서울시립대학교

Applied Microprocessor Lab VS Code Env Setting

전력전자연구실

Index

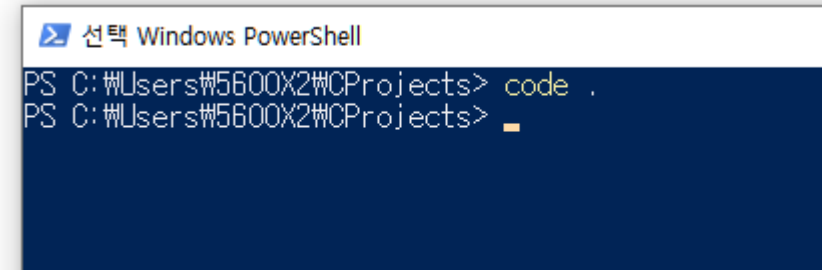
- ✓ CubeSTM32 into VS Code
- ✓ Make a Project with CubeMX
- ✓ Build and Download into MCU

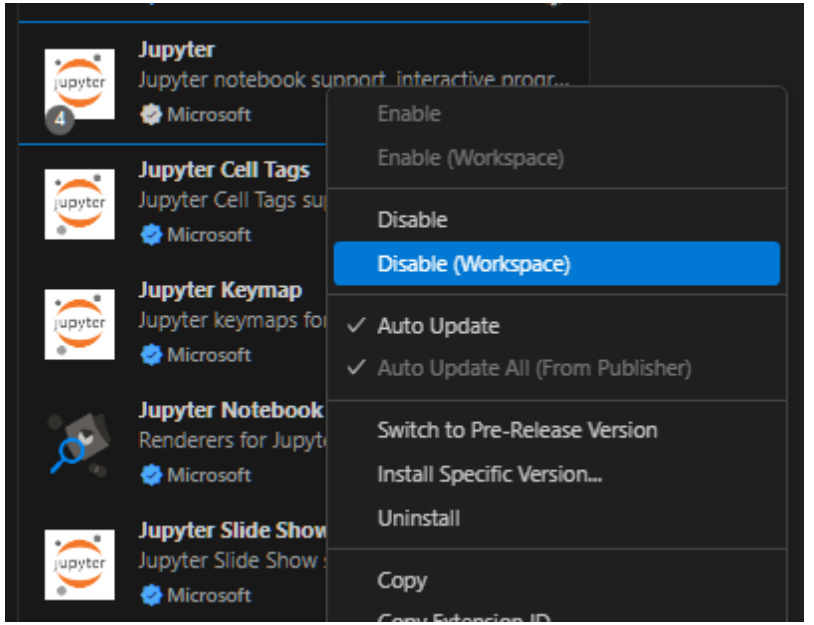
CubeSTM32 into VS Code

✓ Launch VS Code

- Do not use Onedrive / Gdrive / 한글 in Project Path to avoid permission issues.

- Turn off extensions you will not use





CubeSTM32 into VS Code

✓ Pros and Cons

✓ Pros

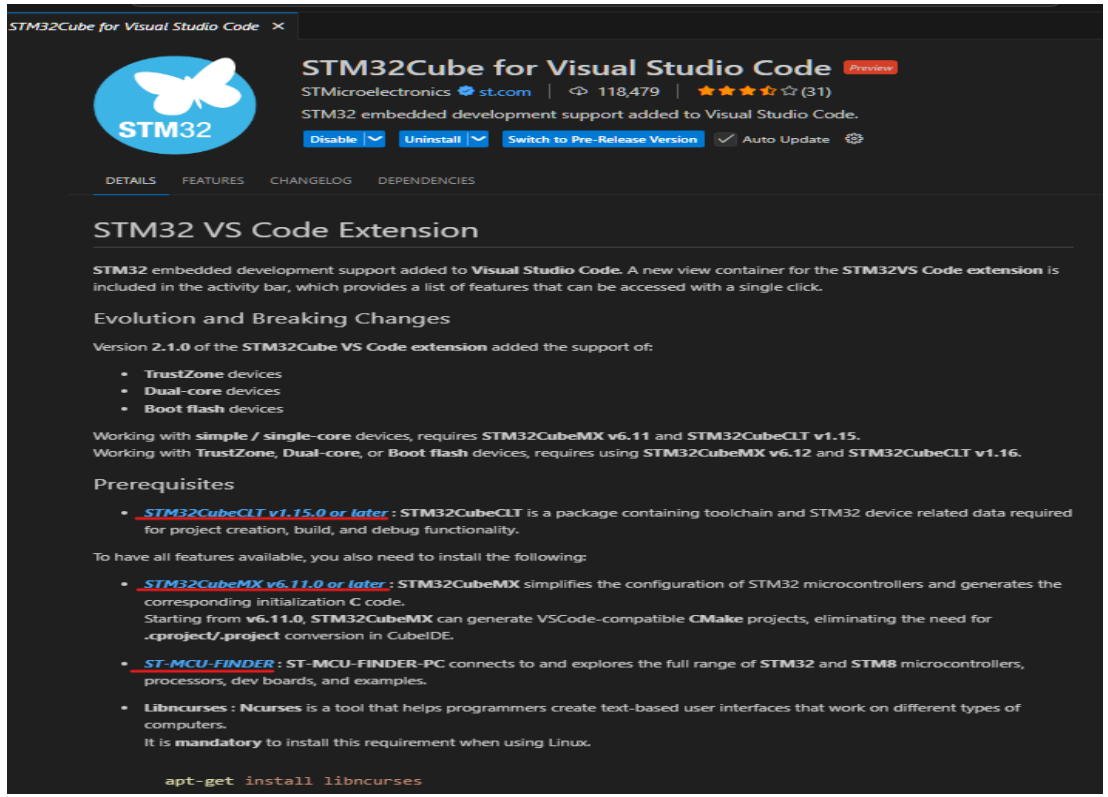
- Can use VS Code
- Works on all OS (Windows / Mac / Linux)
- Free to use / Opensource

