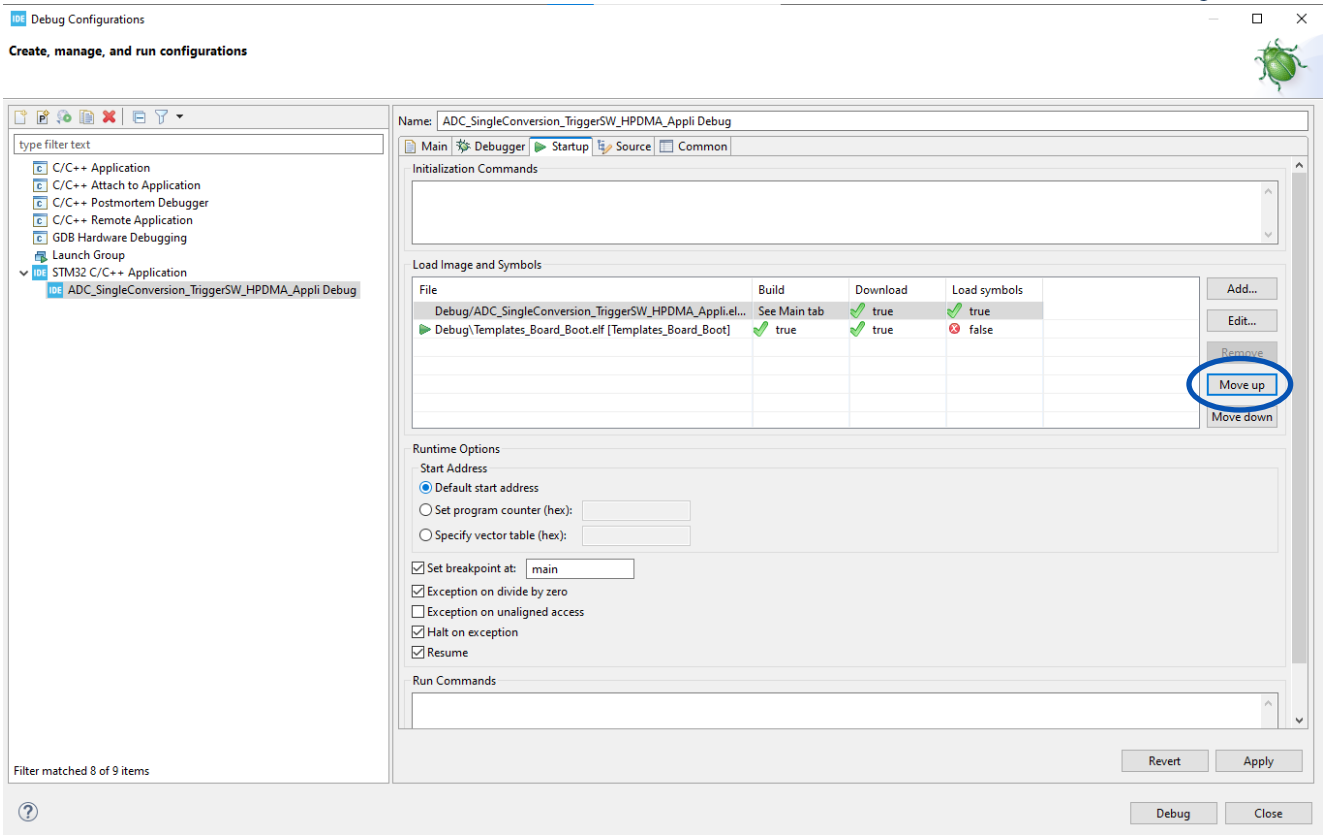


- In [Startup] tab, section “Load Image and Symbols”:
  - Click on [Add]
  - If your project contains a boot project:
    - click on “Project” and then select the boot project.
    - click on Build configuration and select “Use active”.
    - then select the following options:
      - “Perform build” checked.
      - “Download” checked.
      - “Load symbols” unchecked.
  - For example, you can import the boot project into workspace as follow.



