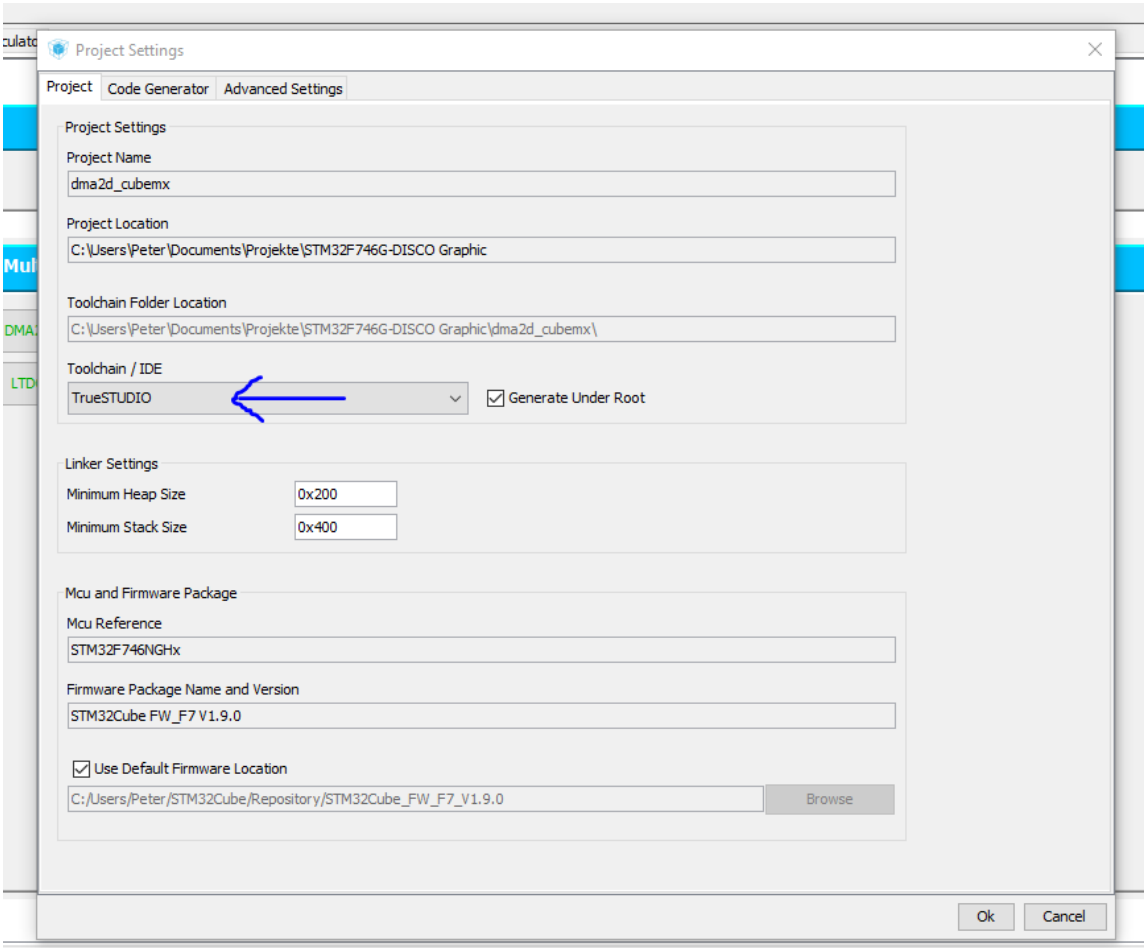
Manual for porting CubeMX Projects to EMBitz (Example for STM32F7)

The project name I used is “dma2d”, because I wanted to test some things with dma2d (just to understand in which directory you have to copy the files). Of course, you can use every project name you want. But I advise you to use the same name for the EmBitz project and the CubeMX project and just add a suffix like “_cubemx” as I did to the CubeMX project name.

This tutorial is long, but it does not mean, that it is complex. This manual is NOT complex it is only very detailed. If you have done these steps one time, it is done in several minutes the second time.

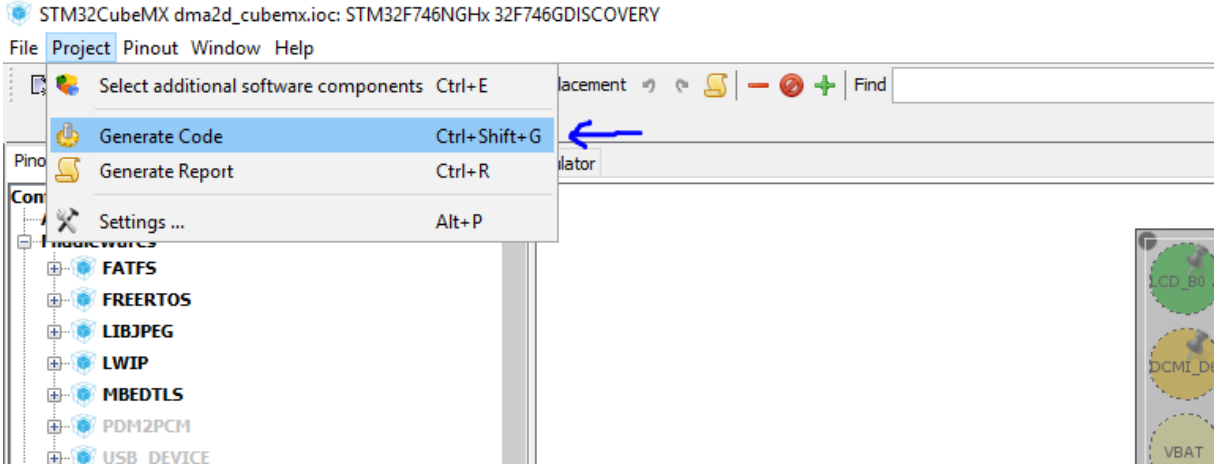
1 CubeMX Configuration

After complete configuration of CubeMX go to project settings and select TrueSTUDIO as Toolchain



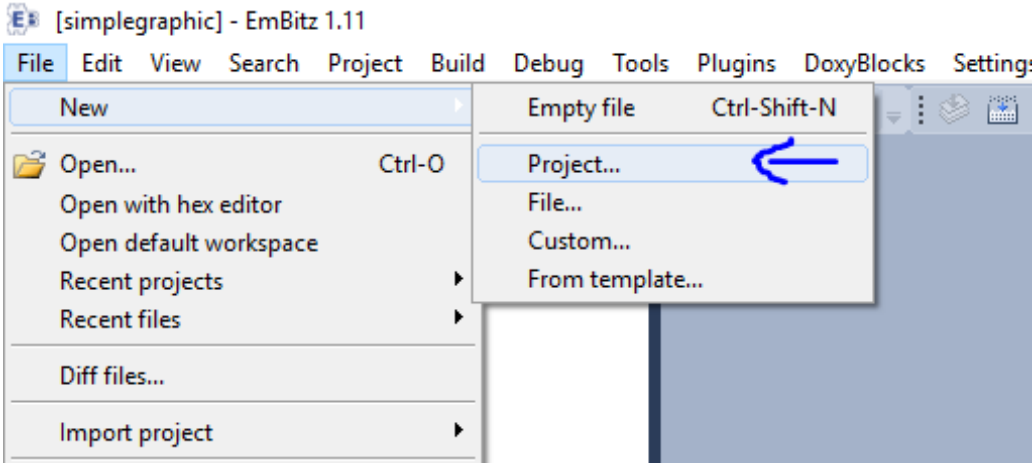
2 Generate project

I prefer to choose the project name for the CubeMX project the same as the EMBitz project name with "_cubemx"-suffix. I also keep generating the report, because it is perfect for developing the PCB for example.

