



life.augmented

STM32 AZURE RTOS workshop

Tools installation – Check environment

- Purpose :
Guideline to **install tools and material** for the STM32 AZURE RTOS Workshop
Check of the **prerequisite** (knowledge and **homework**)
- Materials provided :
STM32-AZURE-RTOS-WS-material.exe
 - > STM32SecuWS material (Projects, source files)
 - > STM32CubeH7 and X-CUBE-AZRTOS
 - > TraceX Installation tool
 - > Slides
- Environment supported : PC laptop with Windows 10

Agenda

- Step 1 : Install tools and check configuration
- Step 2 : Install workshop material
- Step 3 : Install packs
- Step 4 : Check installation through an example

If you face any issue during the setup please describe it here:

<https://community.st.com/stm32-azure-rtos-workshop>

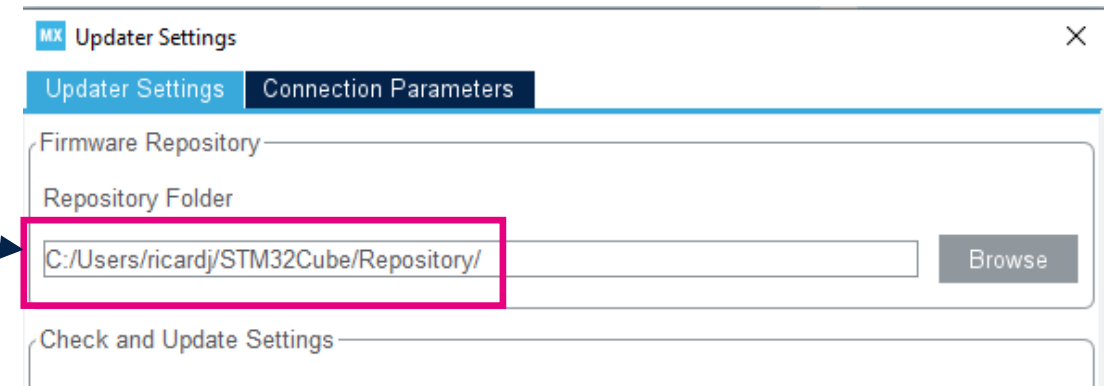
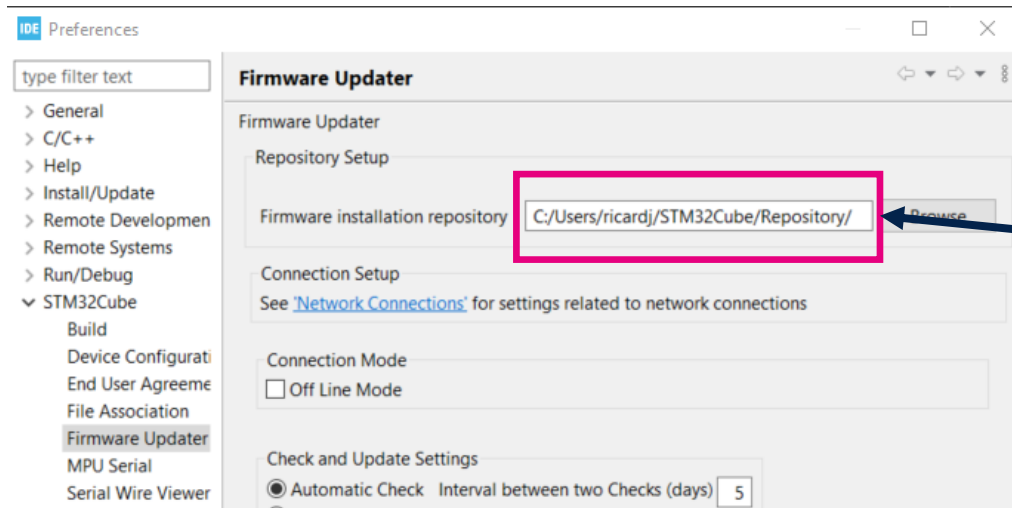
We will try to help you !

Step 1 : Tools installation

- Get and install STM32CubeIDE 1.7.0 from st.com
 - <https://www.st.com/en/development-tools/stm32cubeide.html>
- Installation guide for STM32CubeIDE
 - https://www.st.com/resource/en/user_manual/dm00603964-stm32cubeide-installation-guide-stmicroelectronics.pdf
- Get STM32CubeMX 6.3 from st.com
 - <https://www.st.com/en/development-tools/stm32cubemx.html>
- Installation details for STM32CubeMX provided in
 - https://www.st.com/resource/en/user_manual/dm00104712-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf

Check tools configuration

- In case, you have already installed these tools, please ensure package repository is stored at **the same location** for both STM32CubeIDE and STM32CubeMX.
- STM32CubeIDE
 - Select Menu Window/Preferences
 - And then STM32Cube/FirmwareUpdate
- STM32CubeMX
 - Select Menu Help/Updater Settings



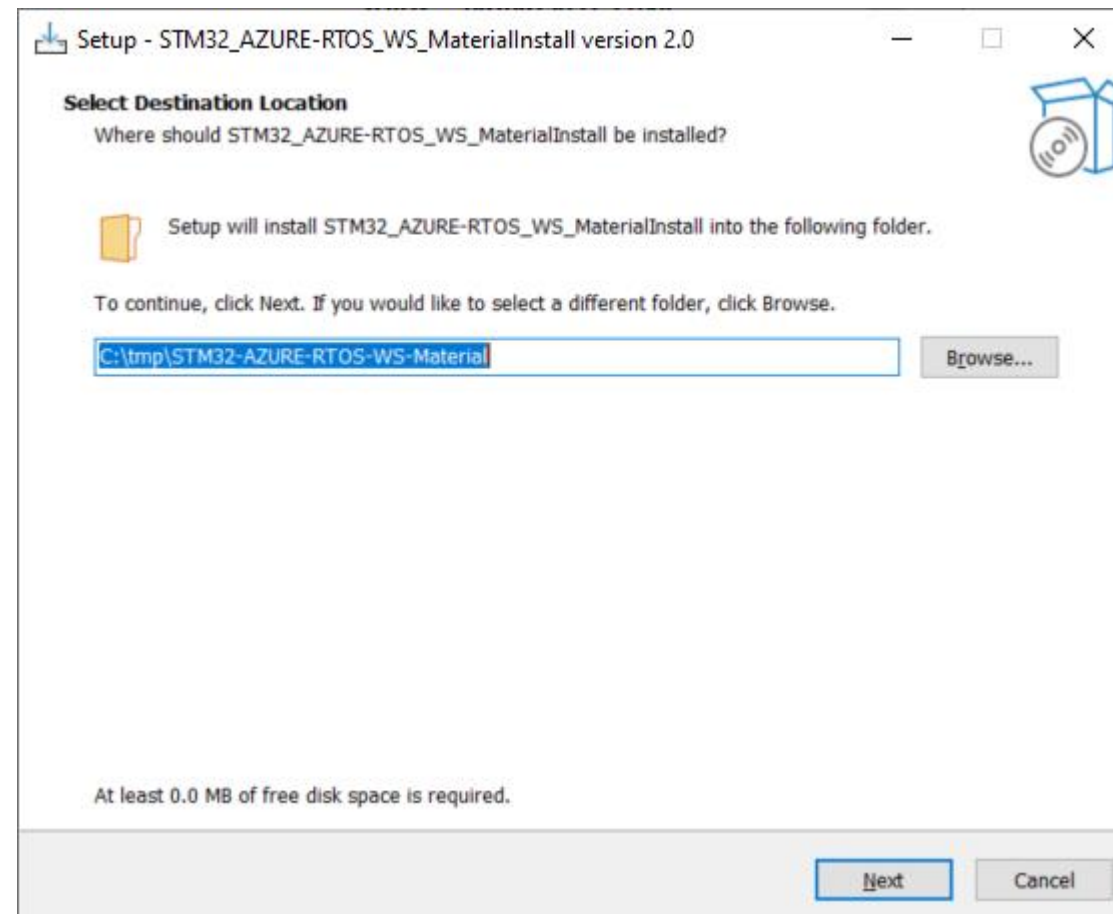
Step 2 : Workshop material installation

Get workshop material

- Workshop material is provided as a windows executable allowing selection of components to be installed.
- Executable name is STM32-AZURE-RTOS-WS-material.exe
- It is available in google drive at following address
 - https://drive.google.com/drive/folders/16_M2C8ZxRTmknrtYsOwdDvK7vM9NrRys

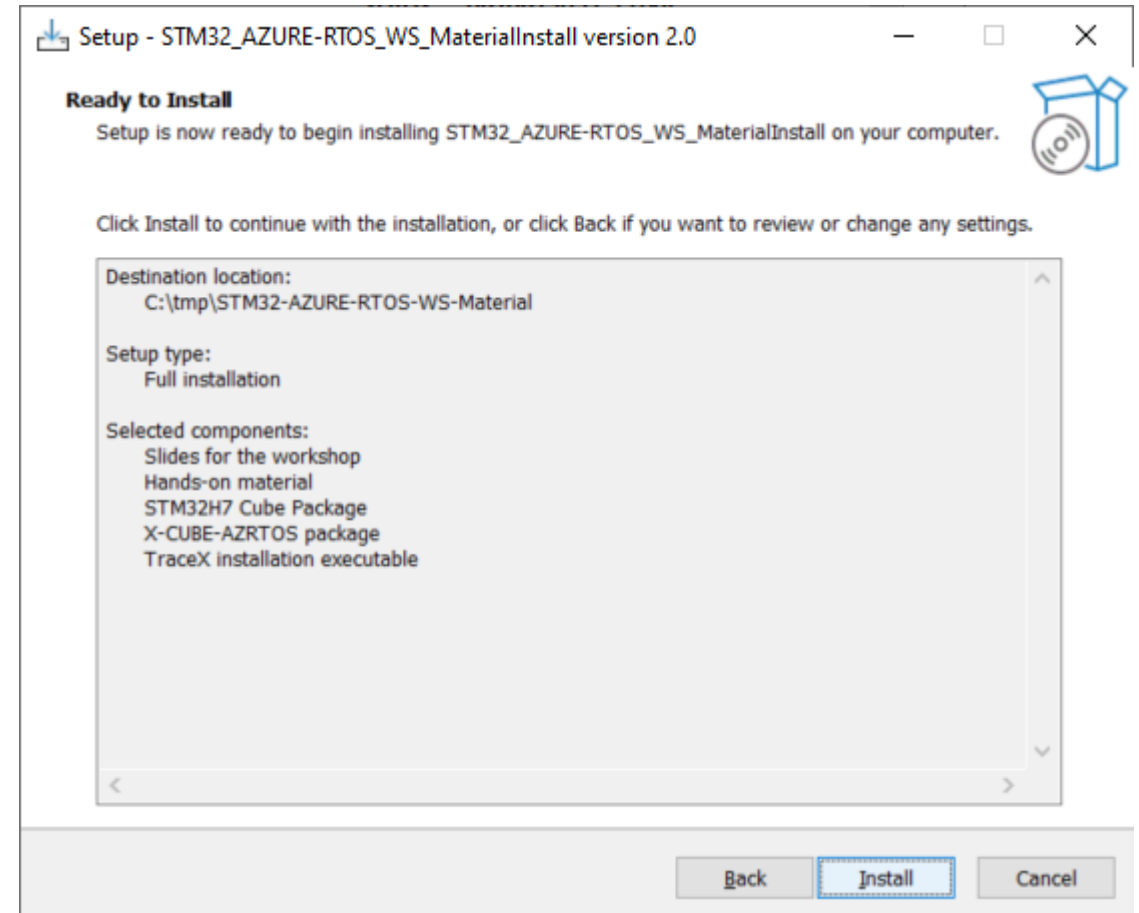
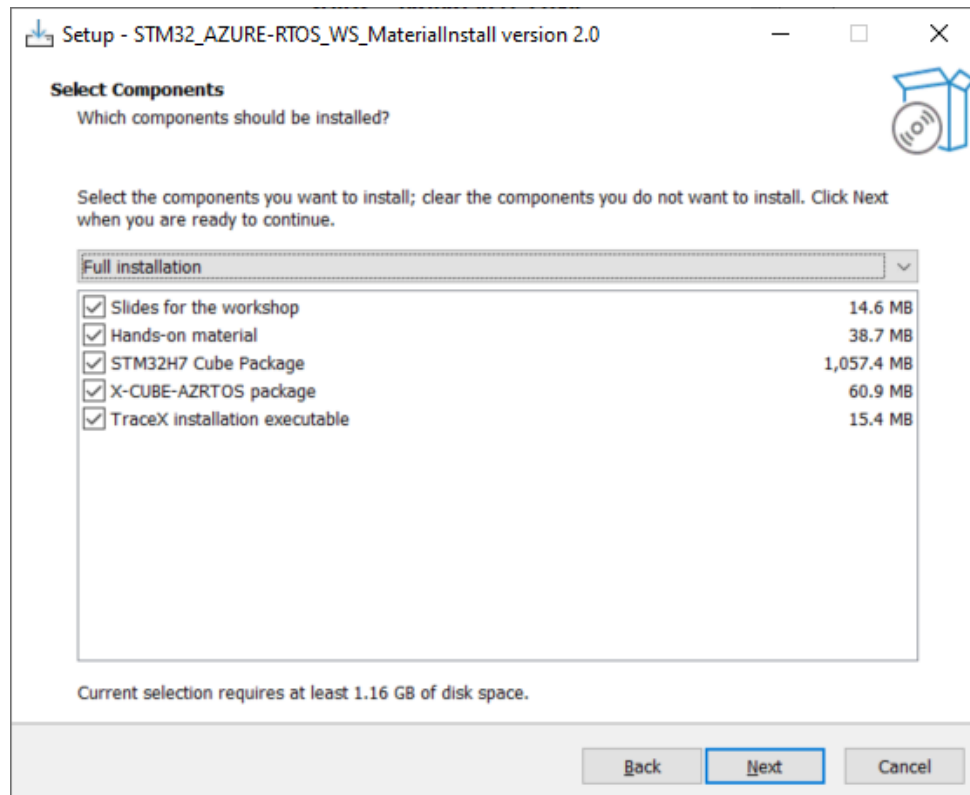
Launch workshop material installer