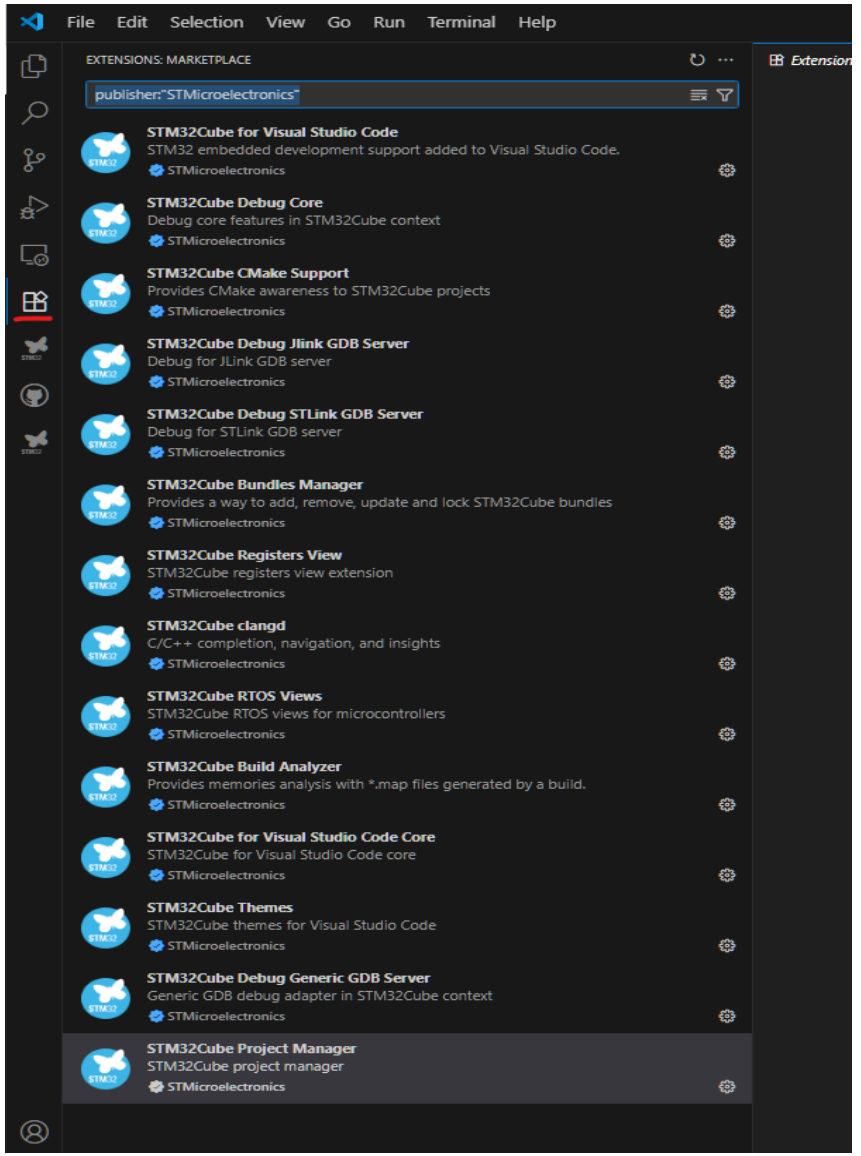
✓ Cons

- No official technical support or warranty
- Many companies prefer paid tools with professional support

CubeSTM32 into VS Code

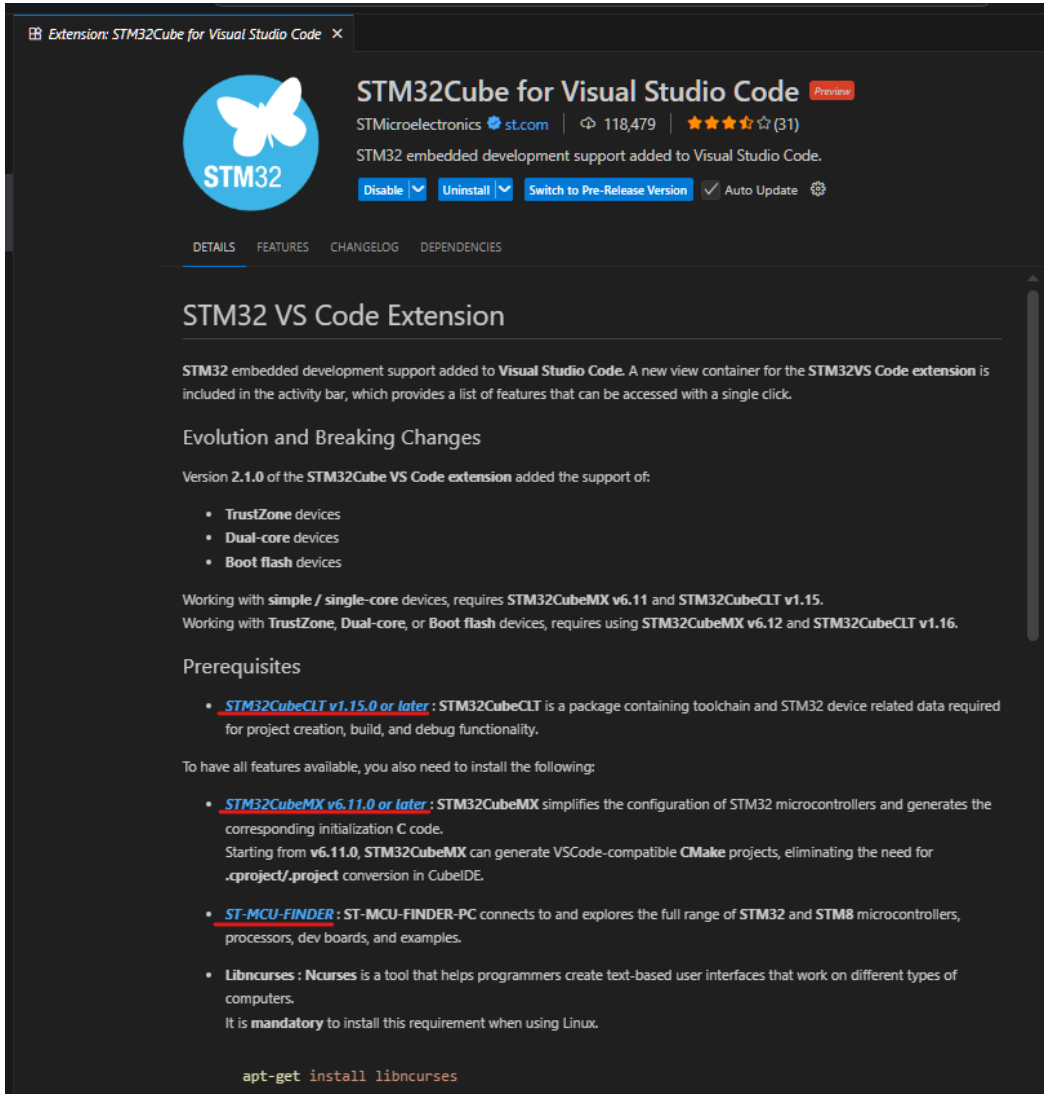
- ✓ Install extensions
  - Search ‘stm’ in extension menu in VS Code
  - Install Prerequisites