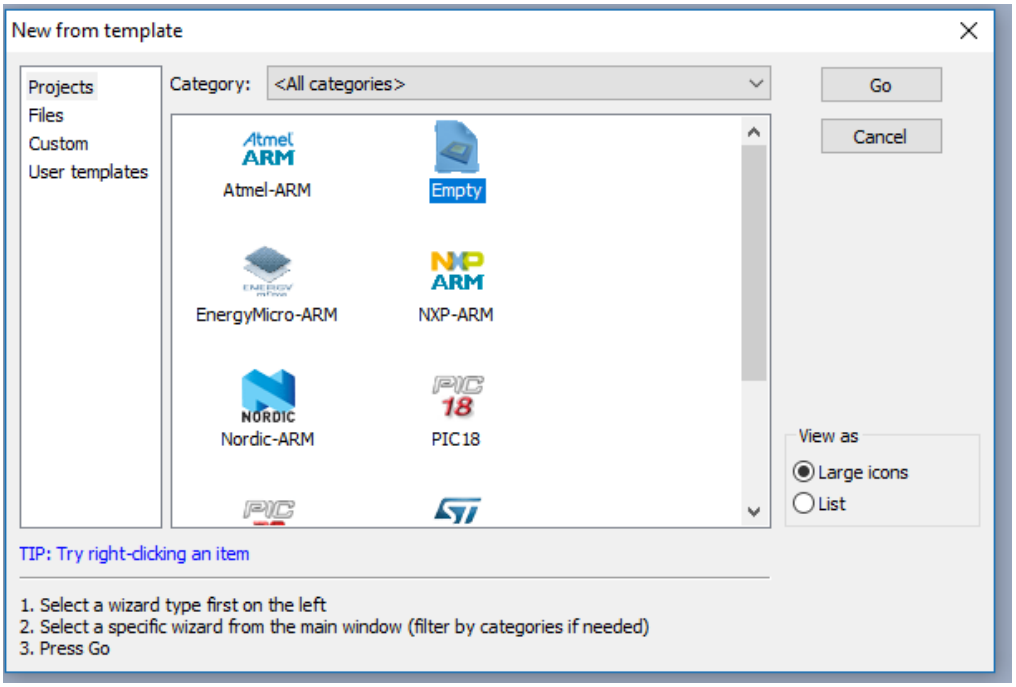


3 Open EmBitz and create new empty project

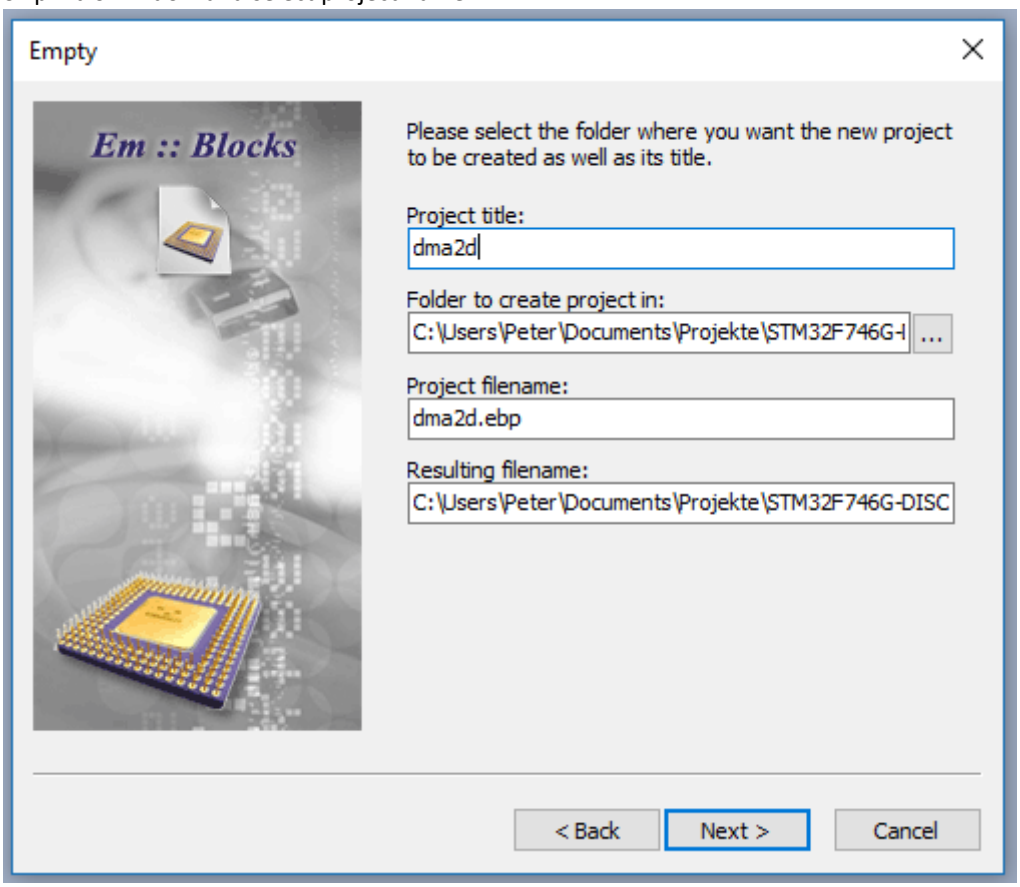
Select new project



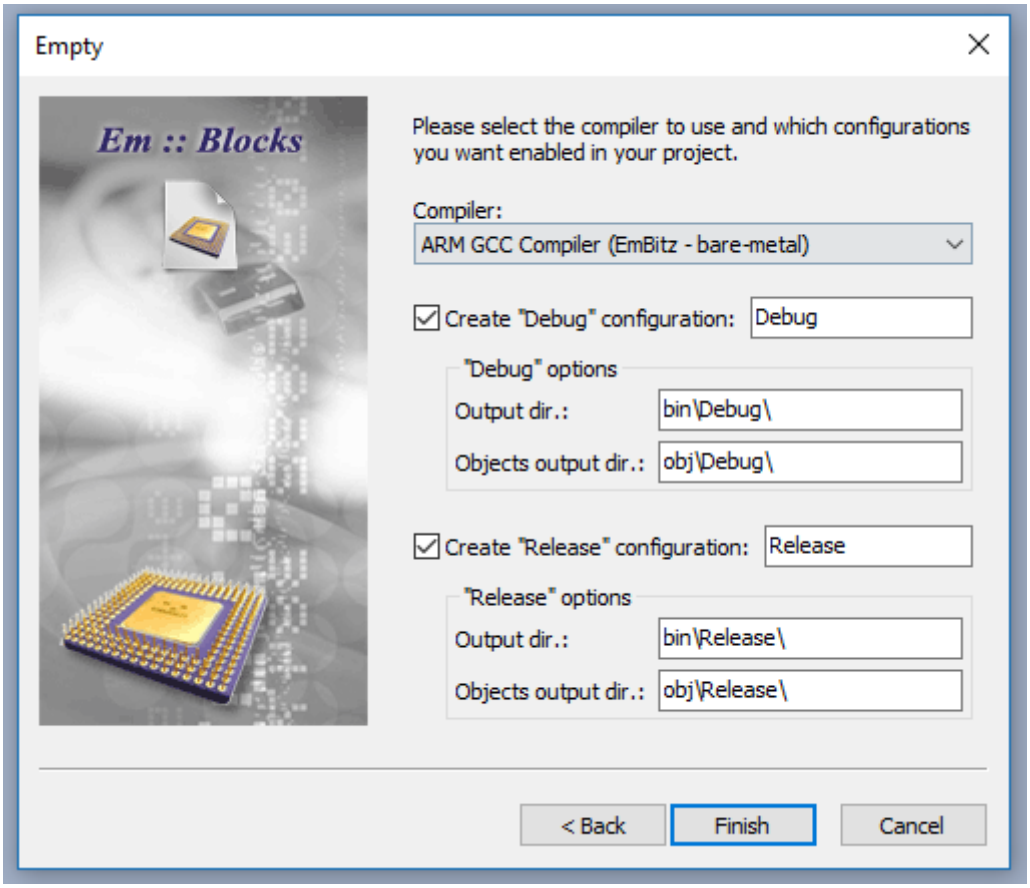
Click empty and Go



Skip intro window and select project name