- Please launch: STM32-AZURE-RTOS-WS-material.exe

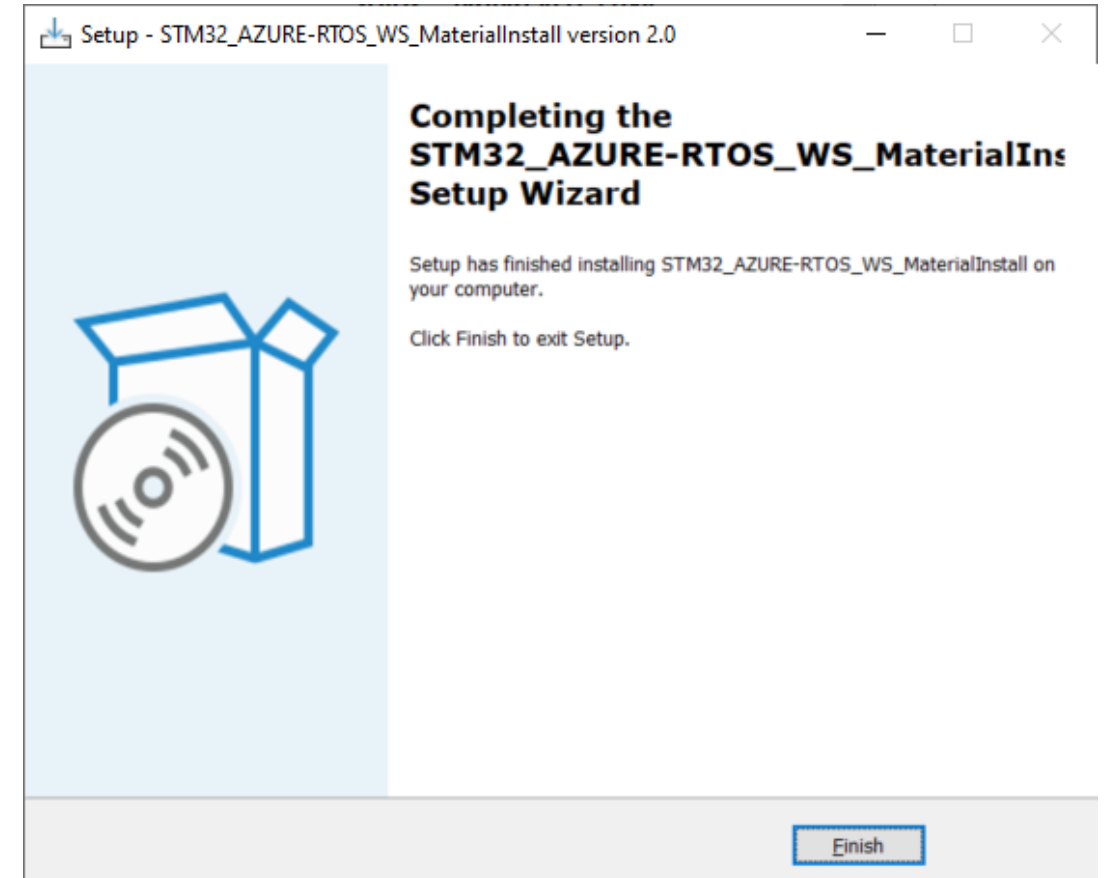
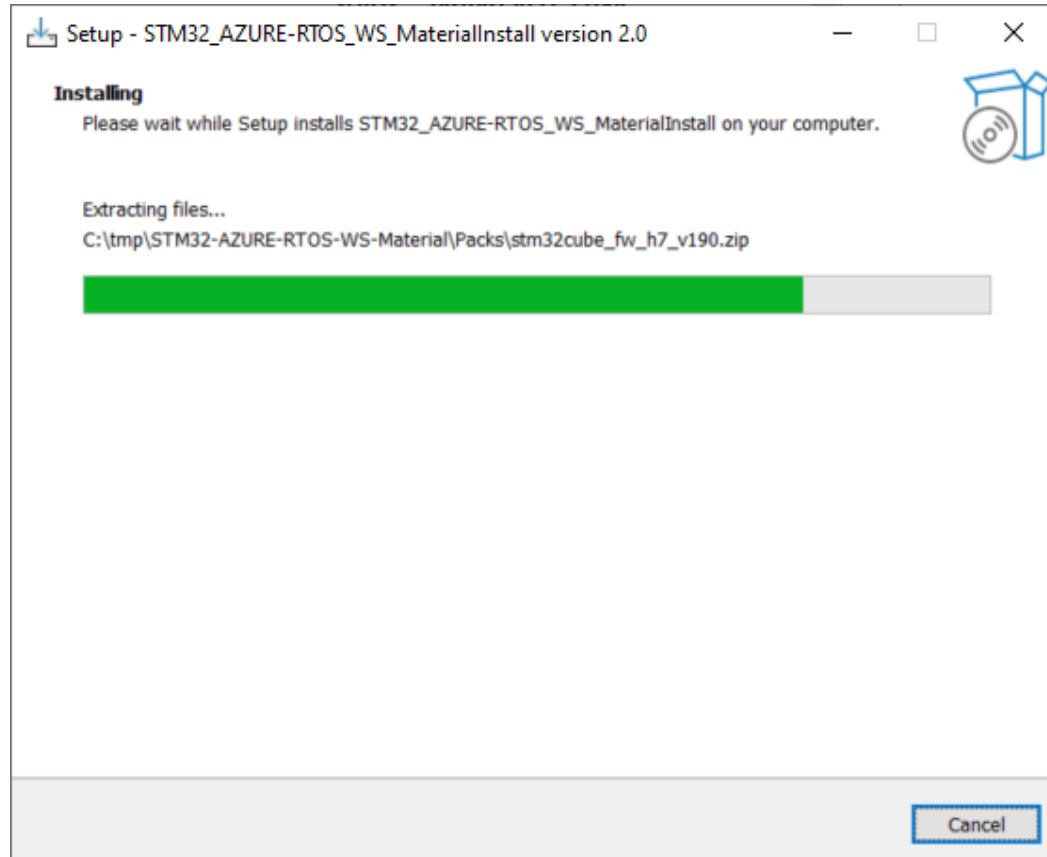


Workshop material component selection

- Select the components you want to install



Workshop material extraction



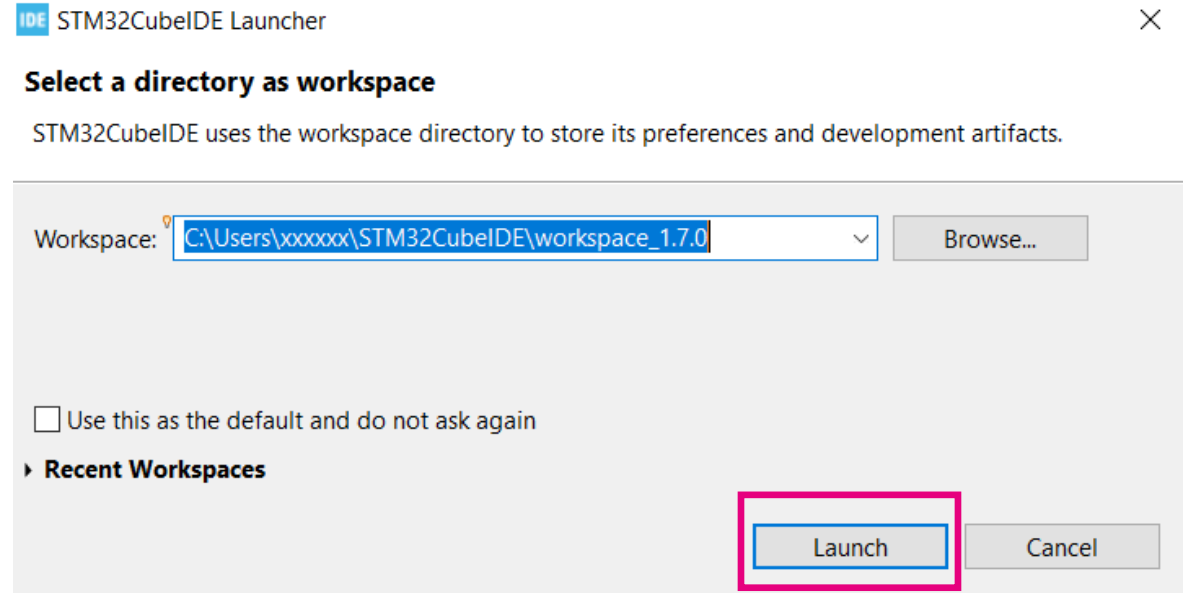
TraceX Installation

- Please go to directory c:\tmp\STM32-AZURE-RTOS-WS-Material\TraceX\
- Execute traceX installer
azure_rtos_tracex_setup_version_6.1.6.1_May_20_2021.exe
- Follow the instructions

Step 3 : Software packs installation

Use STM32CubeIDE to install software packs

- Launch STM32CubeIDE
- Select a workspace
- Launch



STM32CubeIDE start page

workspace_1.7.0 - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Information Center


STM32CubeIDE Home

Start a project

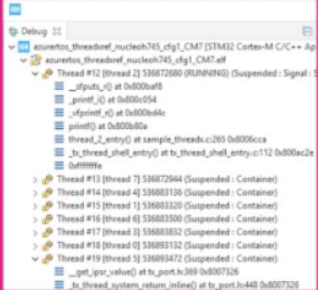
- Start new STM32 project
- Start new project from STM32CubeMX file (.ioc)
- Import SW4STM32 or TrueSTUDIO project
- Import STM32Cube example