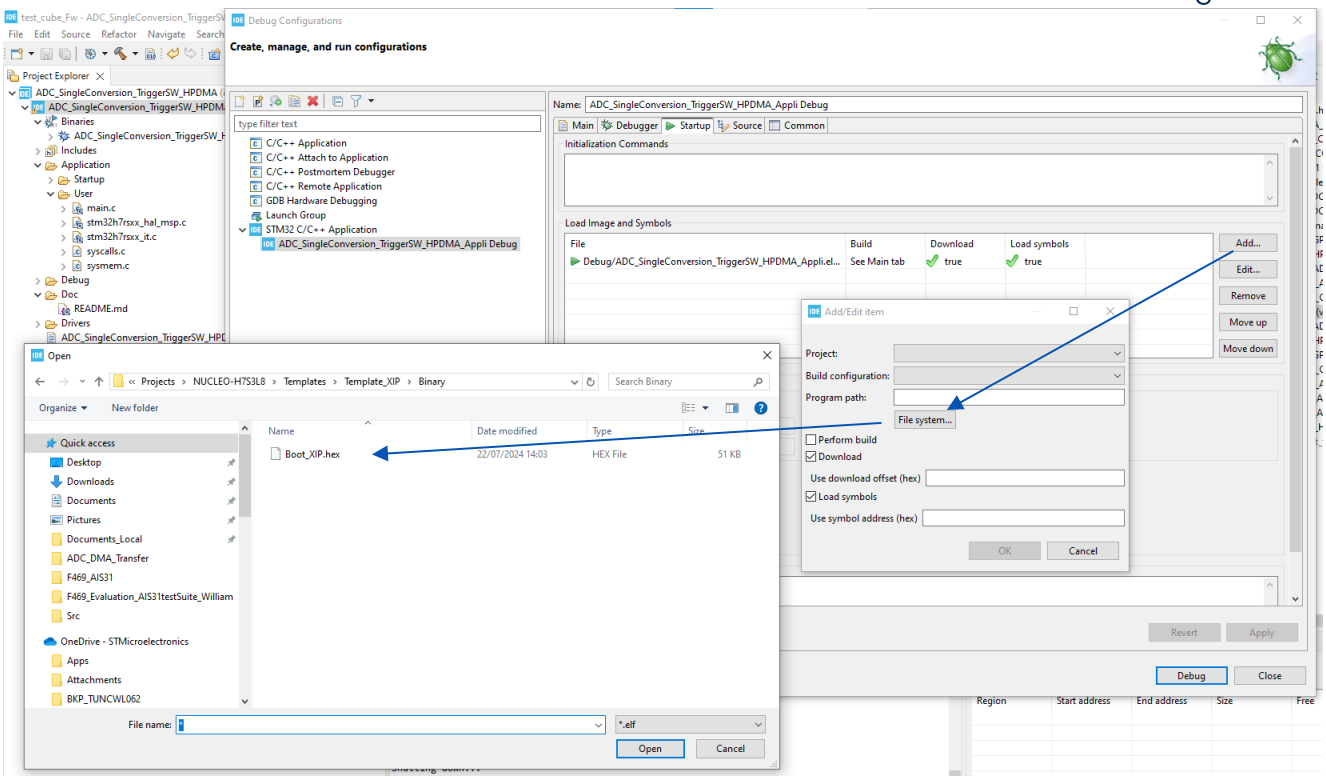
- Select then the added boot project and build/compile it first.
- Then click on add button and proceed as follow:



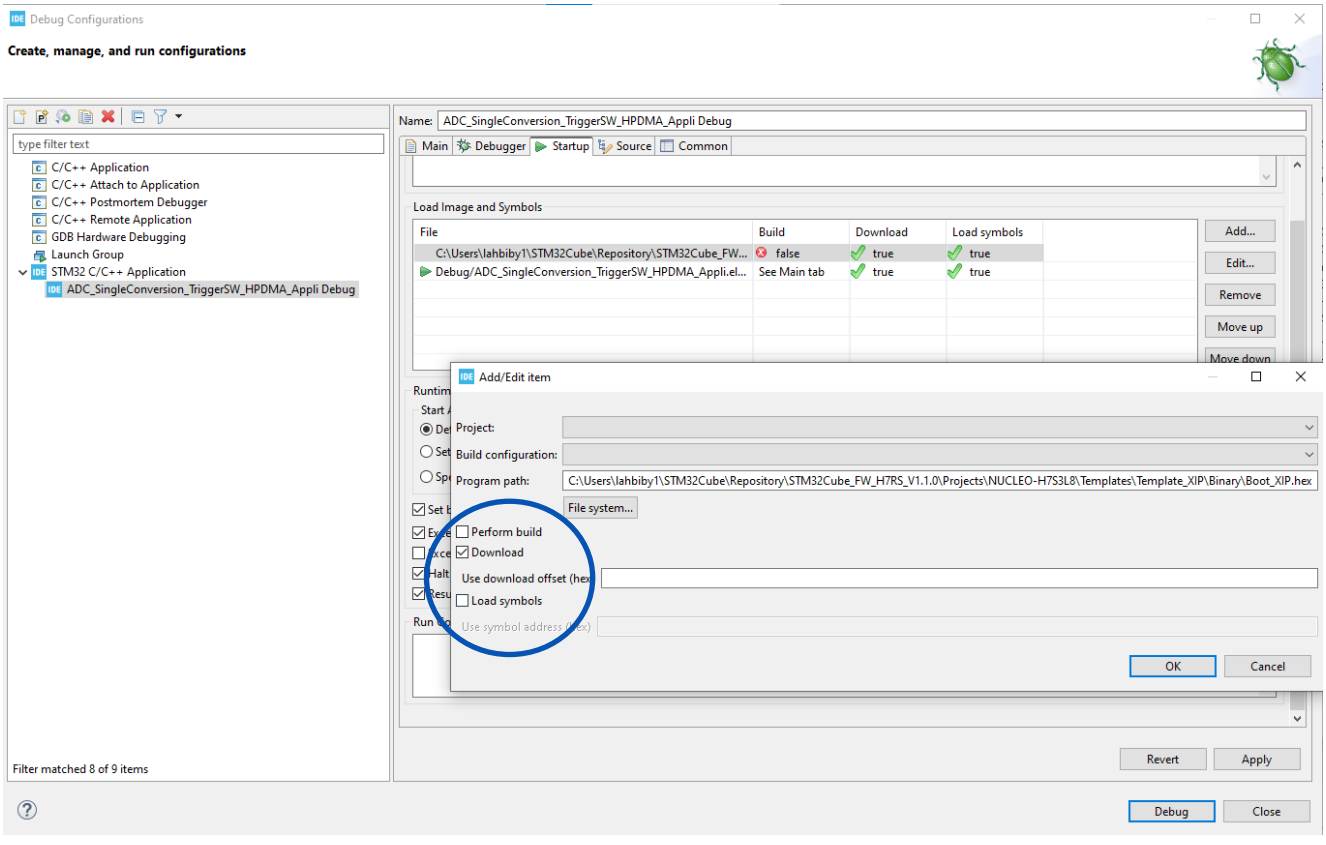


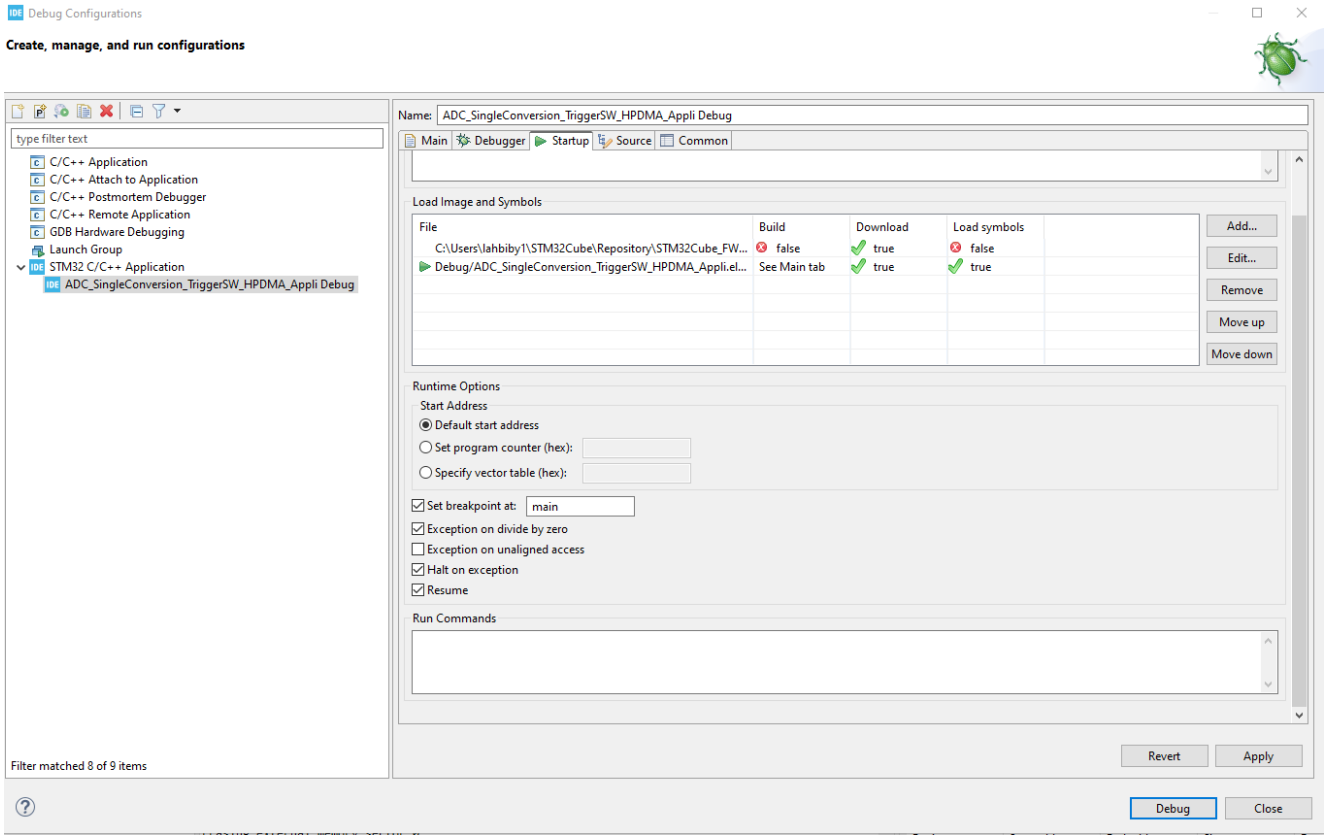
- If your project doesn't contain a boot project:
  - click on [File System] and select the Boot HEX file corresponding to your board