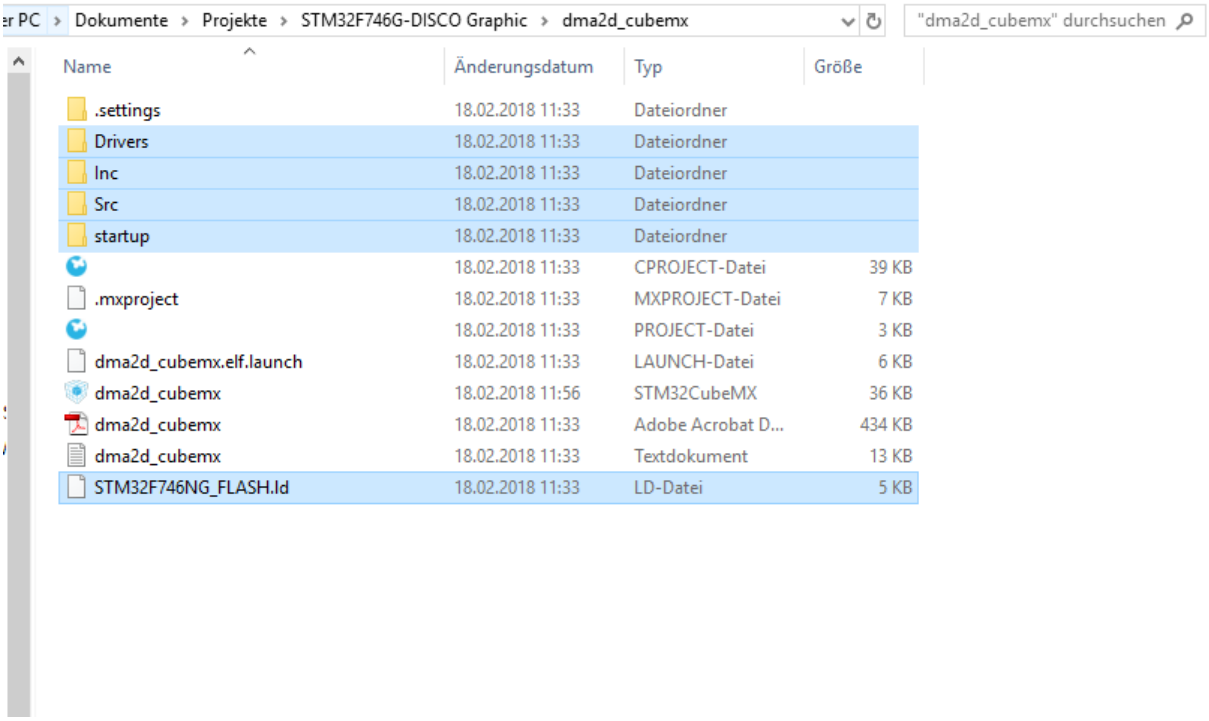


Leave settings as they are and click Finish

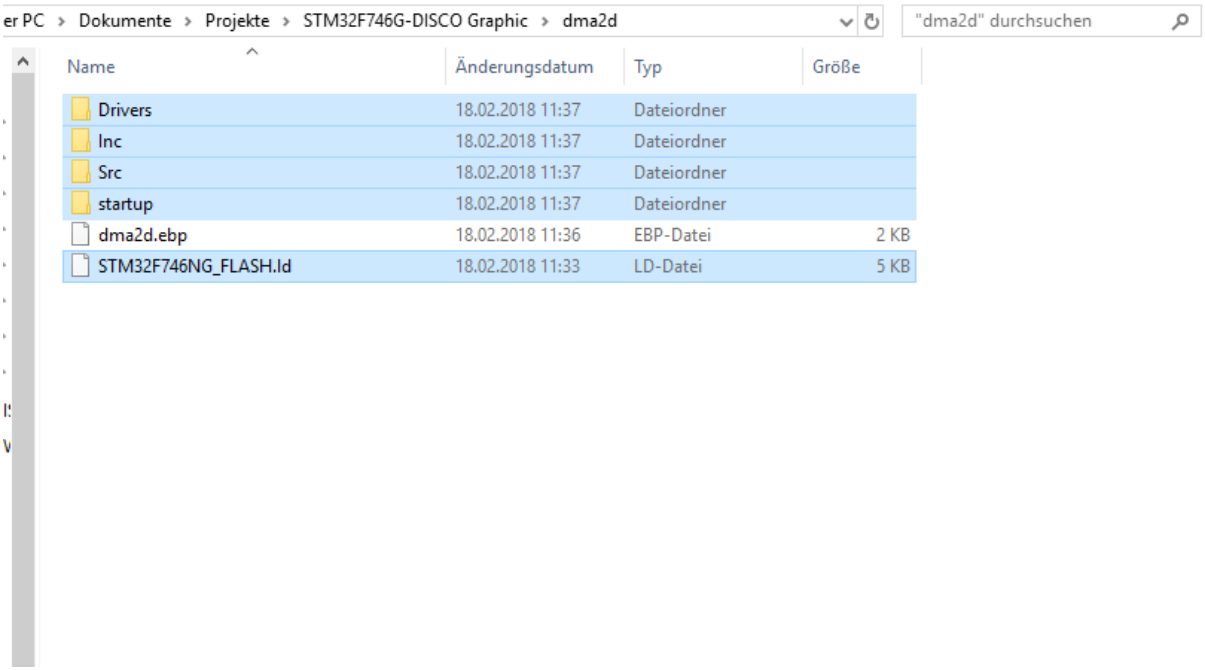


4 Copy CubeMX generated files in EmBitz project folder

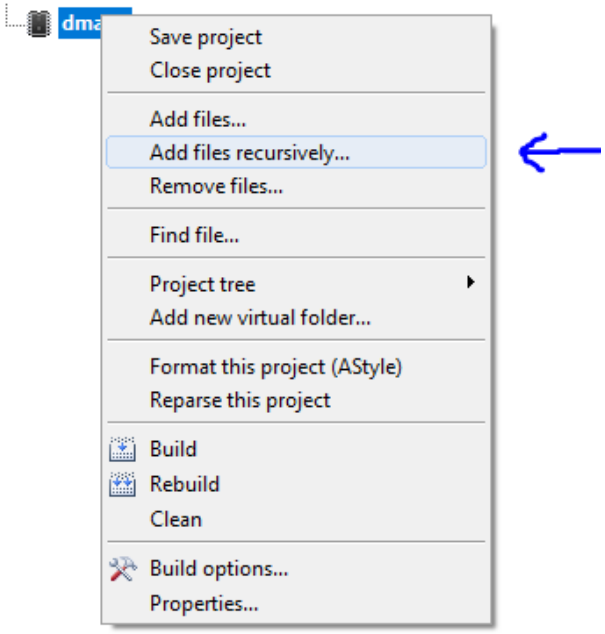
Copy these 5 selected folders/files from the CubeMX folder…