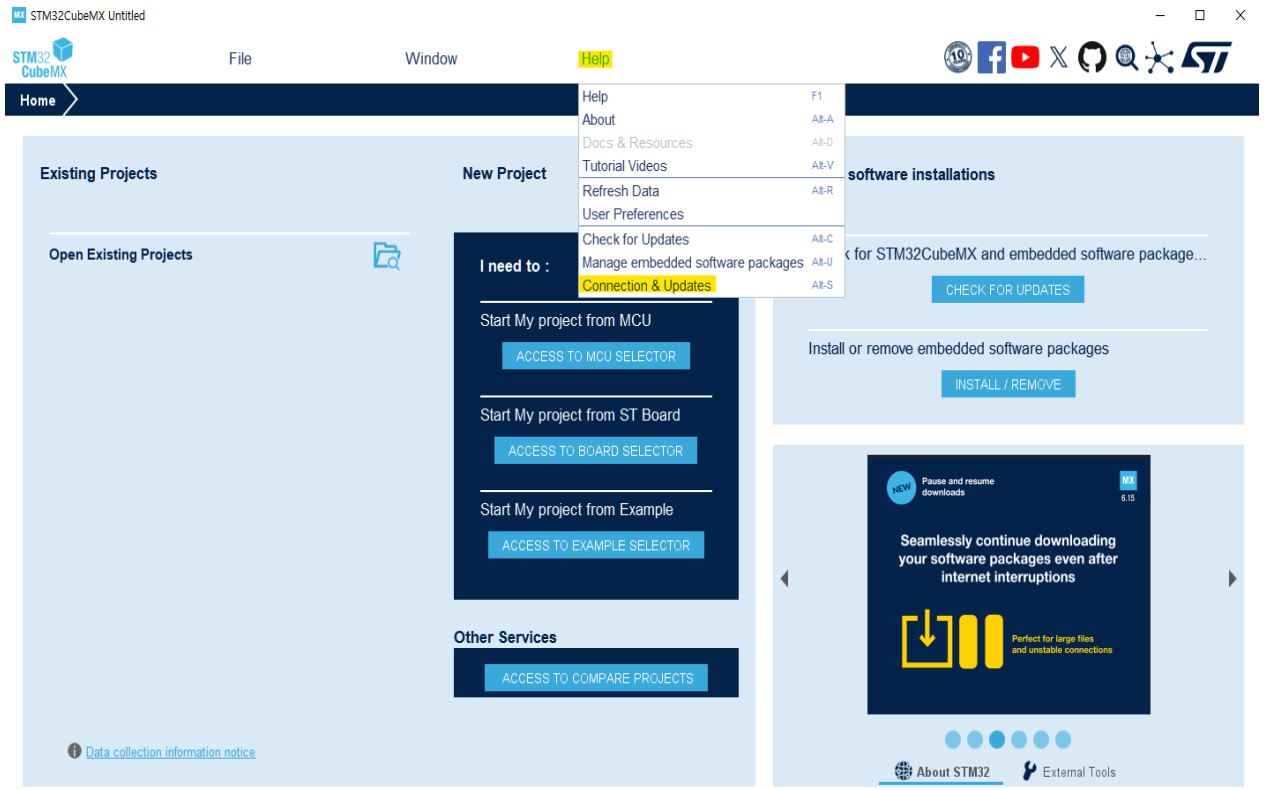
CubeSTM32 into VS Code

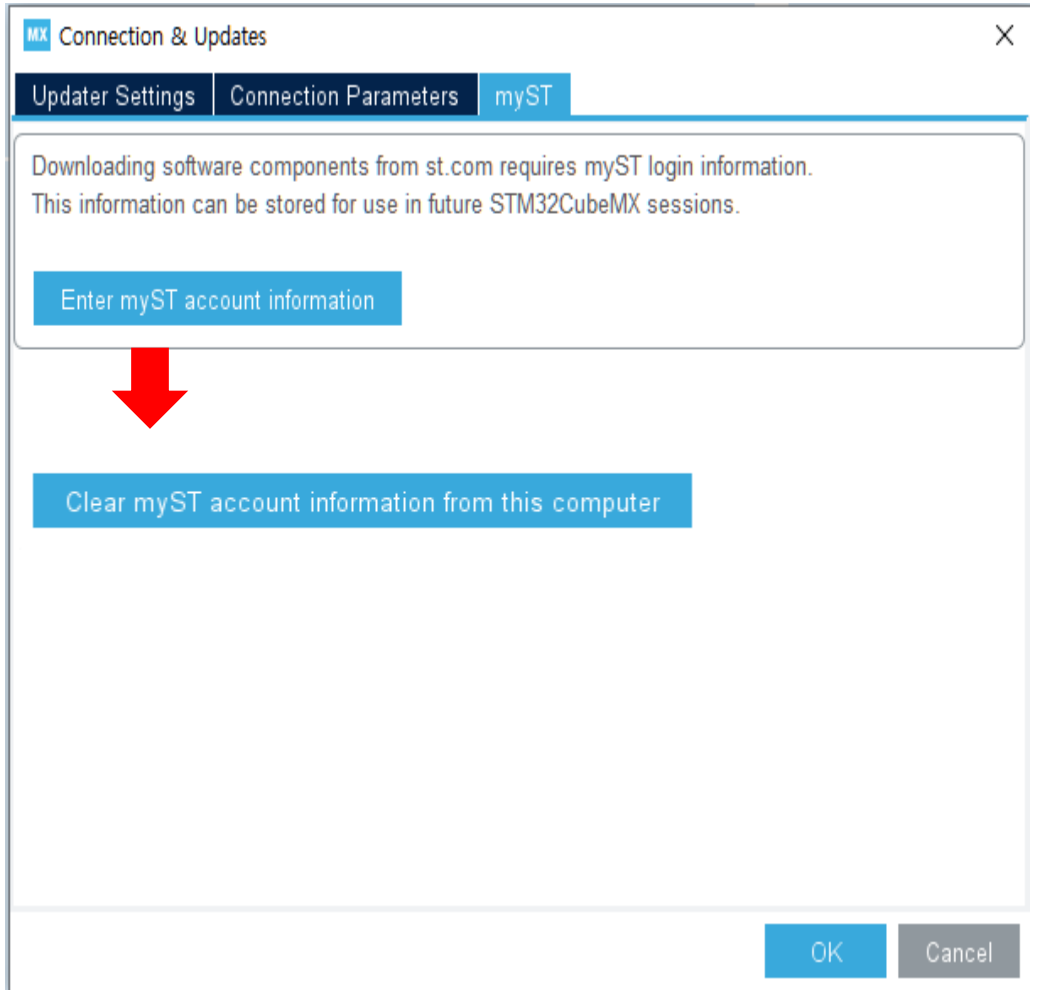
- ✓ Install extensions
  - STM32CubeCLT : ARM C Compler, Debugger, Library
  - STM32CubeMX : Help you make STM Projects
  - STMCUFinder : Find and Download into mcu board