Welcome to STM32CubeIDE

What's new



Debug experience with Azure RTOS ThreadX or FreeRTOS real time operating systems



How to access it : Project built with ThreadX > Debug enabled RTOS Proxy > "Debug view"

Quick links

- [Read STM32CubeIDE Documentation](#)
- [Getting Started with STM32CubeIDE](#)
- [STM32 MPU wiki](#)
- [STM32 MCU wiki](#)
- [Explore What's New in STM32CubeIDE](#)

Support & Community

- Twitter
- Facebook
- Youtube
- ST Home
- ST Community
- ST Longevity Commitment

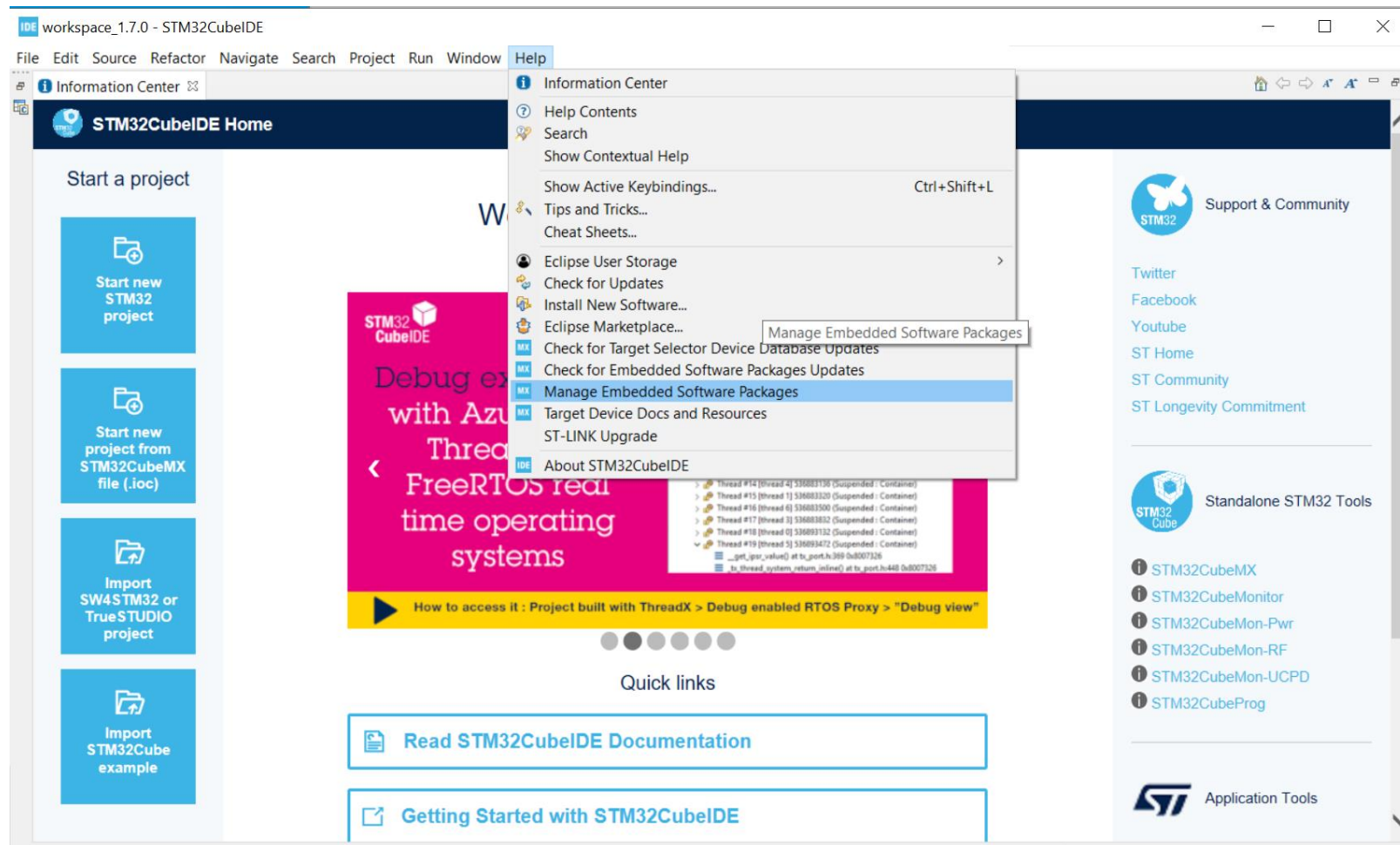
Standalone STM32 Tools

- STM32CubeMX
- STM32CubeMonitor
- STM32CubeMon-Pwr
- STM32CubeMon-RF
- STM32CubeMon-UCPD
- STM32CubeProg

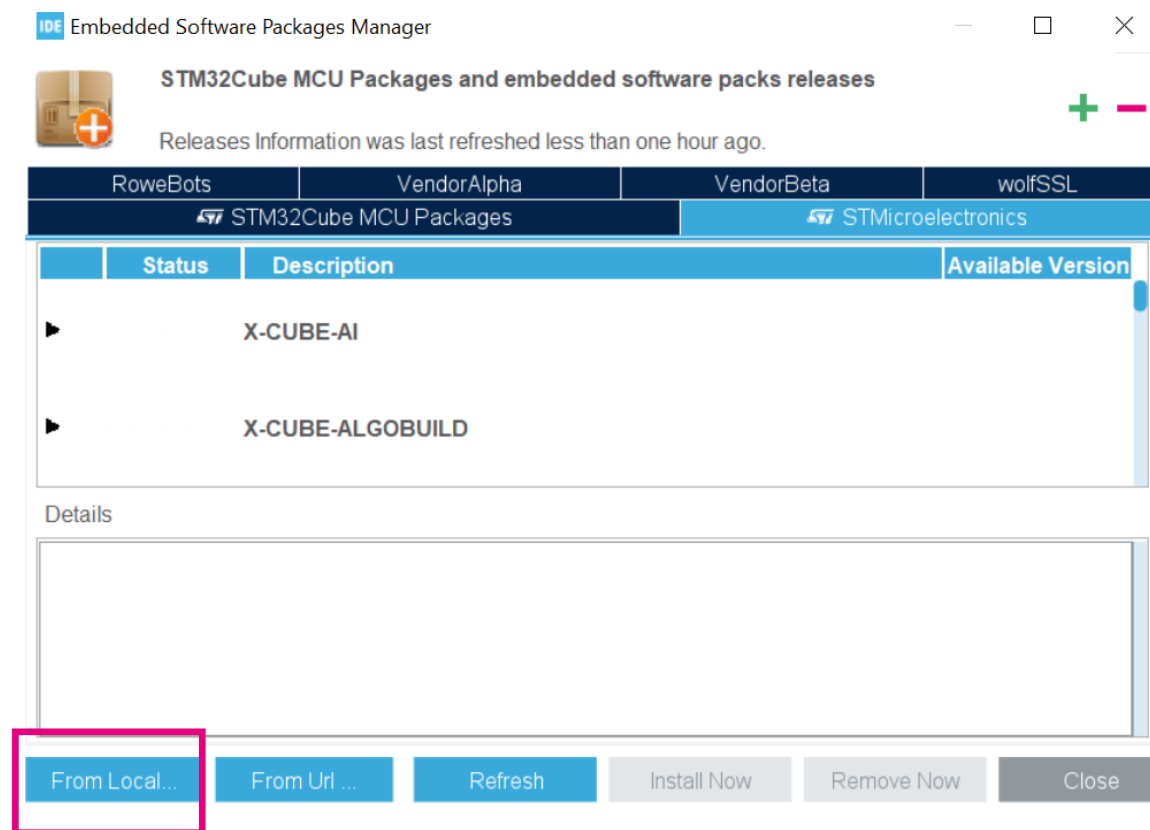
Application Tools

- eDesignSuite
- AlgoBuilder
- ST-MC-Suite

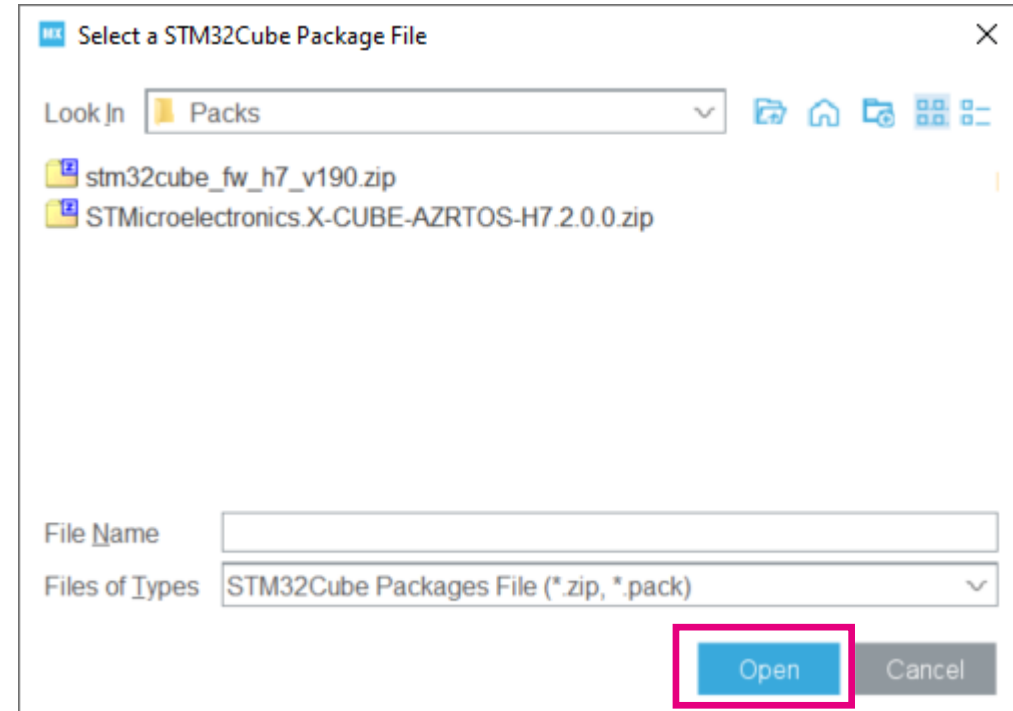
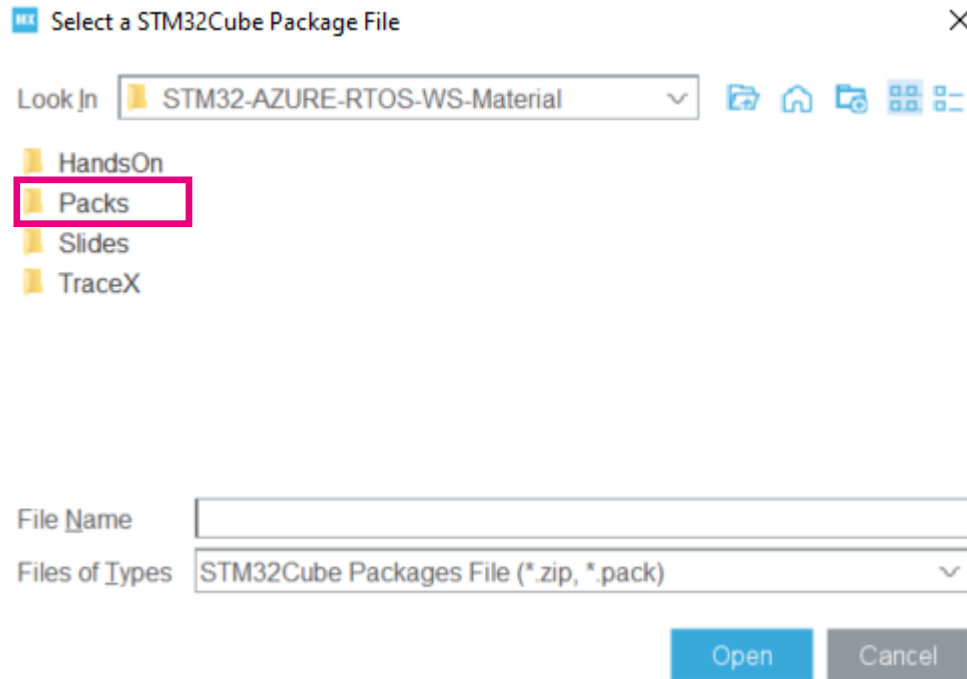
Select Manage Embedded Software Packages