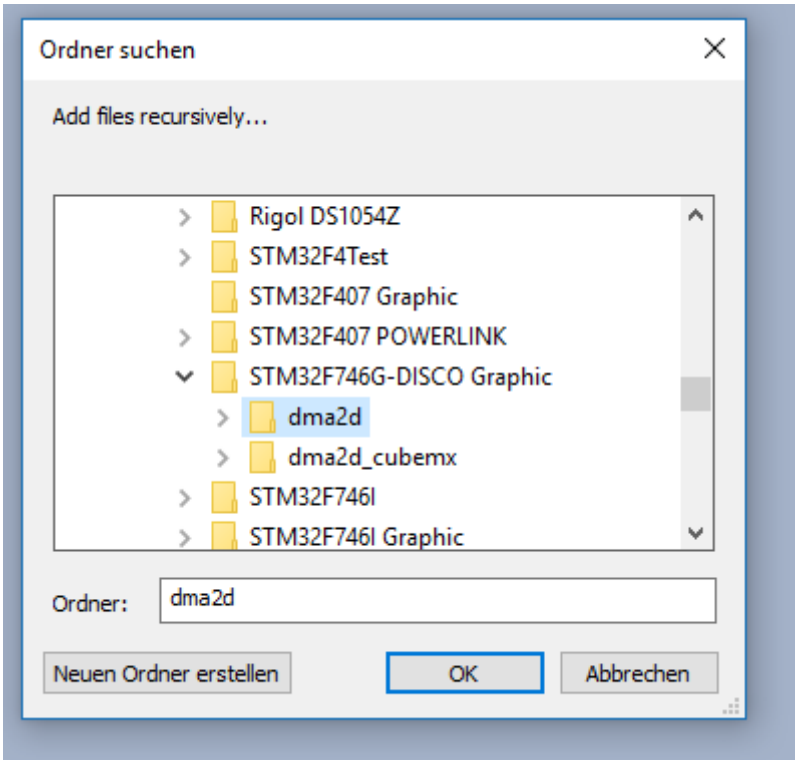
into your EmBitz project folder



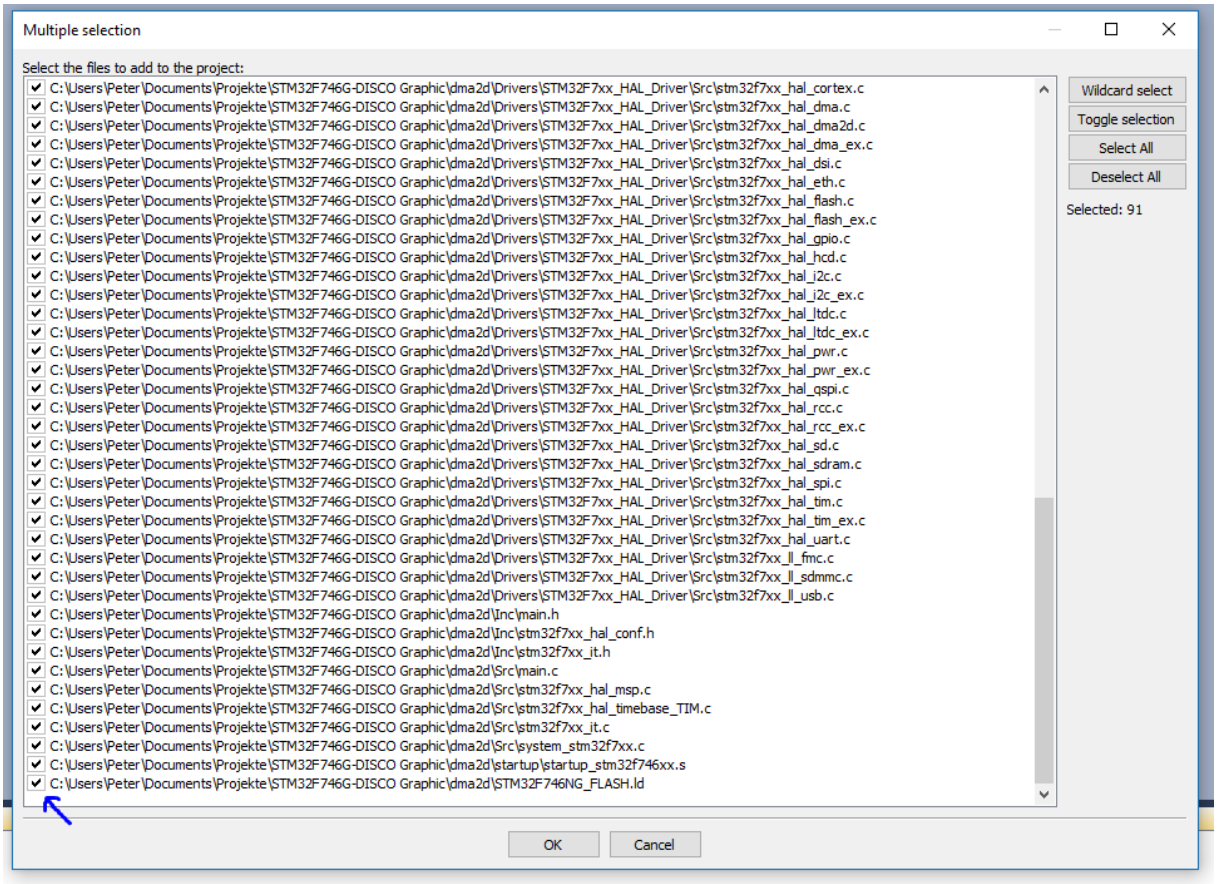
Then right-click your project in EmBitz and “Add files recursively”



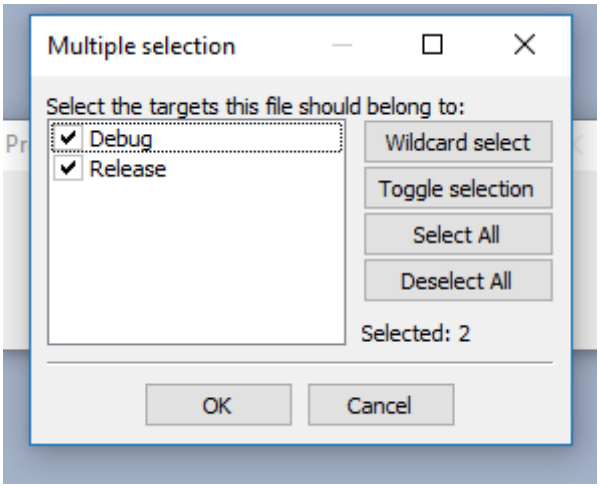
Then click OK, if your EmBitz project folder, where you put the files in, is selected



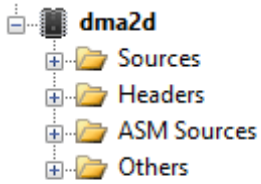
In the selection window make sure everything is selected, especially the last one “STM32XXXXXX_FLASH.ld”, which is not automatically selected and click OK



Select targets and click OK

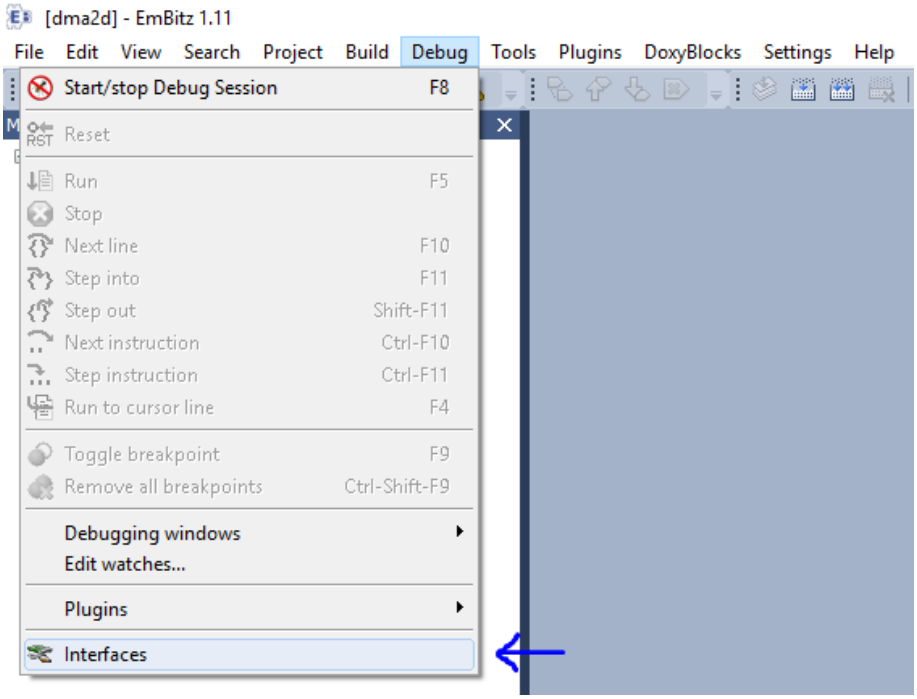


Files are now added to your workspace. For further instructions, make always sure, that this project is selected (bold fonts)