Make a Project with CubeMX

- ✓ Make a new C Project with CubeMX
  - Login first myST

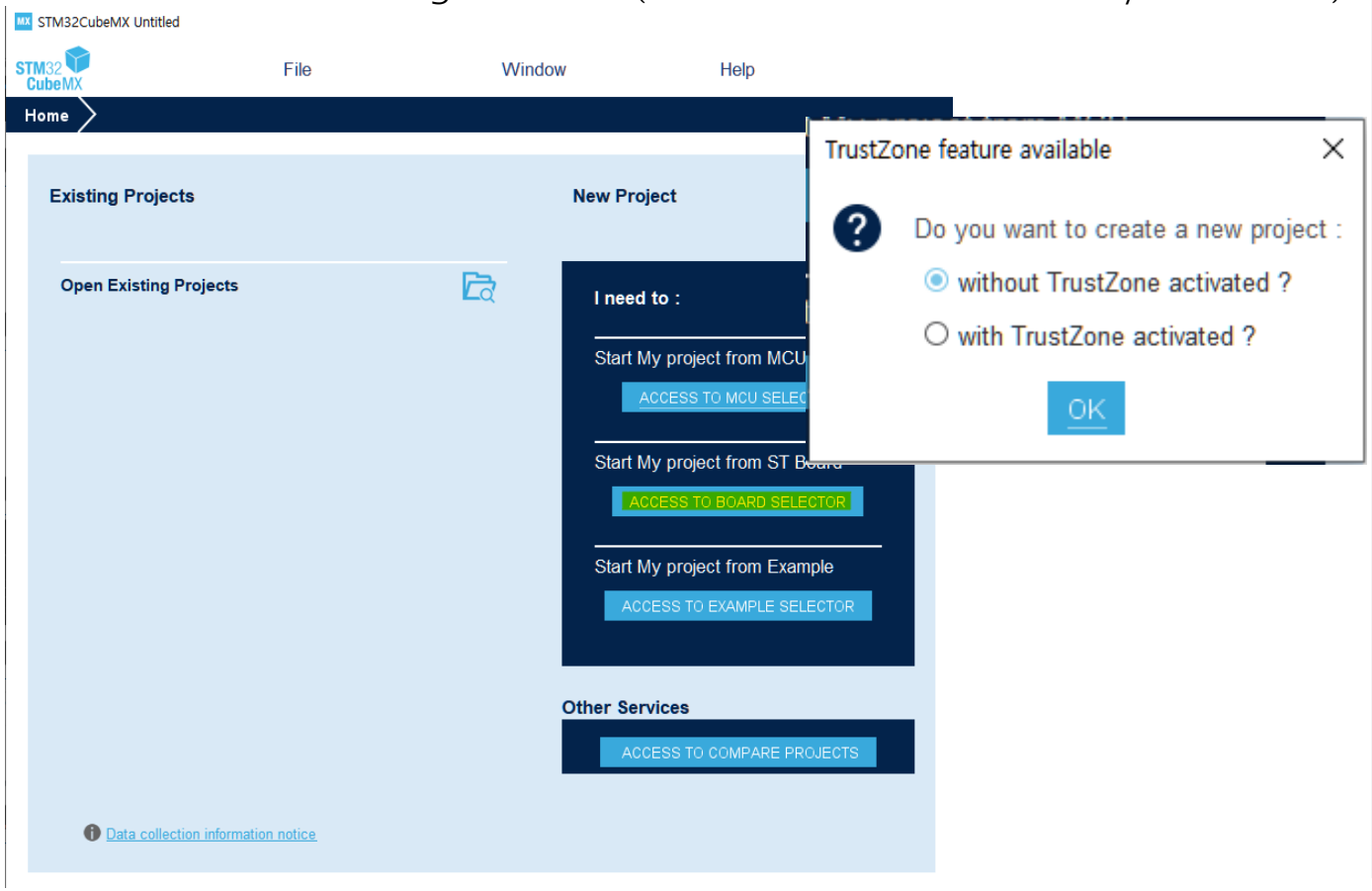


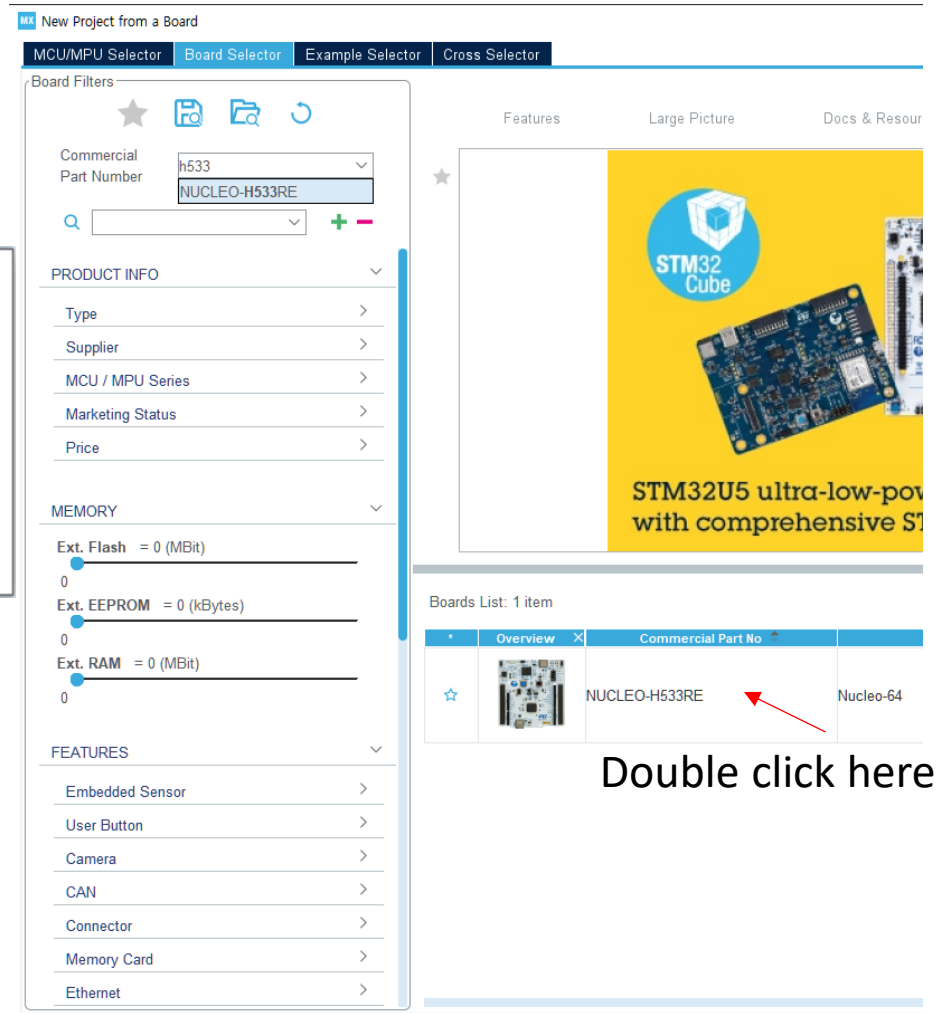


Make a Project with CubeMX

✓ Make a new C Project with CubeMX

- Select the right board (check model number of your board)