Install packages from local



Install packages from local



- Please repeat the installation for the 2 packs provided if not already installed

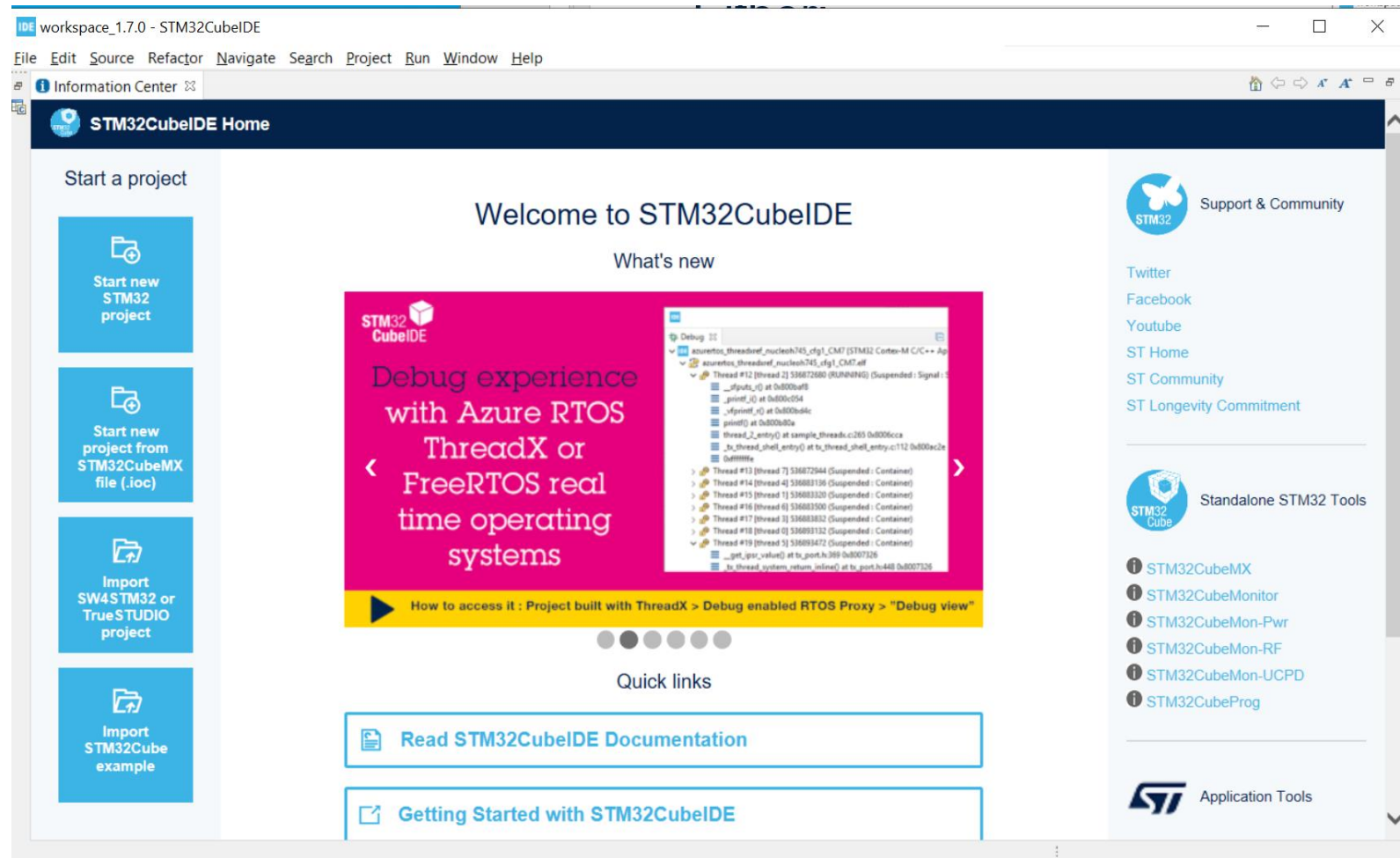
Step 4 : Check installation

Check your working environment

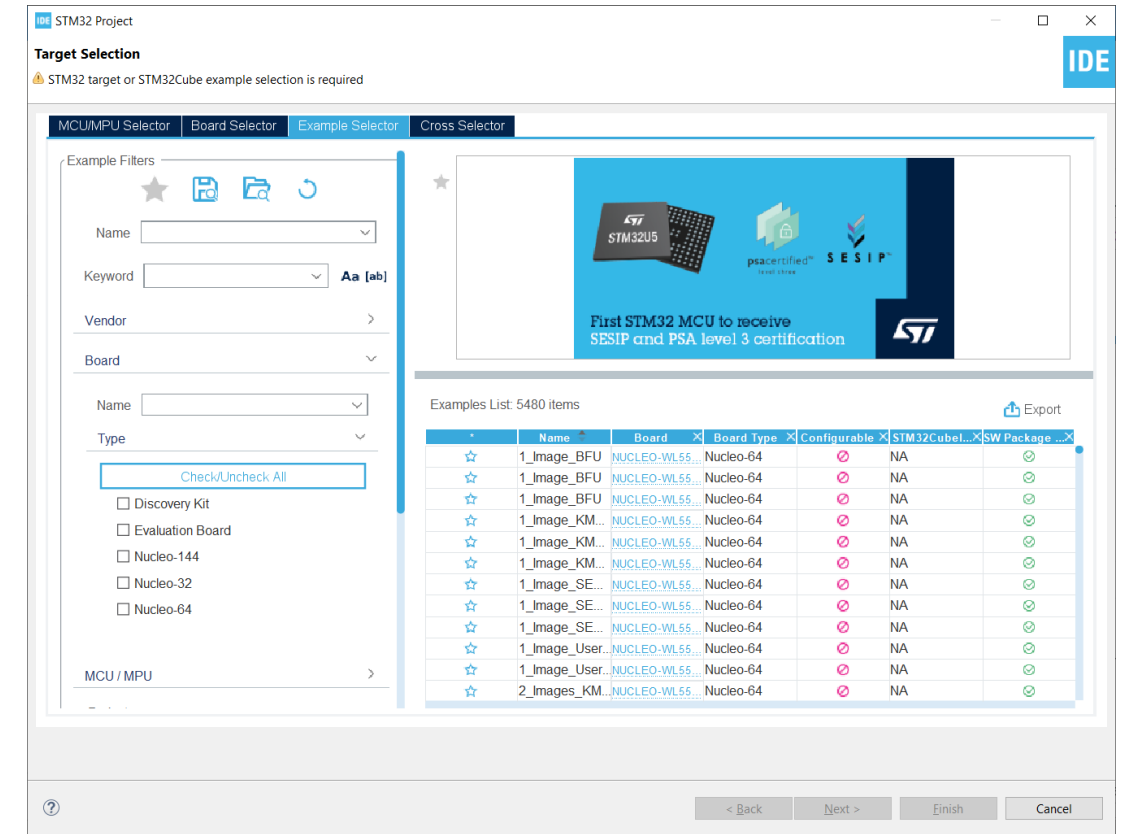
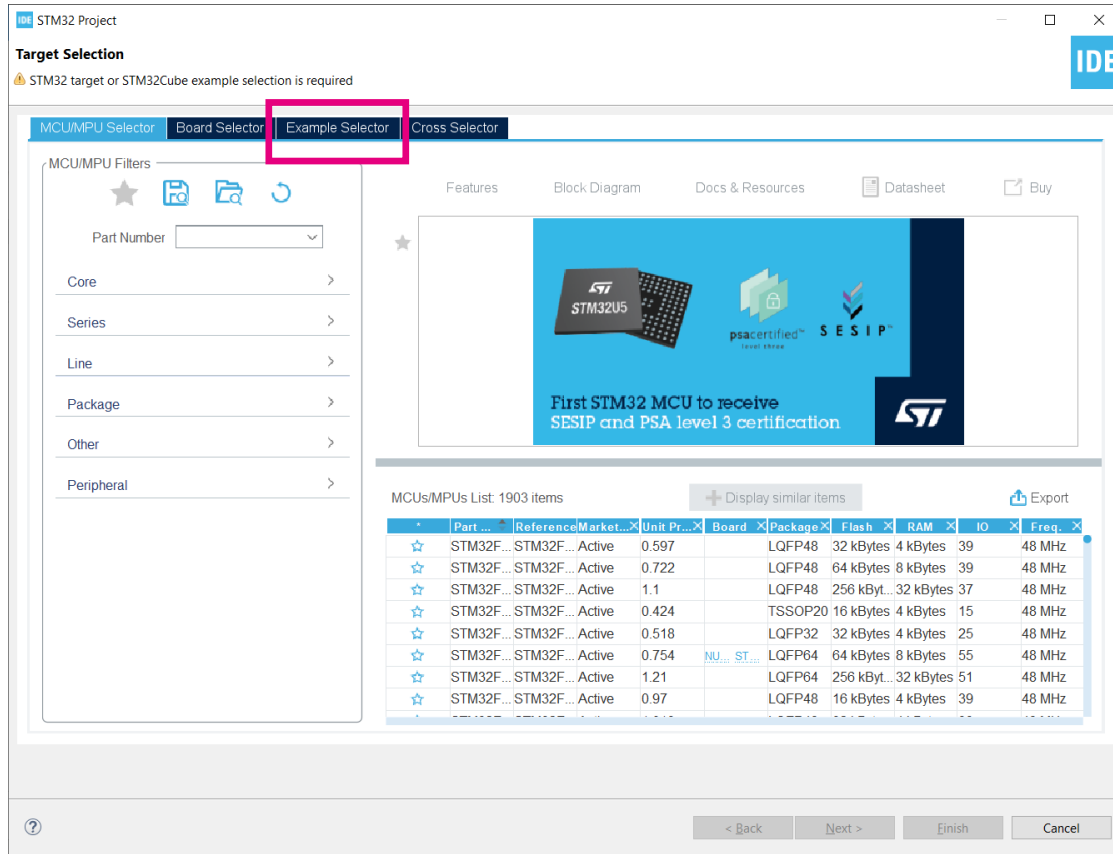
- The purpose is here to make sure everything is well installed and properly working
- We will go through the compilation and execution of an example based on AZURE RTOS
- Let's start from the opened STM32CubeIDE window ...
- You can also close the “Information Center” window

Create new STM32 project

- Either:
 - Click on Start new STM32 project
 - Or menu File/New/STM32 Project
- A target selection window should pop-up