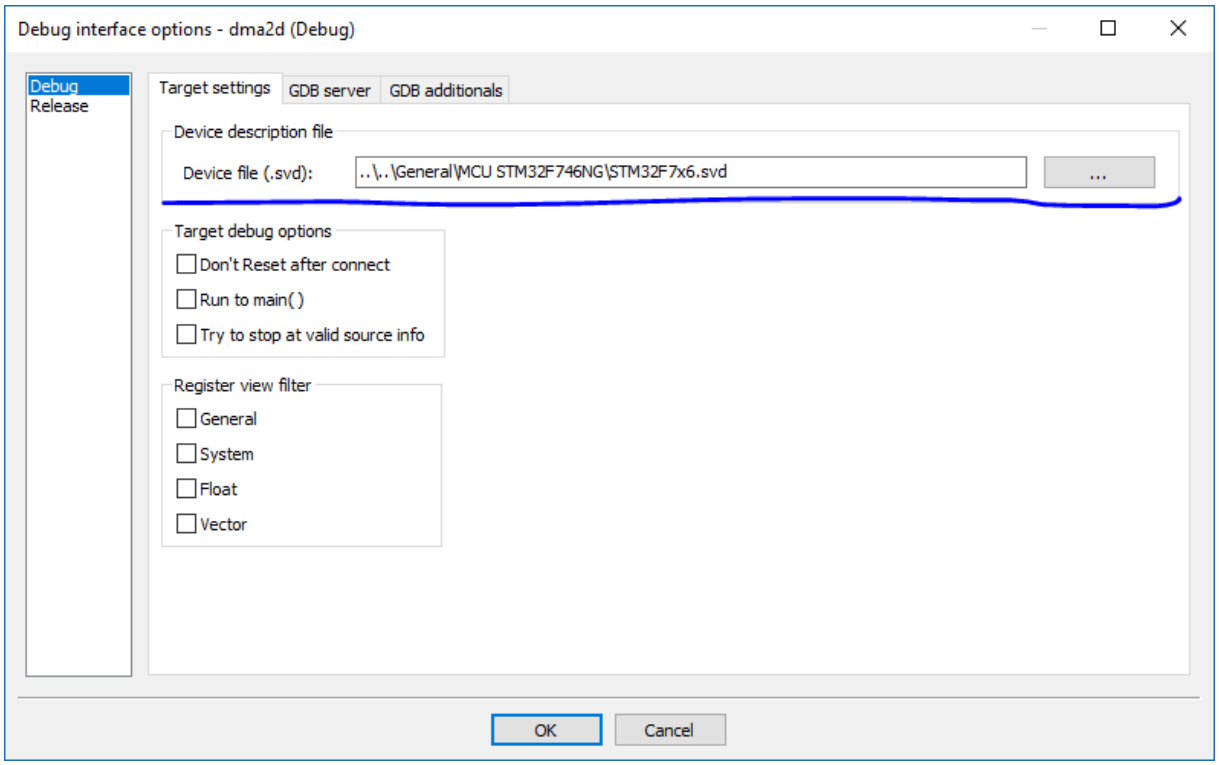


5 Set interface options

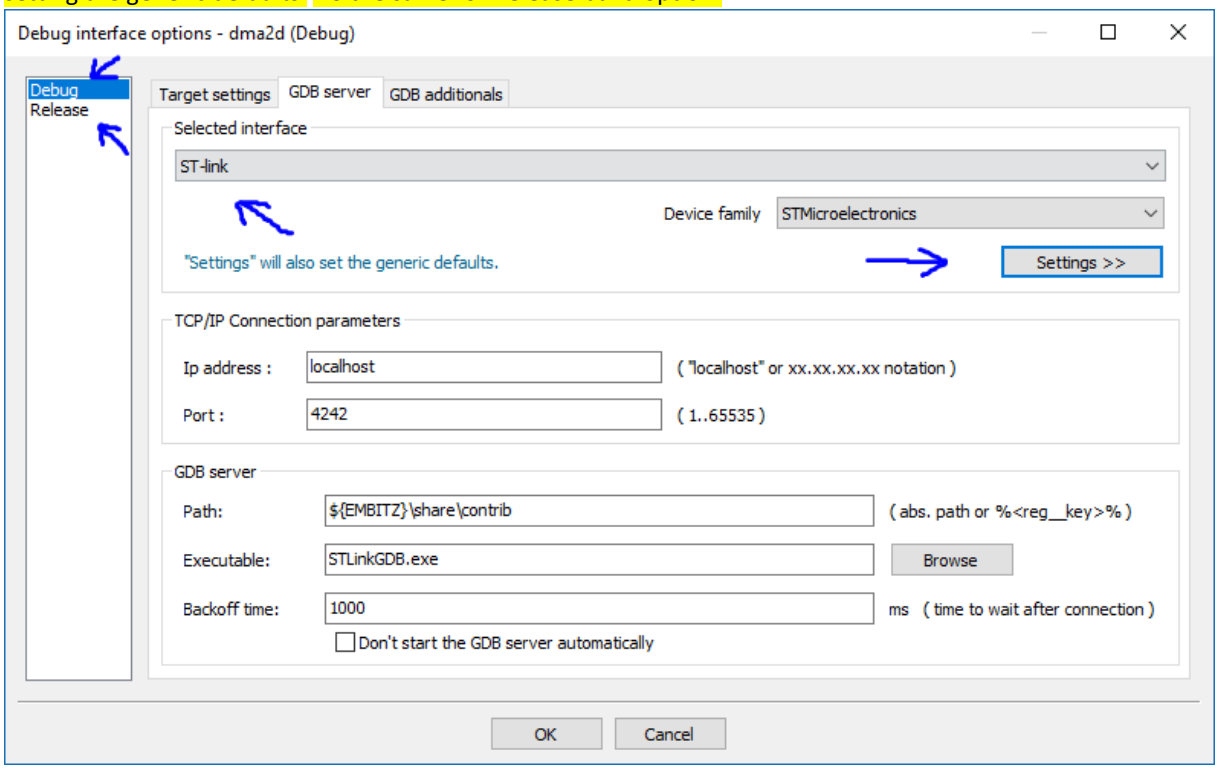
Go to Debug->Interfaces



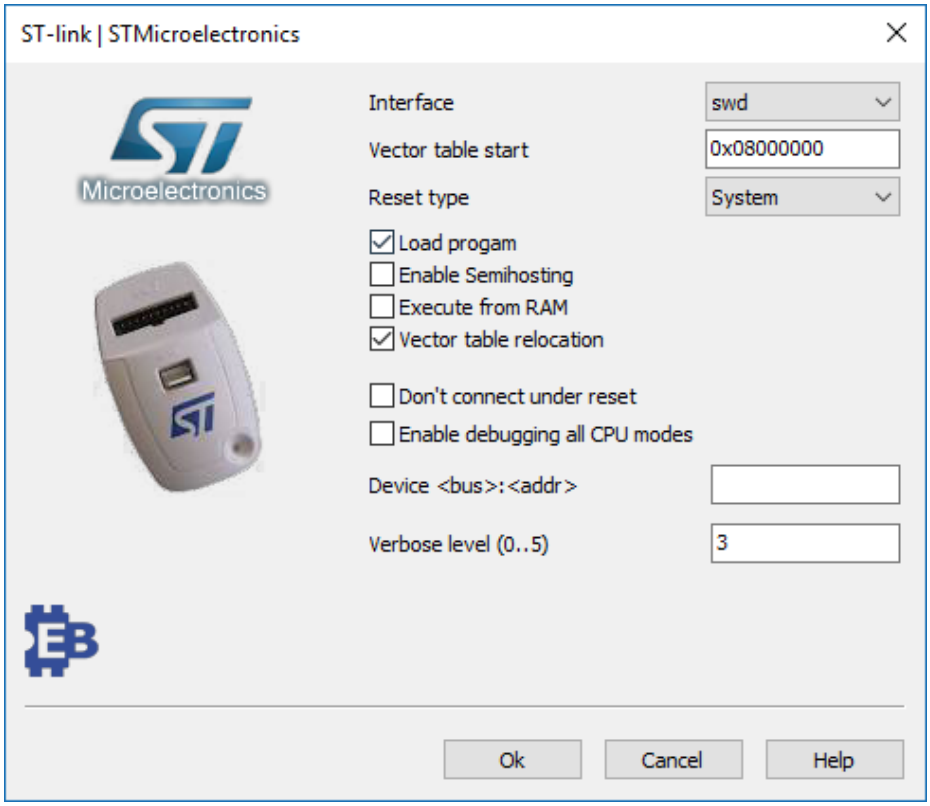
1 Add the device specific System View Description file (.svd), available at http://www.st.com at the specific product page, for watching the registers while debugging. You don't need to add .svd for release build option, because there is no need to watch the registers.