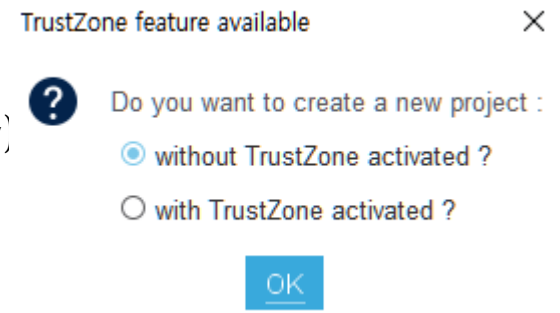




Make a Project with CubeMX

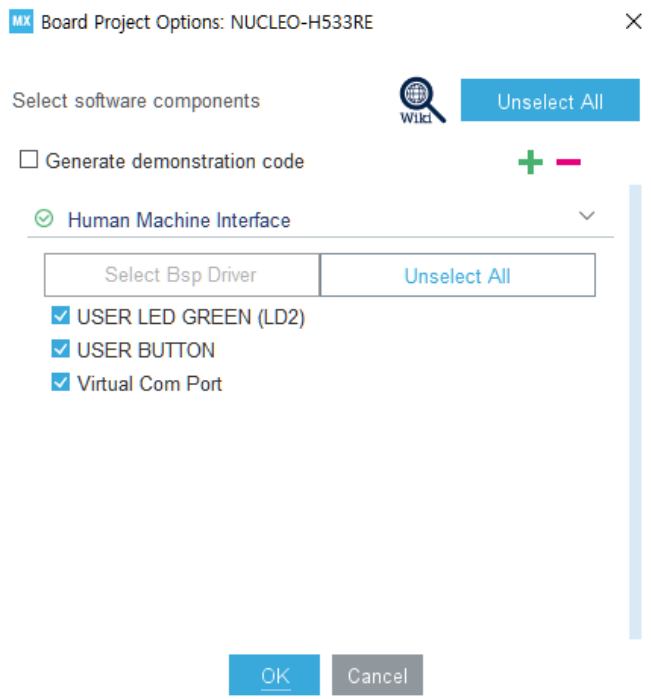
✓ Make a new C Project with CubeMX

- TrustZone is not covered in this class (need for product-level security)
  (It may not appear depending on the board)