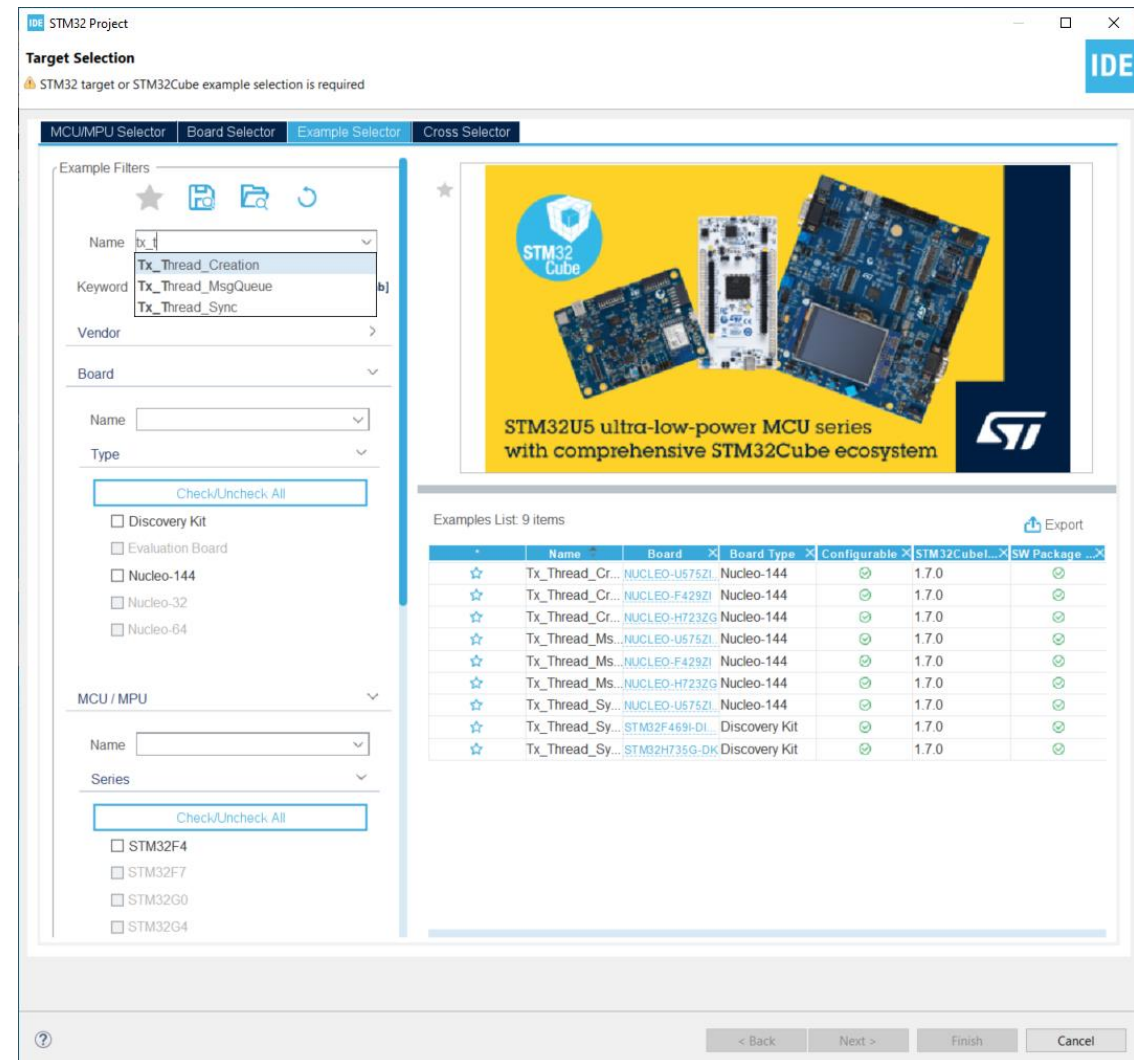
Access to Example selector



- Select TAB “Example Selector”

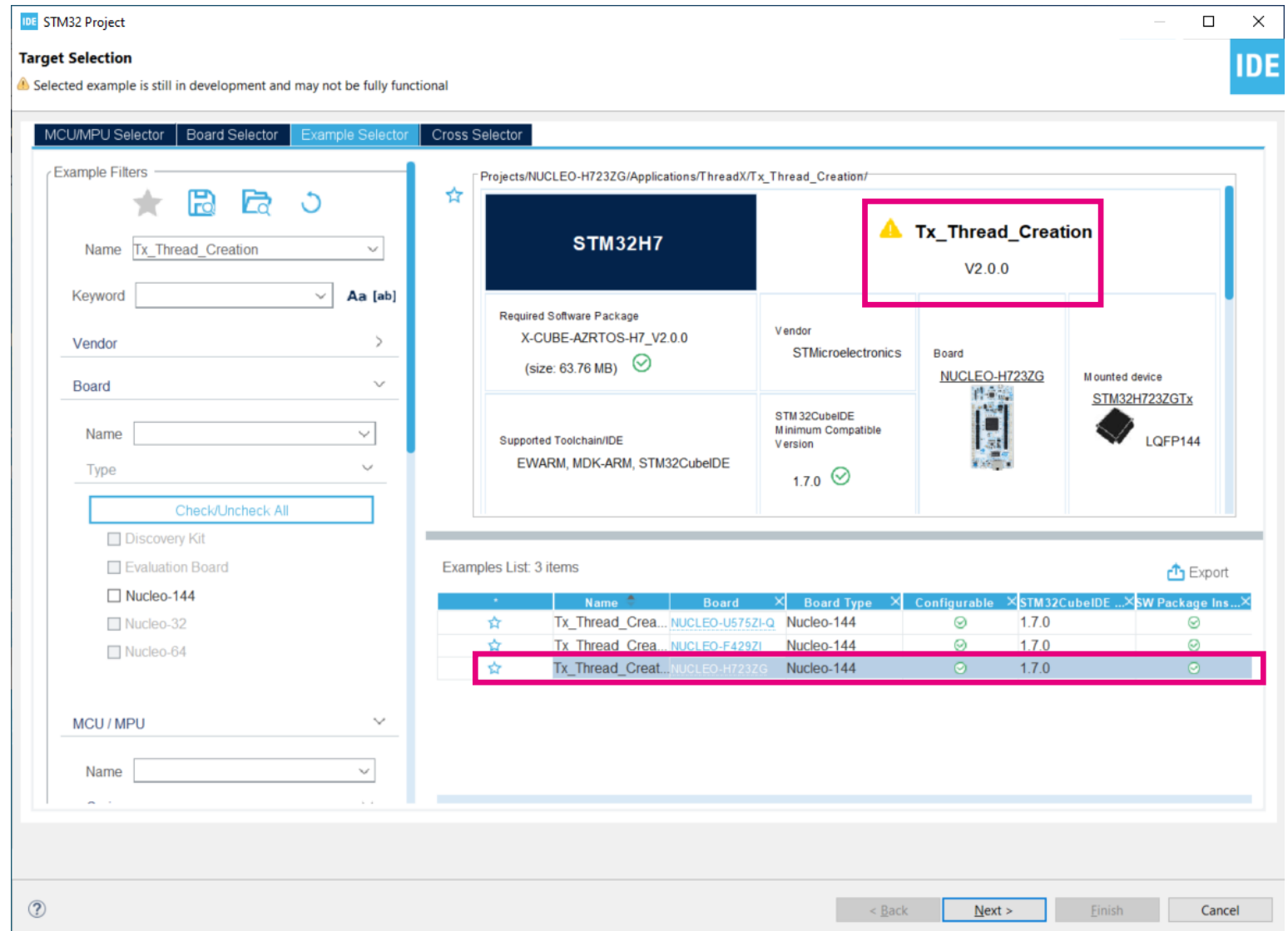
Find the example

- In Name field, start typing tx_th
- This will pop
Tx_Thread_Creation
- Click this
Tx_Thread_Creation
proposal



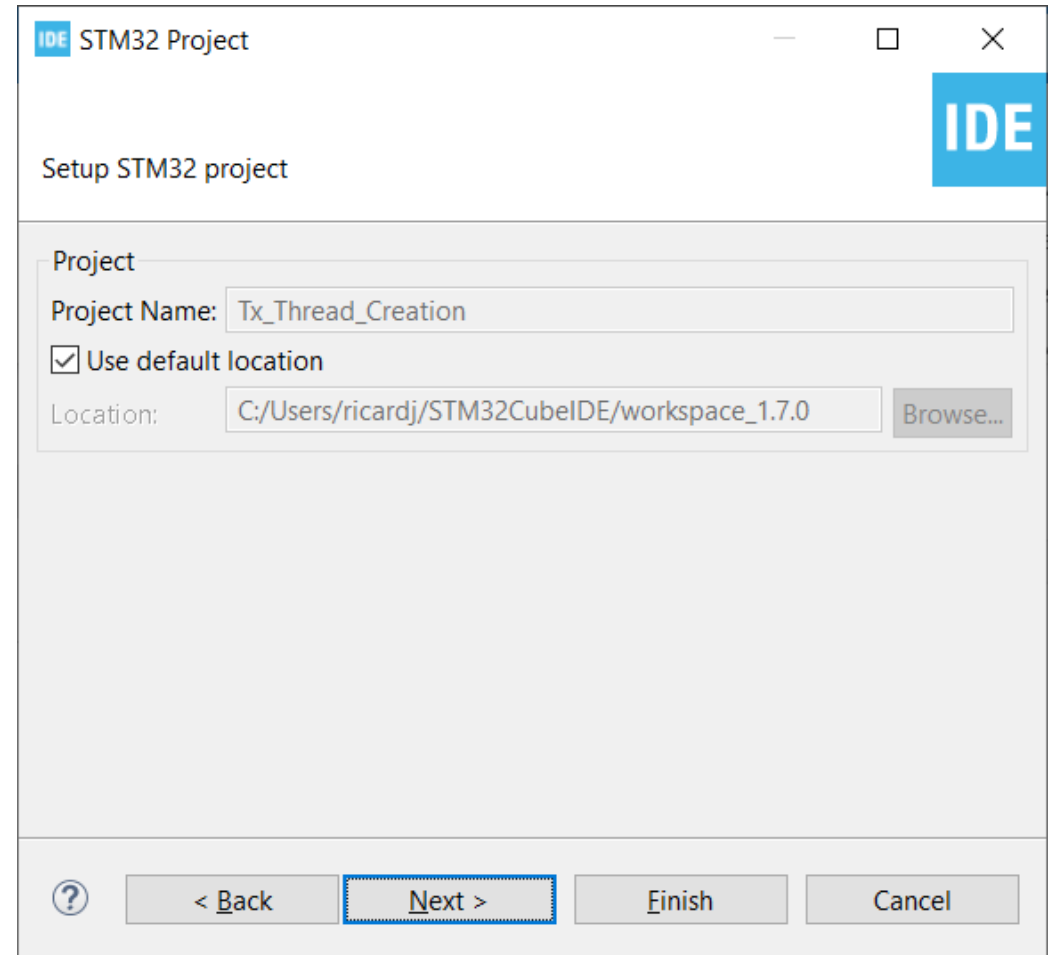
Select example

- Please select on the right part the Tx_Thread_Creation example made for STM32H723 on Nucleo 144.
- Then click Next