2 Select your debugging interface (mine is ST-link V2). You have to click Settings and then click OK, for setting the generic defaults. Do the same for Release build option.



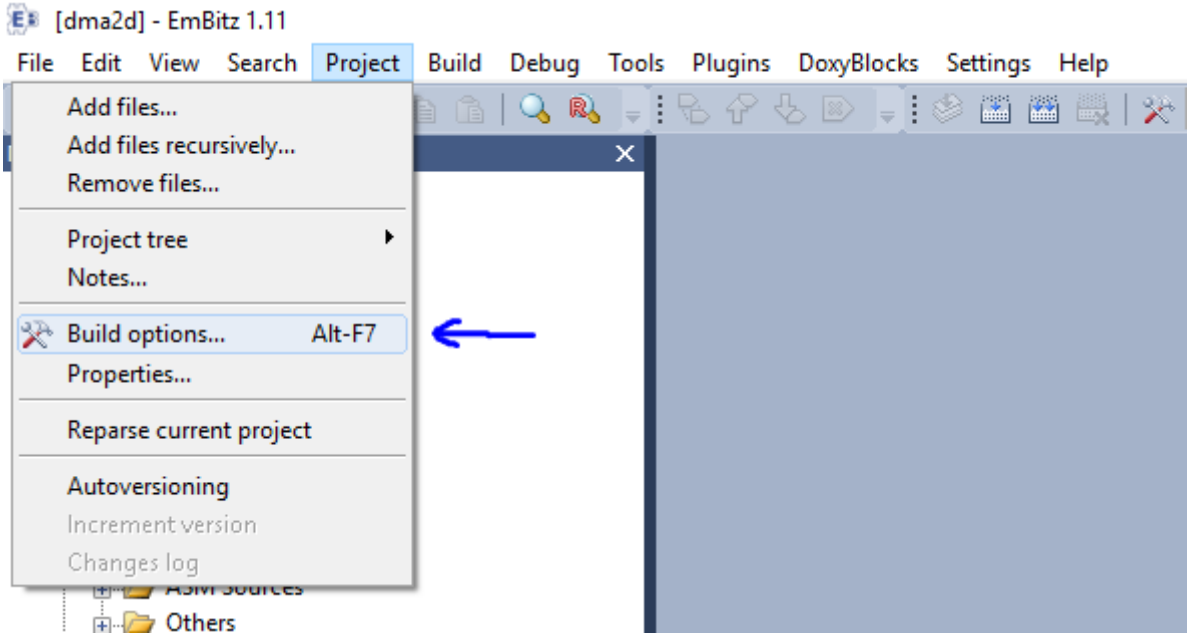
Leave default values and just set your debugging standard (swd/jtag)



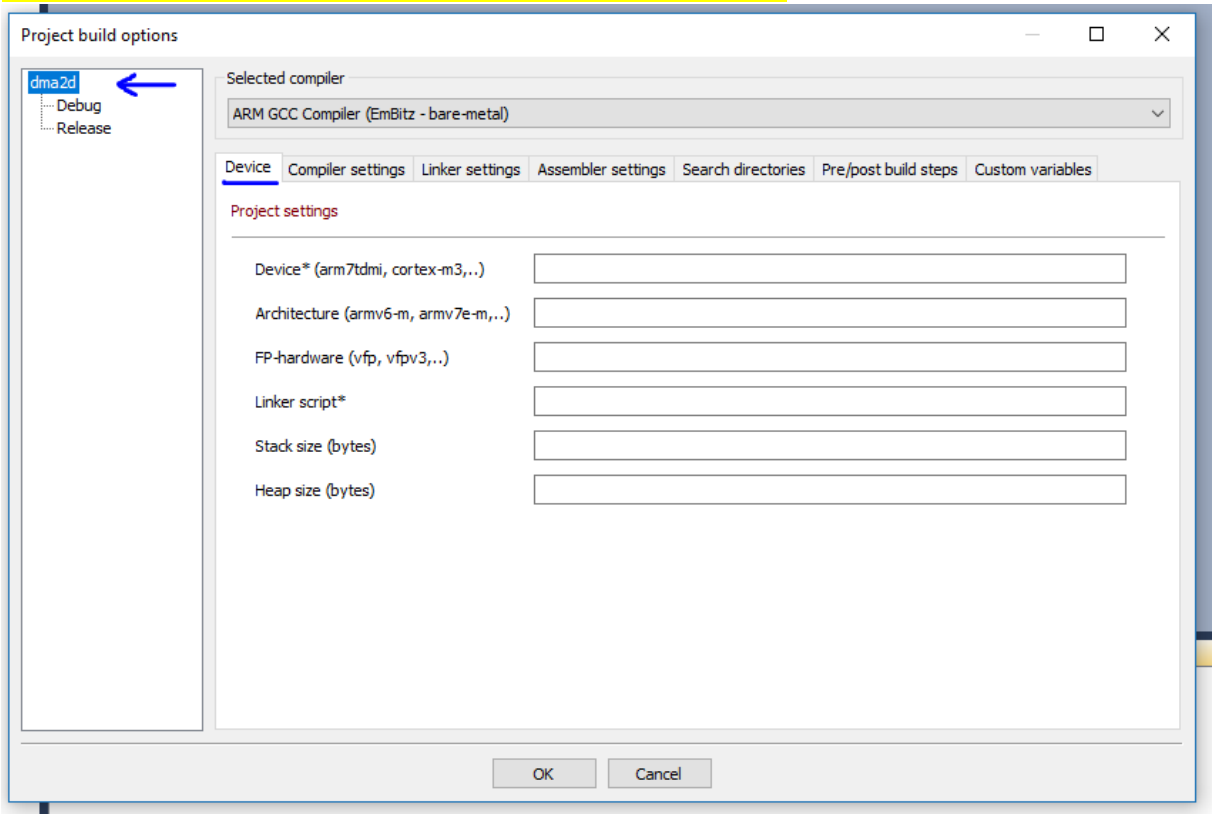
Accept changes with OK in both windows to close debug interface options