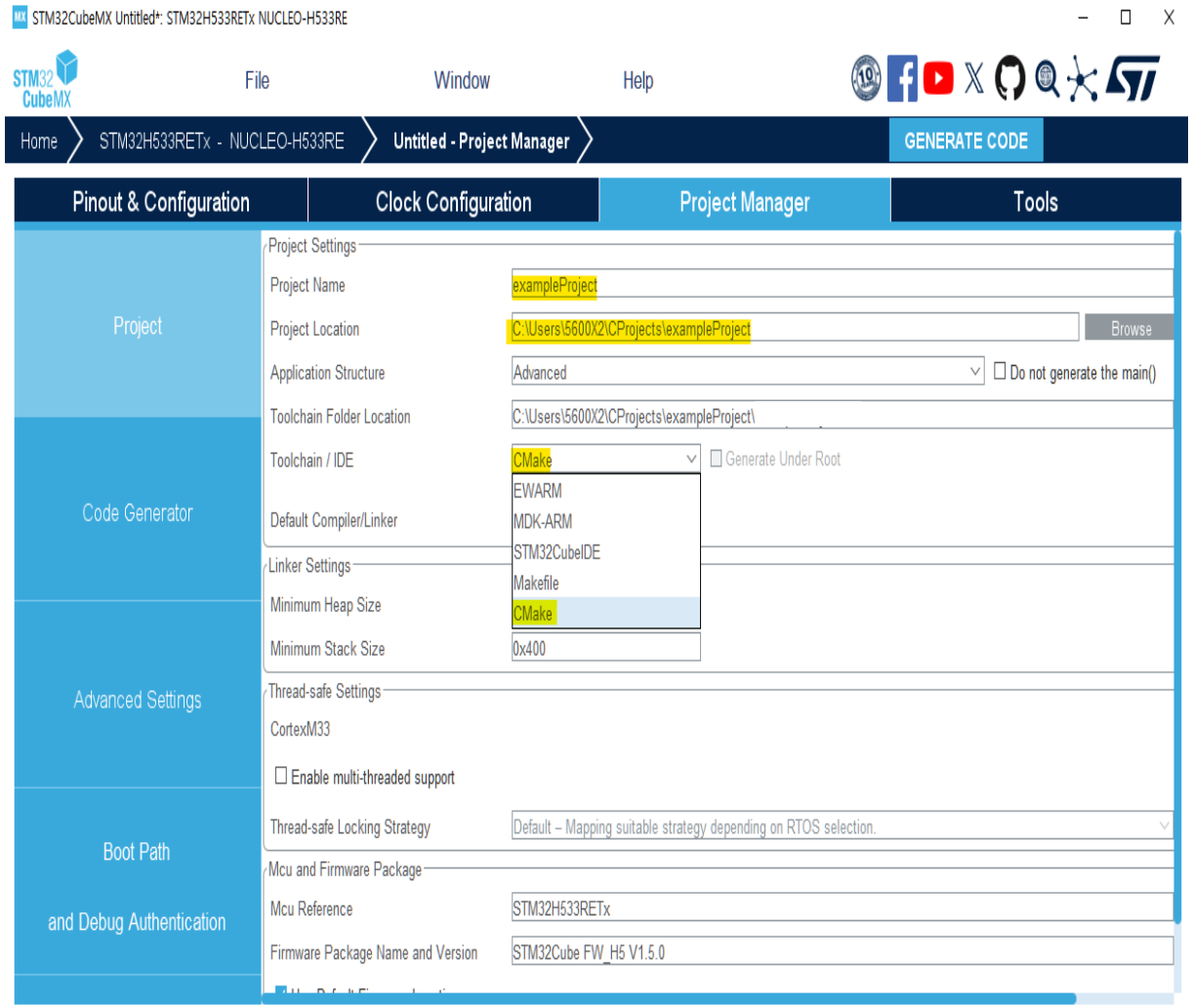
- Press OK if no error occurs

- LED and Button, useful for build testing



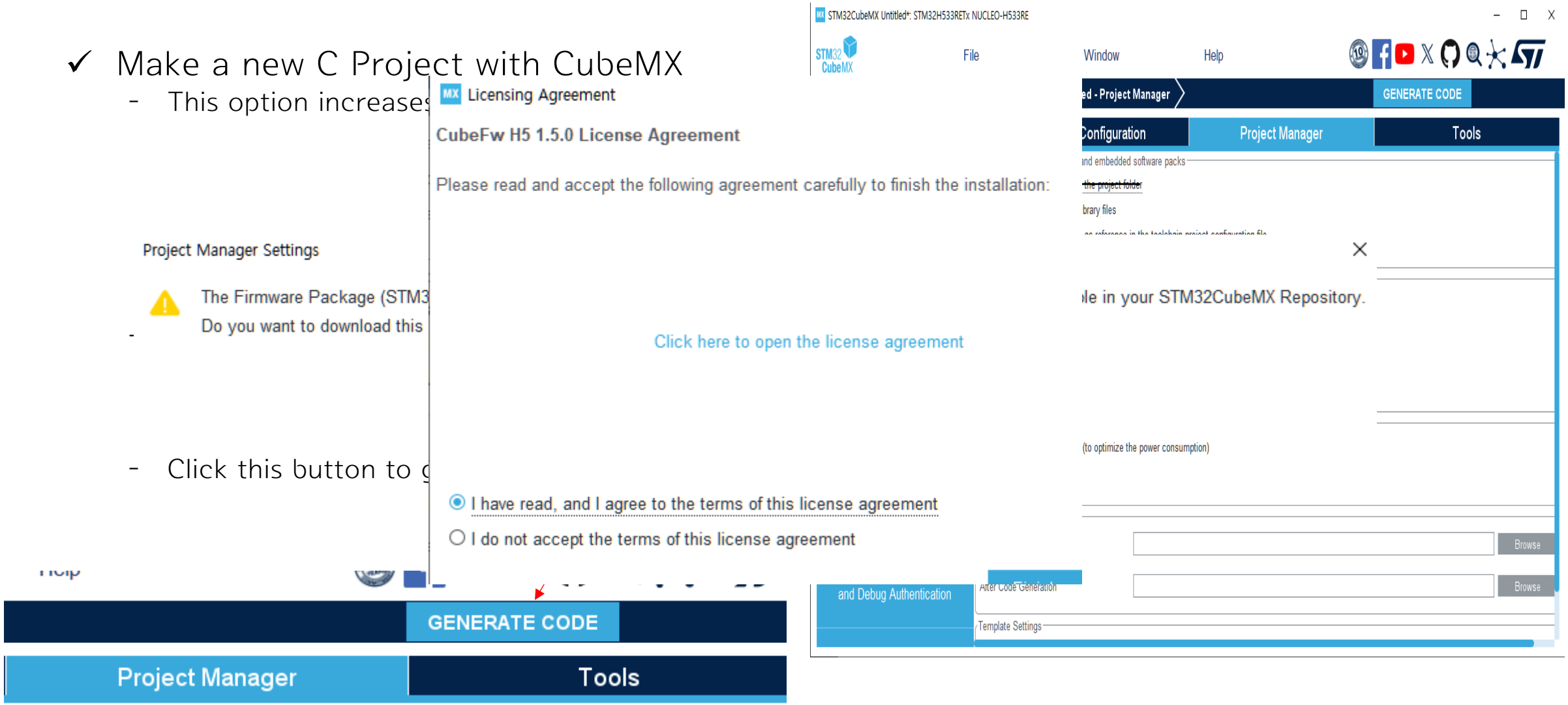
Make a Project with CubeMX

- ✓ Make a new C Project with CubeMX
  - Use the exact path and select Cmake toolchain
  - Pinout and clock configuration will be covered later