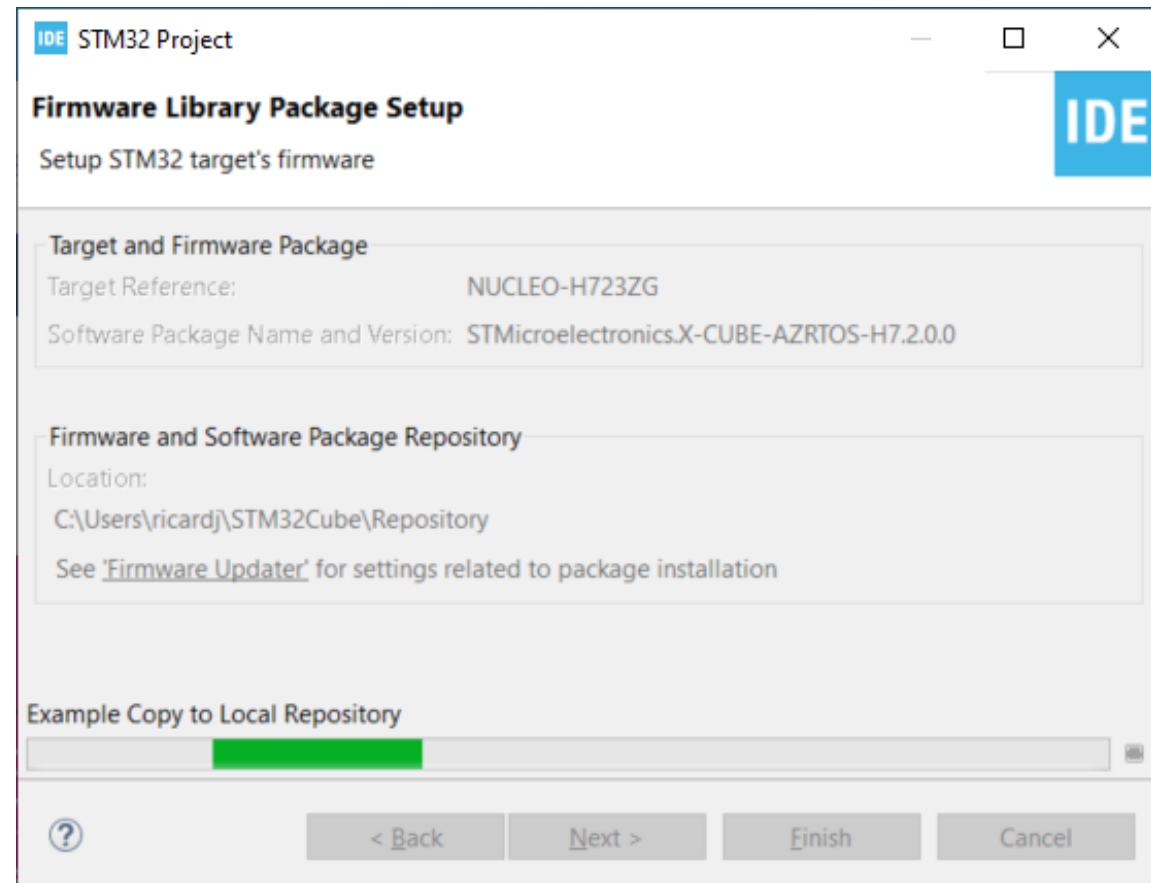
Select example import location

- Next pop-up proposes to set project location inside workspace.
- You can use this default location
- Click Next and then Finish.
- The example will be imported in the workspace



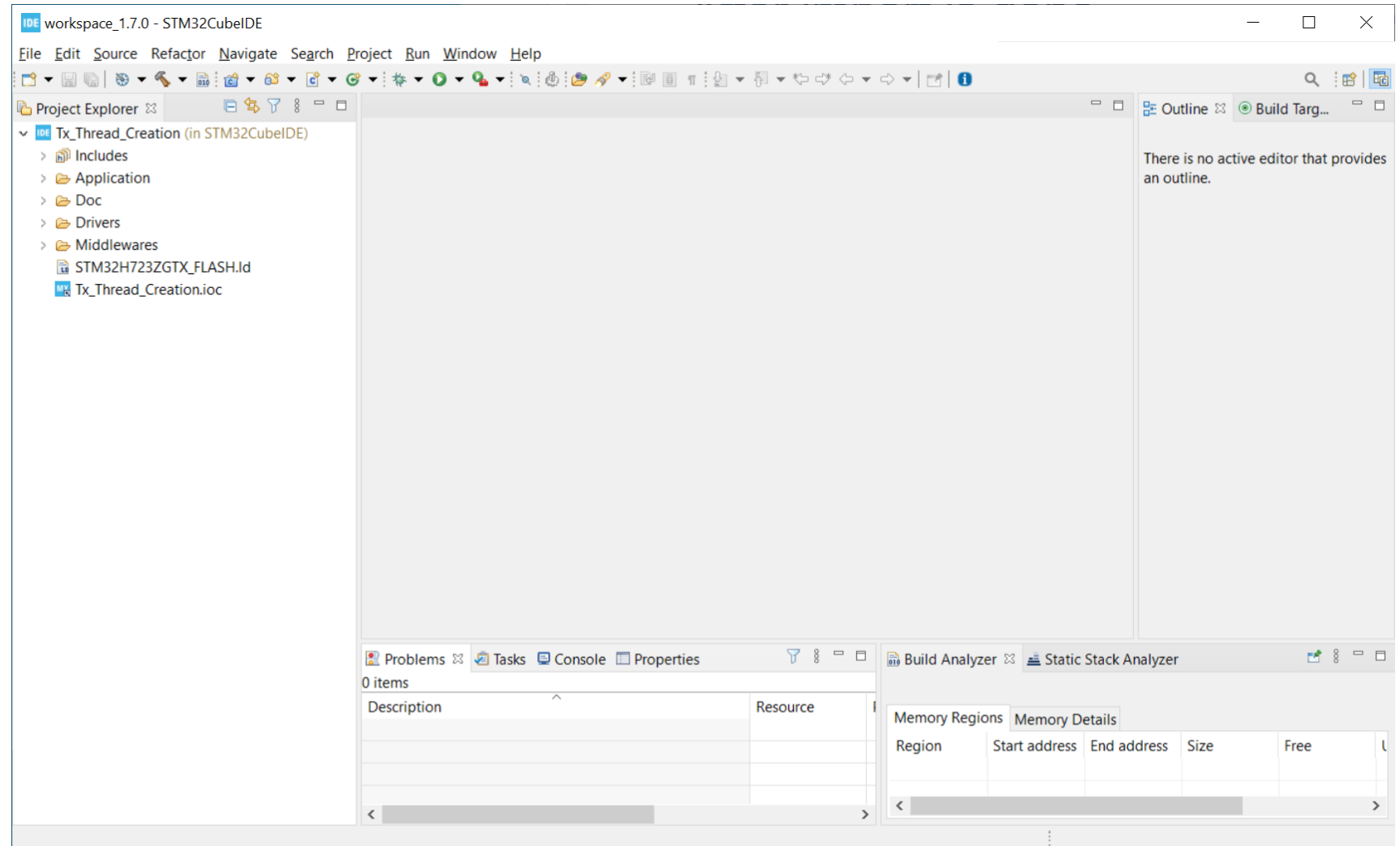
Example importing

- Just wait until the full example is imported.
- No need for any further click



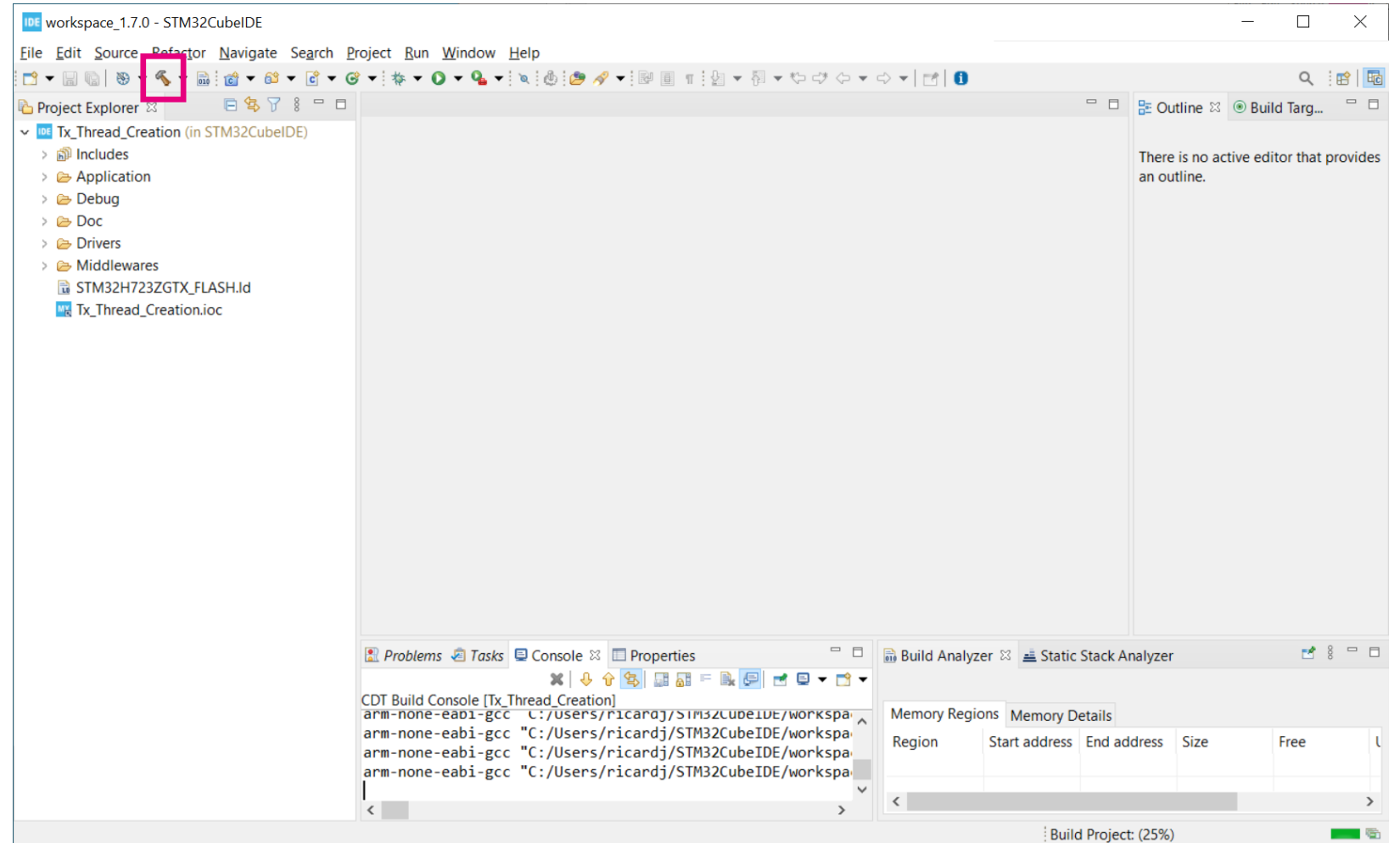
Check state after example import

- After import is done, the project is ready to be compiled
- You can open the items in the project explorer to see the sources files



Launch build

- Click on the hammer to launch the project build
- Compilation lines will appear in the Console window in the bottom



Check application build result

- State after build complete build

The screenshot displays the STM32CubeIDE interface after a successful build. The Project Explorer on the left shows the project structure for 'Tx_Thread_Creation'. The CDT Build Console in the center shows the build process, including the compilation of 'Tx_Thread_Creation.elf' and the generation of 'Tx_Thread_Creation.list' and 'Tx_Thread_Creation.bin'. The Build Analyzer on the right shows the memory regions and their usage.