6 Set Compiler settings

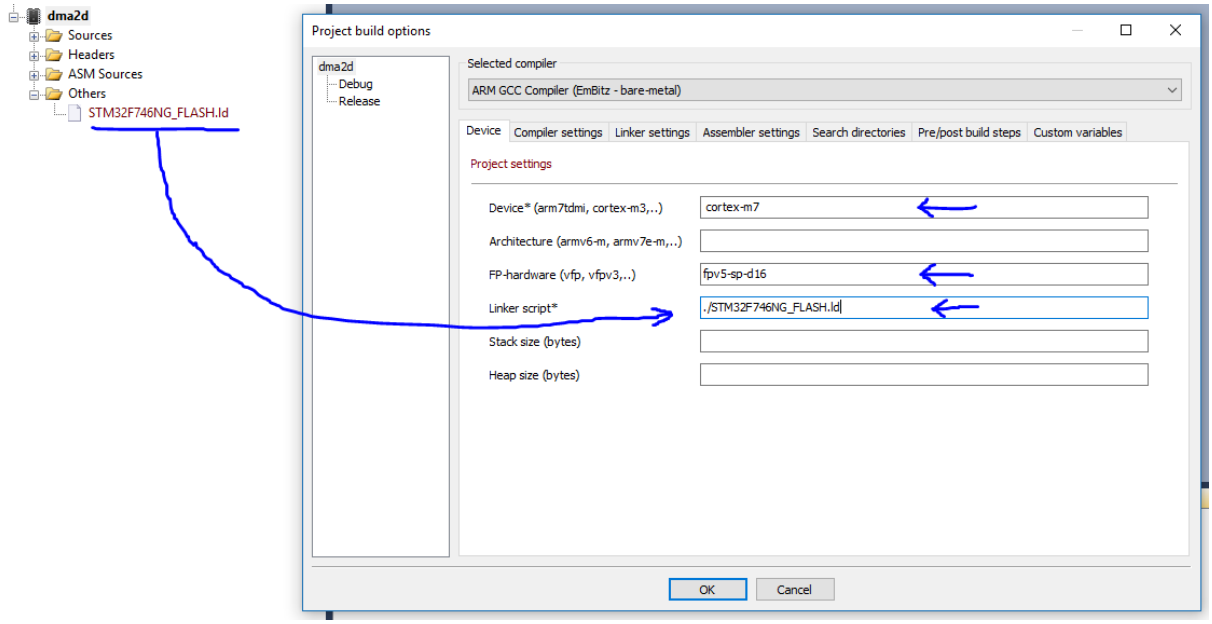
Open Build options



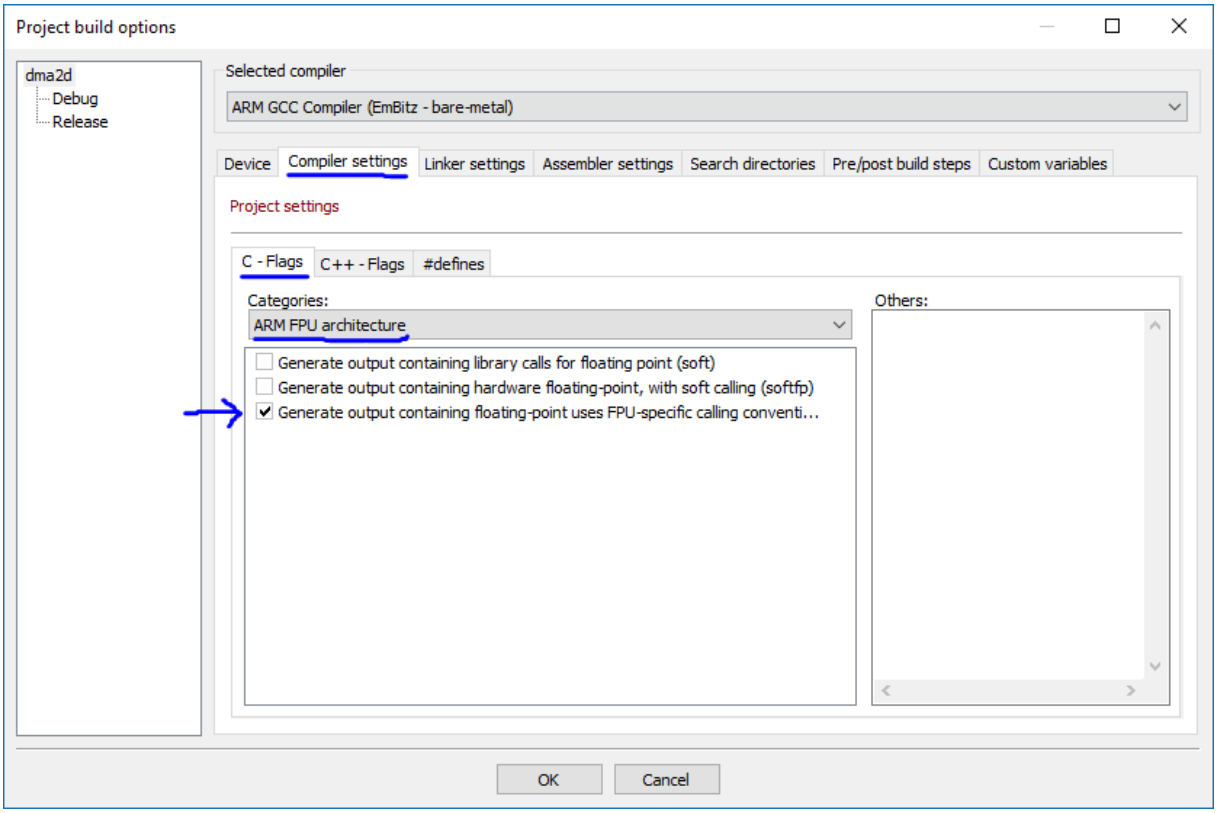
First, select the project name, above Debug, then click Device and fill in the settings. These settings ,Filled in by me, are for Cortex-M7 MCUs, so make sure to fill in different entries for other MCUs. Under Linker script, you have to choose the linker script generated by CubeMX, like in the photo.



Under the point linker script, you have to fill in “./” + the name of the generated linker script file, as done in the photo.