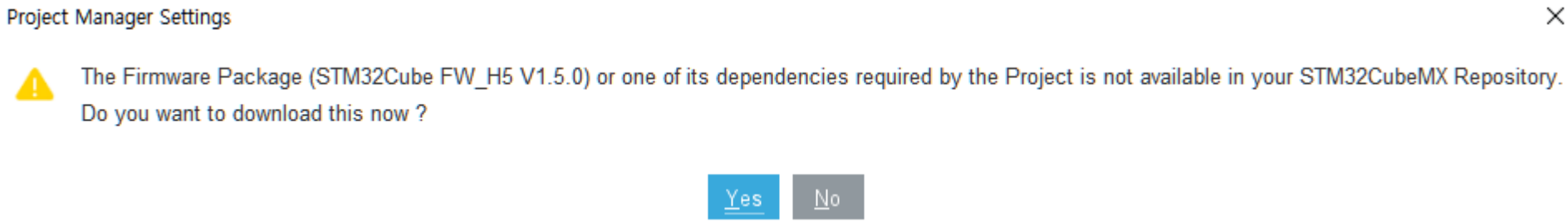
Make a Project with CubeMX

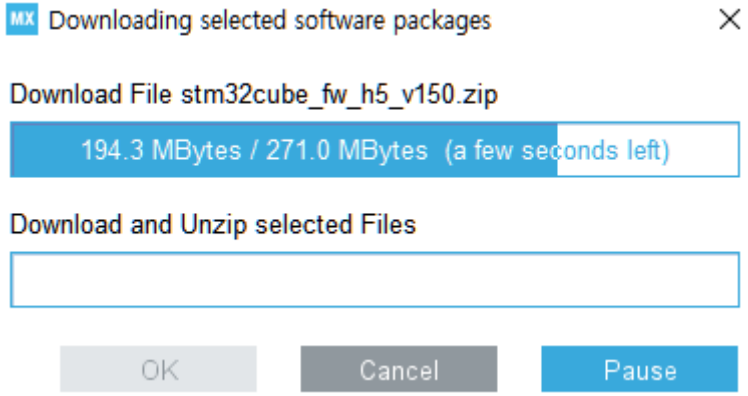
- ✓ Make a new C Project with CubeMX
  - This option increases
  - Click this button to g