CDT Build Console [Tx_Thread_Creation]

```
arm-none-eabi-gcc -o "Tx_Thread_Creation.elf" @objects.list -mcpu=cortex-m7 -T ^
Finished building target: Tx_Thread_Creation.elf

arm-none-eabi-size Tx_Thread_Creation.elf
arm-none-eabi-objdump -h -S Tx_Thread_Creation.elf > "Tx_Thread_Creation.list"
arm-none-eabi-objcopy -O binary Tx_Thread_Creation.elf "Tx_Thread_Creation.bin"
  text    data    bss    dec    hex filename
 21992    40     5912   27944   6d28 Tx_Thread_Creation.elf
Finished building: default.size.stdout
Finished building: Tx_Thread_Creation.bin

Finished building: Tx_Thread_Creation.list

11:51:21 Build Finished. 0 errors, 0 warnings. (took 16s.832ms)
```

Tx_Thread_Creation.elf - /Tx_Thread_Creation/Debug - Aug 30, 2021, 11:51:21 AM

Region	Start address	End address	Size	Free	Used	Usage (%)
ITCMRAM	0x00000000	0x00010000	64 KB	64 KB	0 B	0.00%
DTCMRAM	0x20000000	0x20020000	128 KB	128 KB	0 B	0.00%
FLASH	0x08000000	0x08100000	1024 KB	1002.48 KB	21.52 KB	2.10%
RAM_D1	0x24000000	0x24050000	320 KB	314.2 KB	5.8 KB	1.81%
RAM_D2	0x30000000	0x30008000	32 KB	32 KB	0 B	0.00%
RAM_D3	0x38000000	0x38004000	16 KB	16 KB	0 B	0.00%

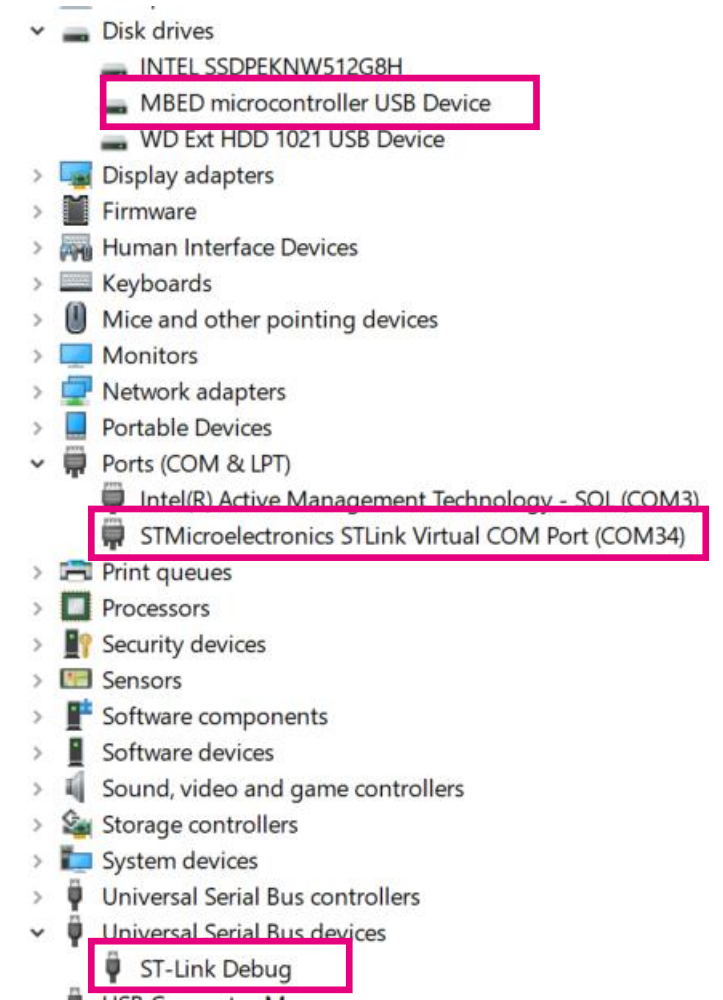
Download on target

- Connect your Nucleo STM32H723 to your PC with USB cable
- Warning: The board has 2 USB micro connectors and also a RJ45 for ethernet
- Use USB connector that is NOT besides the RJ45 connector



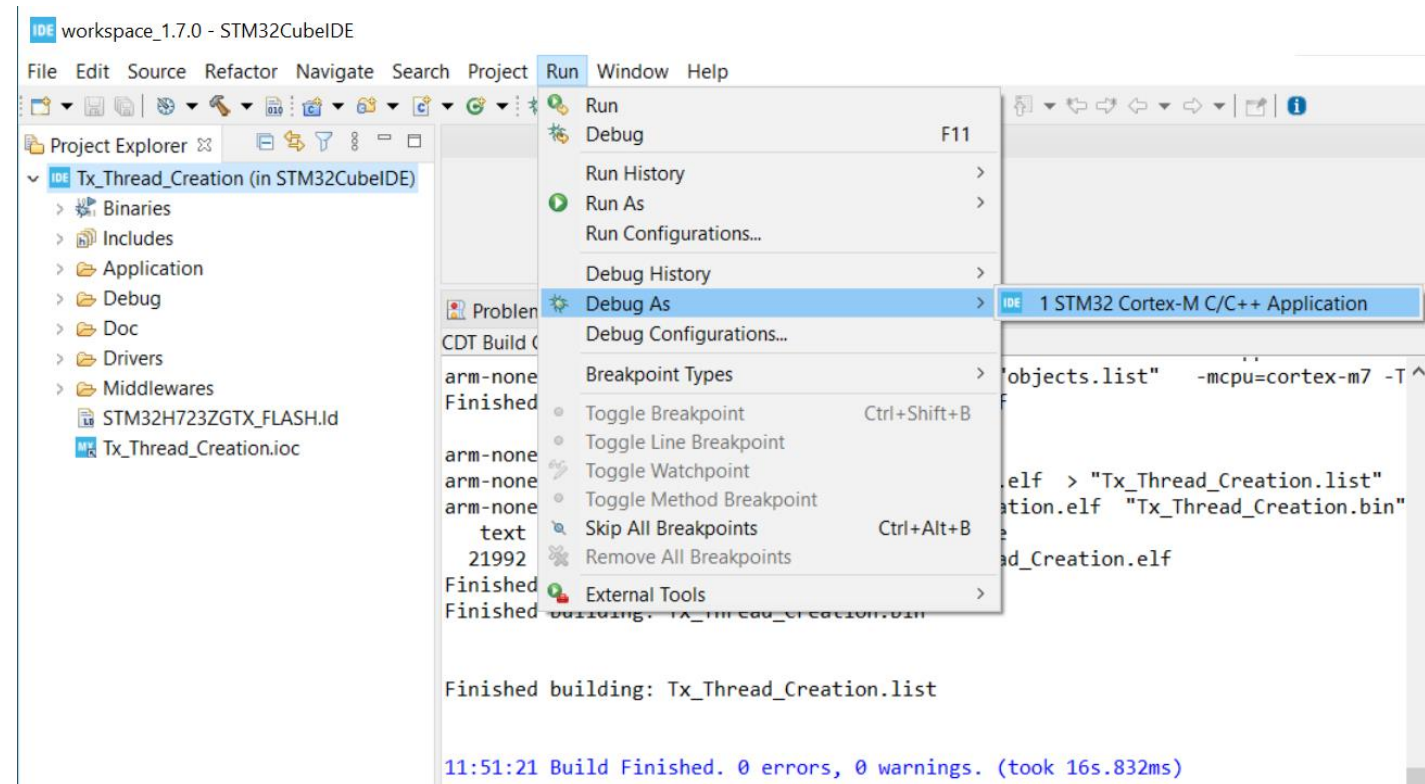
Check connection

- Once Nucleo is connected on your PC you should see:
 - In device manager:
 - MBED microcontroller USB device
 - STMicroelectronics STLink Virtual COM Port (COMxx)
 - ST-Link Debug



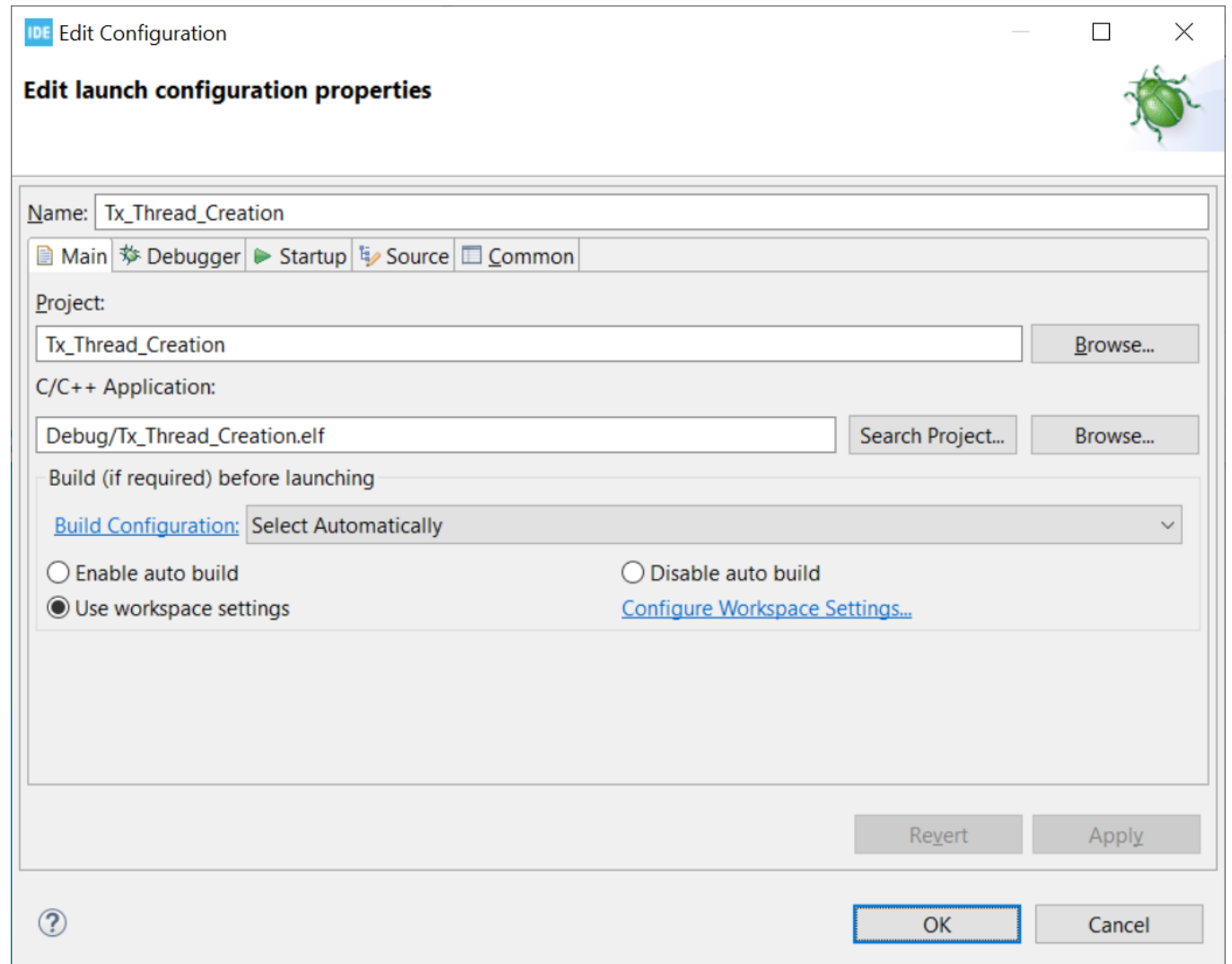
Launch debug

- Click on Tx_Thread_Creation project in Project explorer
- Then on the down arrow near the bug you will find : Debug As
- Only option is STM32 Cortex-M C/C++ Application
- Select it