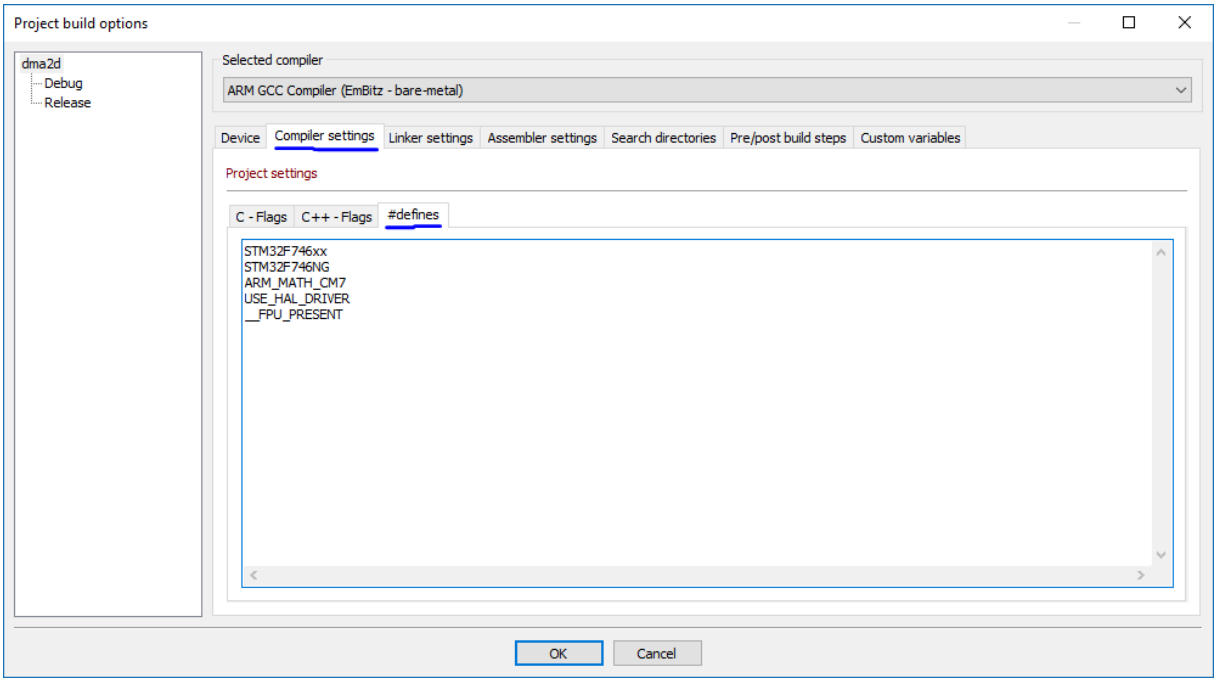


Check the following checkbox

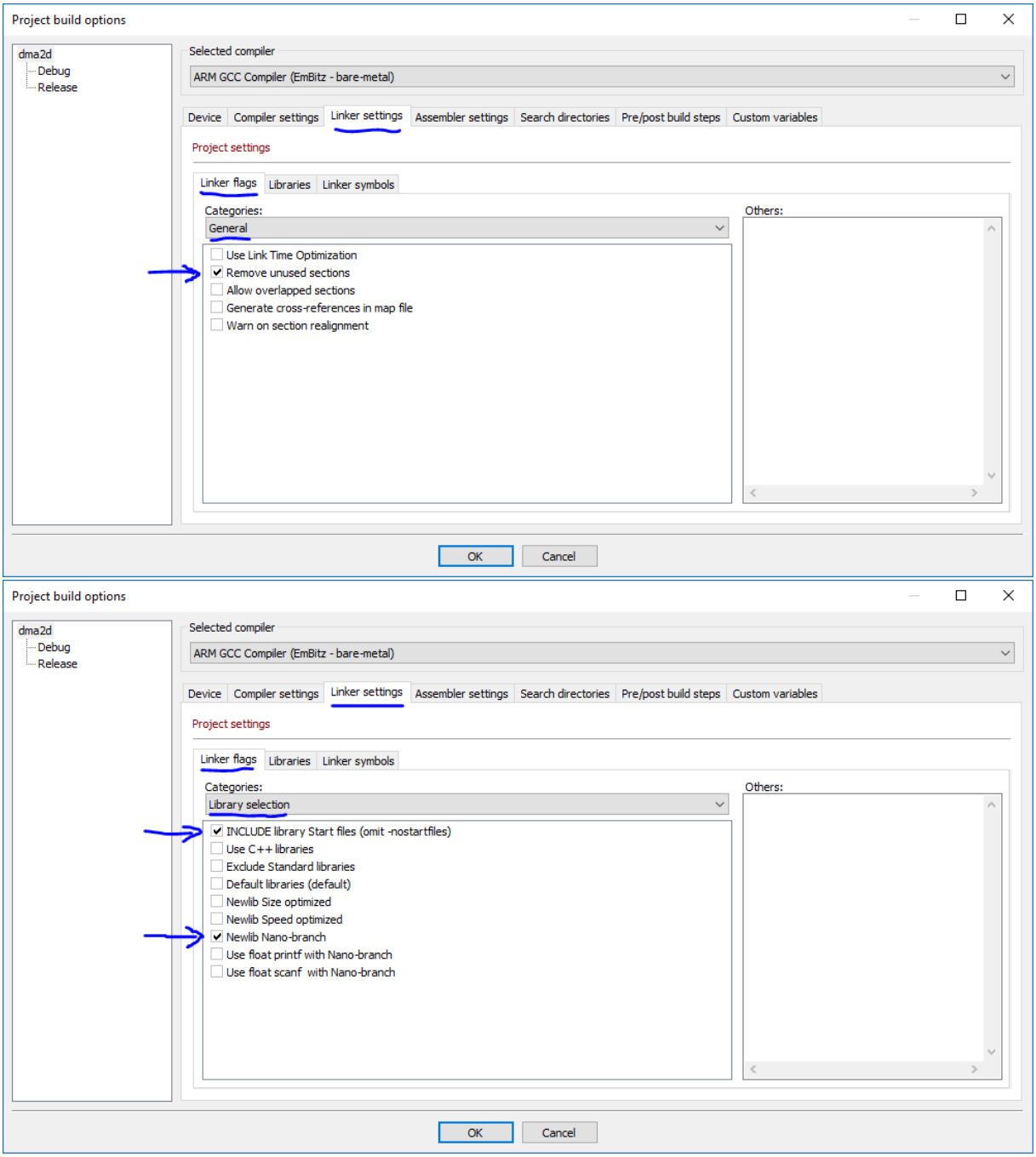


Write the following, device specific expressions under #defines. Adapt the terms for your MCU.

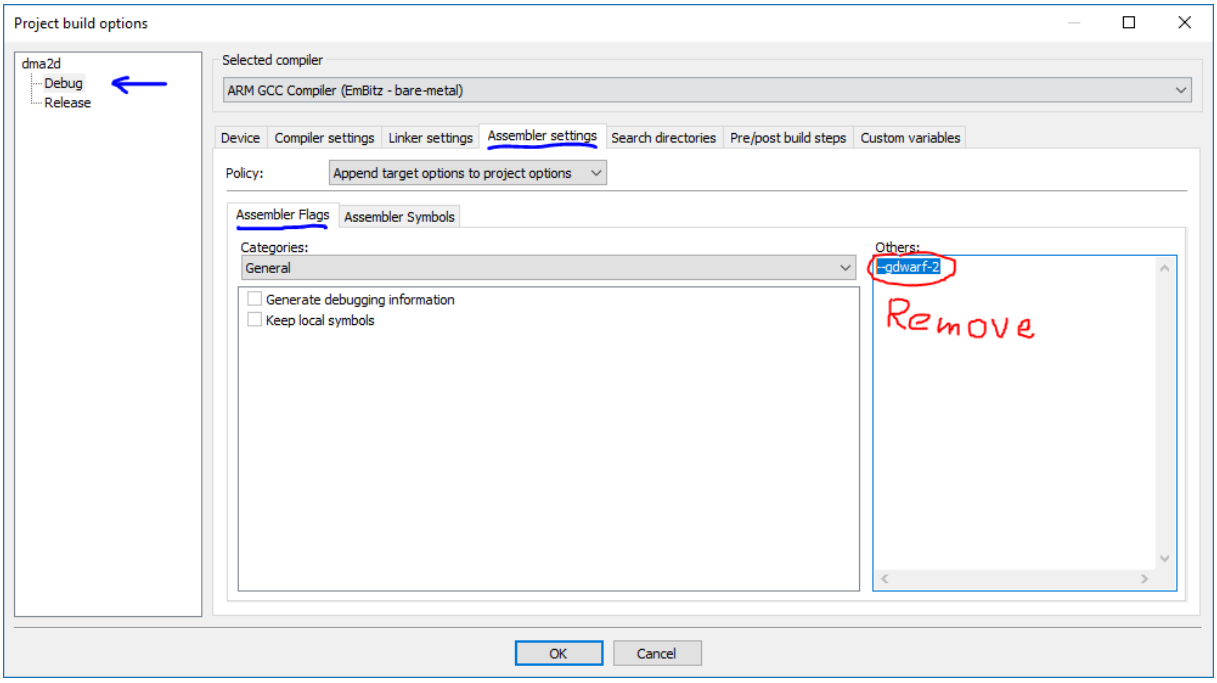
```
STM32F746xx
STM32F746ZG
ARM_MATH_CM7
USE_HAL_DRIVER
__FPU_PRESENT
```



Check the following checkboxes



Remove “—gdwarf-s“ in the “Others“ window under Debug section



7 Finished

This tutorial is tested by myself, but it cannot be guaranteed, that it is working.

Please feel free to contact me for any suggestions and improvements.

If you need help with this tutorial, or need to adapt this one to another MCU, please contact me.

8 License