    Boot_XIP.hex can be found in folder [Binary] on each Template_XIP project. You may need to force the capability to select a .hex file by typing " * " + pressing the "Enter" key in the file name dialog



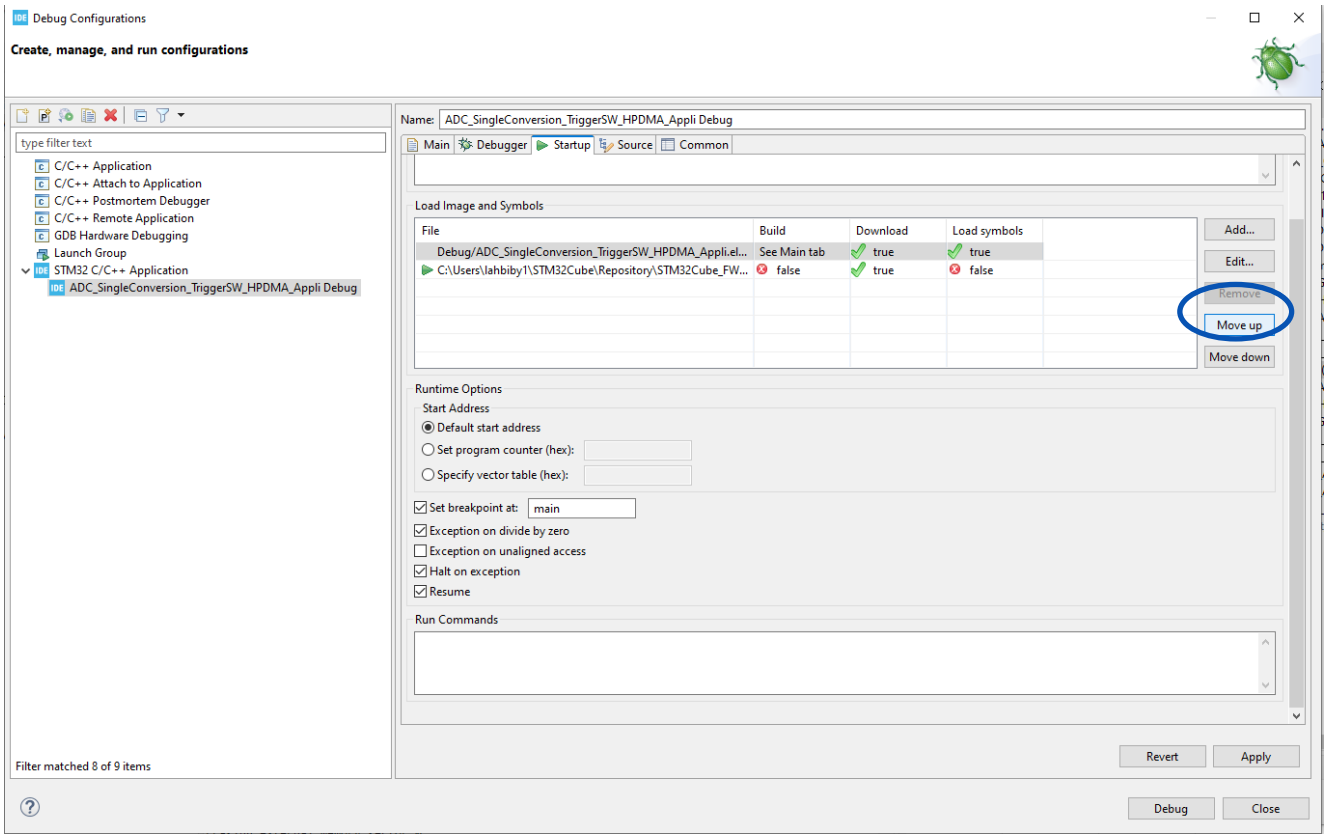
- then select the following options by double clicking on the hex file to change the options
  - “Download” checked.
  - “Load symbols” unchecked.
  - Click Ok





- Back in the in the [Startup] tab, move down the boot project for it to be in second position.

You can do this by selecting the Appli elf file and click on Move up button.



- Our debug configuration is ready to be used.

Application is executed from the flash external memory @0x70000000