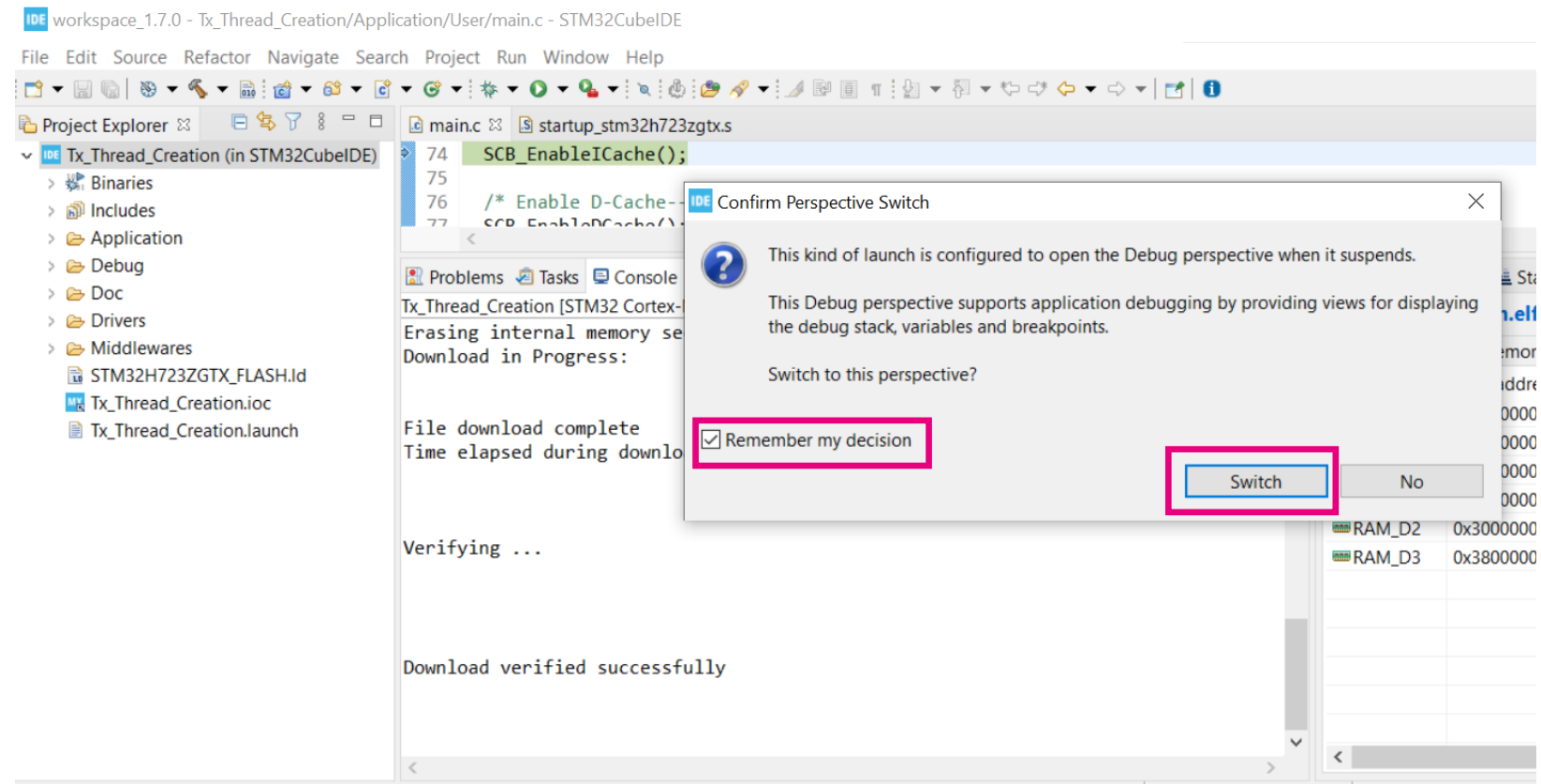
Debug configuration windows

- Window for debug configuration pop-up
- No need to change anything
- Press OK



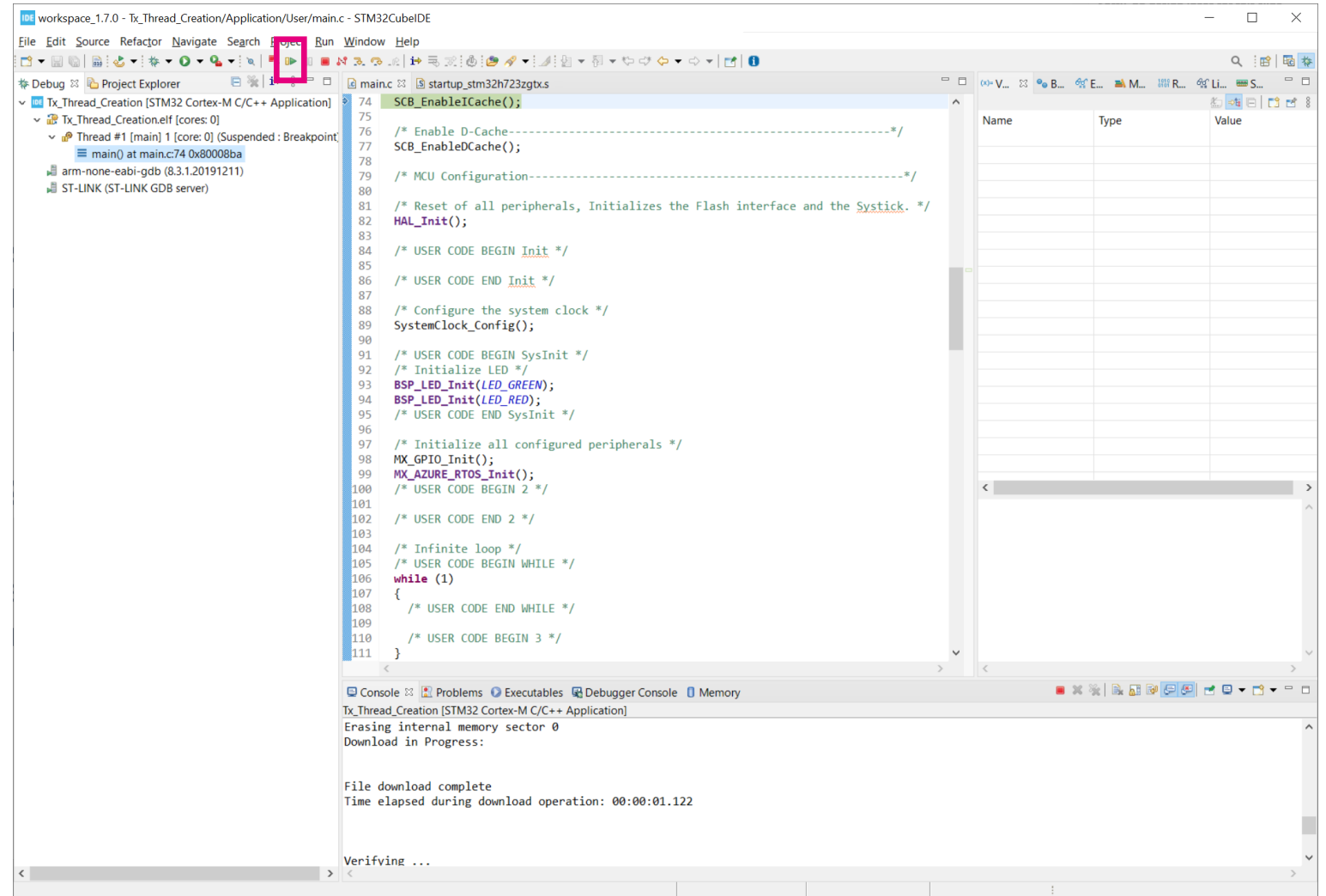
Select debugging perspective

- Debugger will connect to target, download the binary
- First time you will have a pop-up asking to switch to the debugging perspective
- Click remember decision
- Click switch



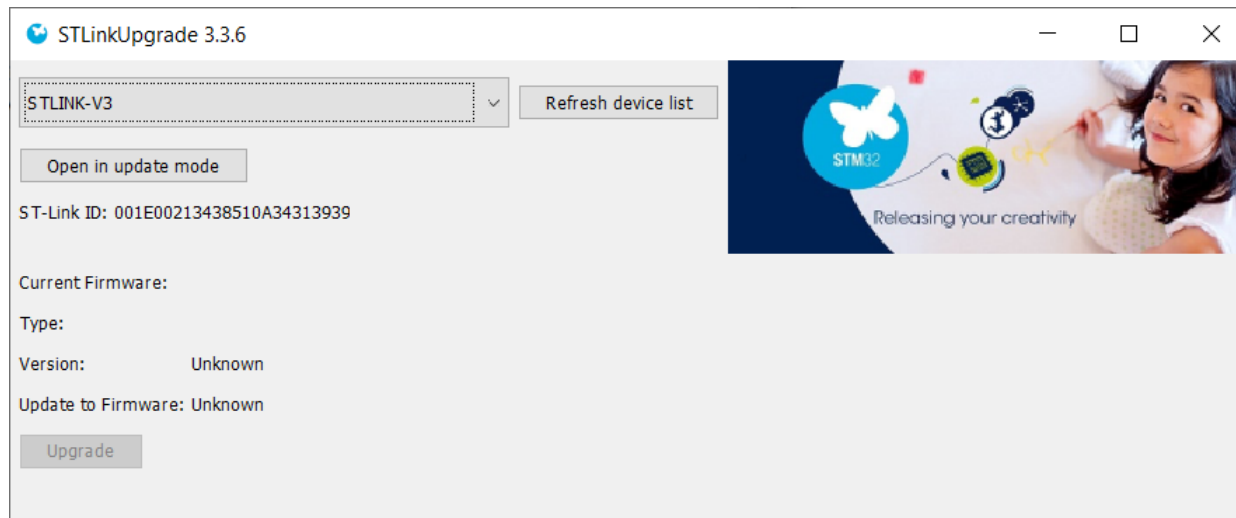
Launch application

- Now the debugger is launched and pointing to the first instruction after main.
- You can launch the example by pressing go button or F8
- **Before debugger is launched, STM32CubeIDE may ask for STLink update (see next slide)**



STLink Upgrade

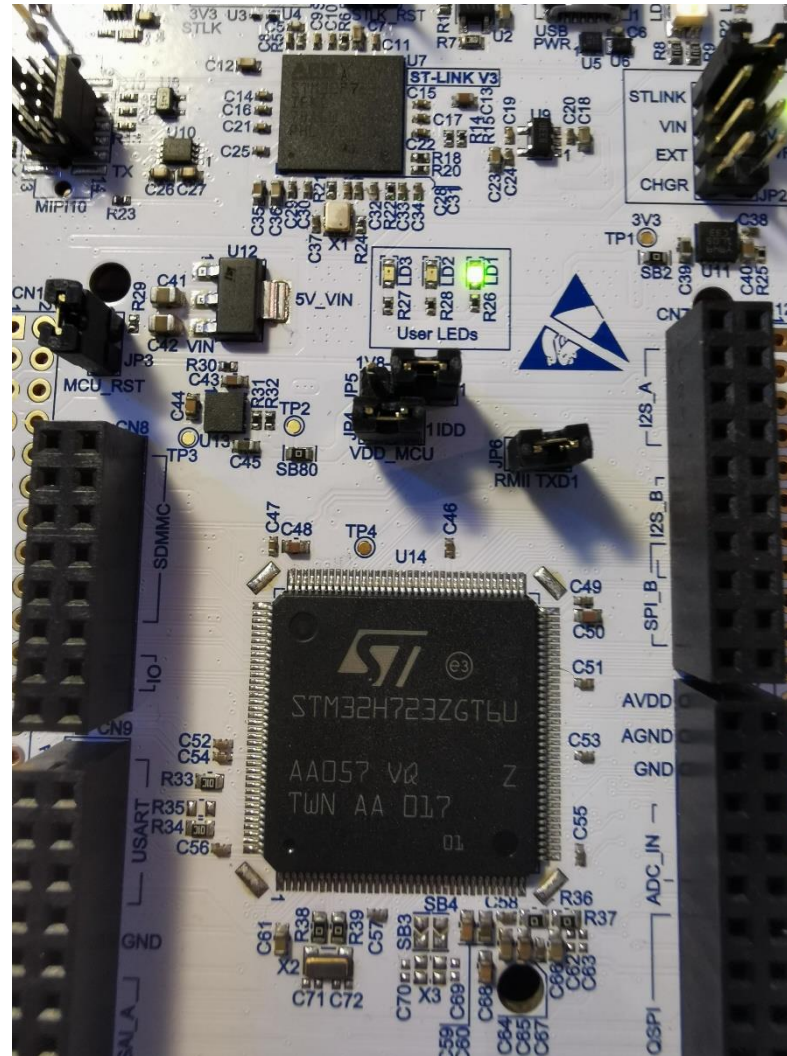
- It is possible that STM32CubeIDE asks for STLinkUpgrade. In that case answer yes and you should get a window like this



- Click open un update mode button and then Upgrade.
- Once finished you can launch again the debugger same way as before

Check LED Blinking on target

- Check on Nucleo target
- LD1 should blink !



Homework is complete

- You could go through all the steps: you are ready for the workshop
- Something went wrong
 - Please describe what happened on community forum
 - We will try to help you!

Thank you