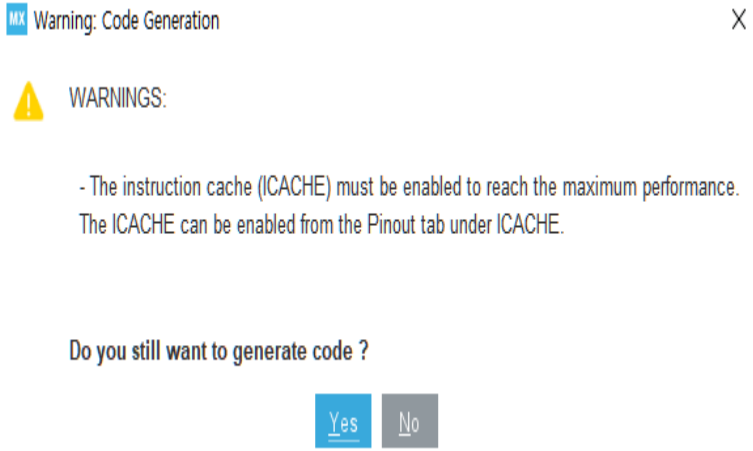


Make a Project with CubeMX

- ✓ Make a new C Project with CubeMX
  - Press OK if no error occurs

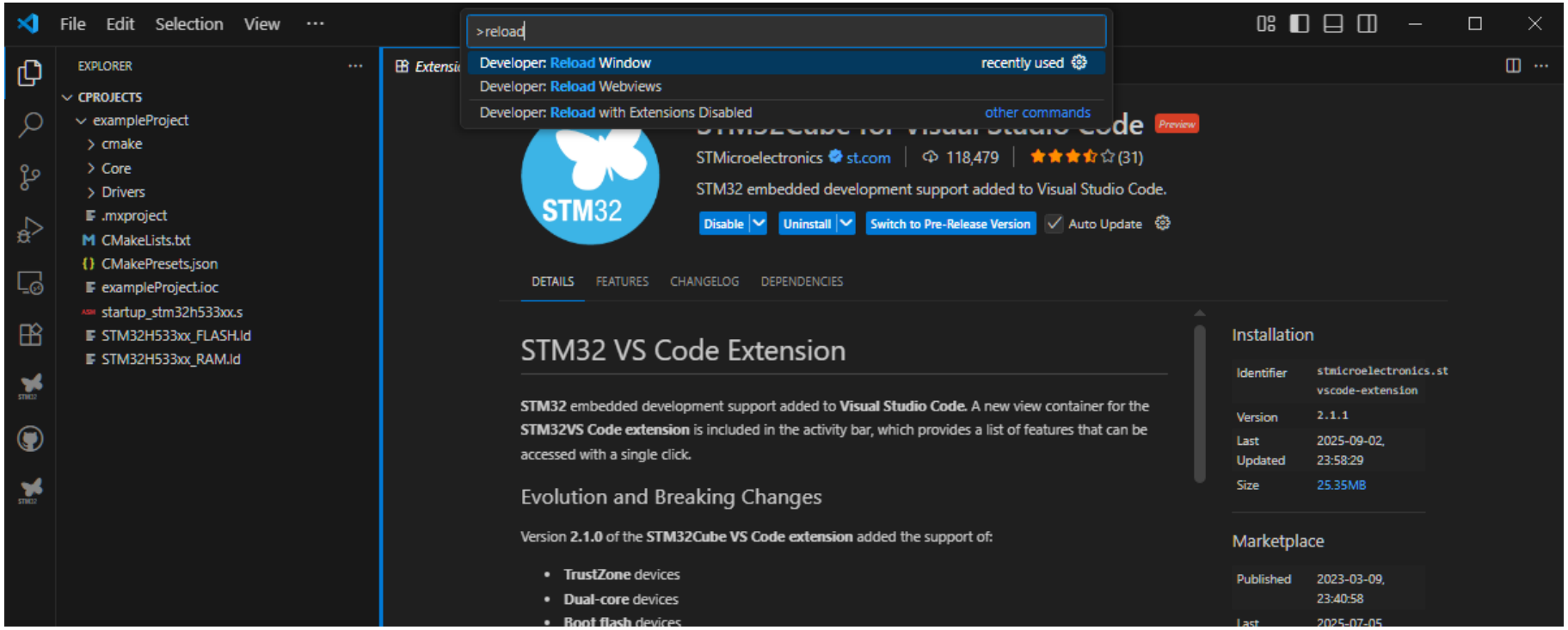






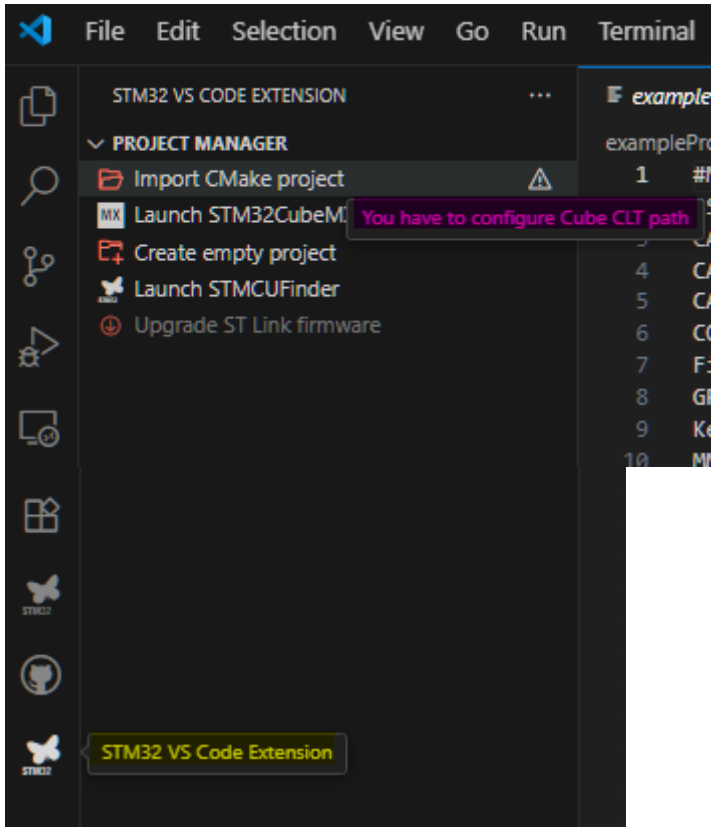
Make a Project with CubeMX

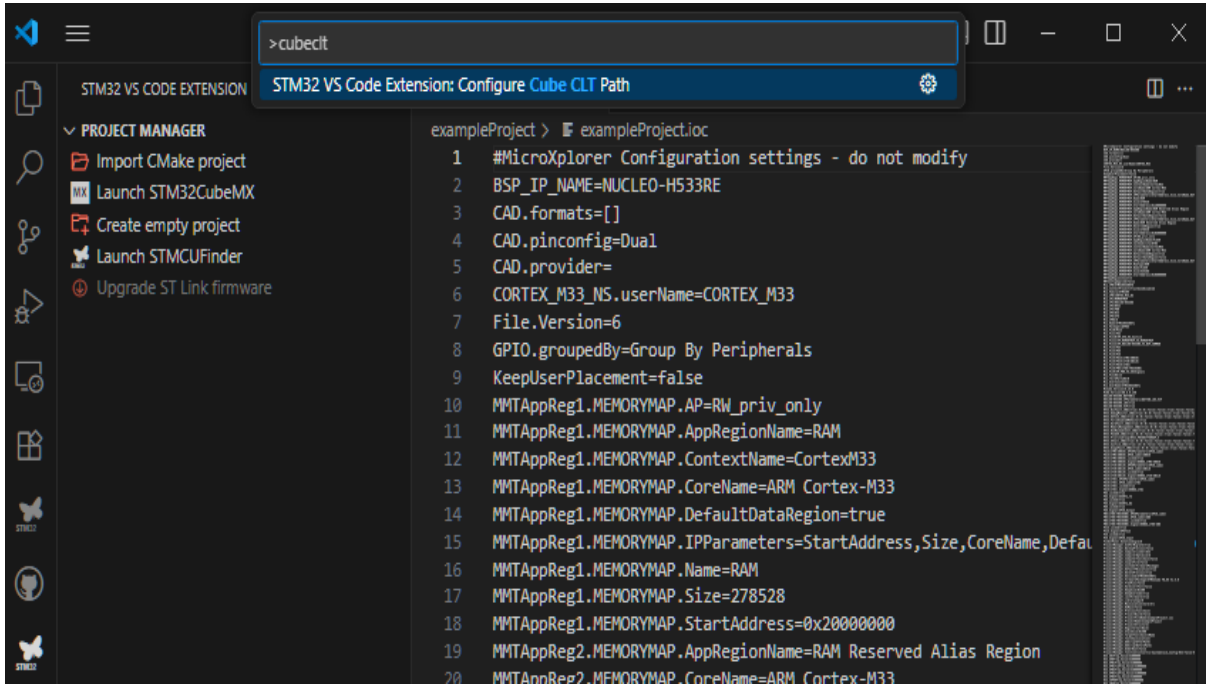
- ✓ Return to VS Code
  - Always reload VS Code when something new is installed
  - Type >reload or press Ctrl + Shift + P and search for “reload



Make a Project with CubeMX

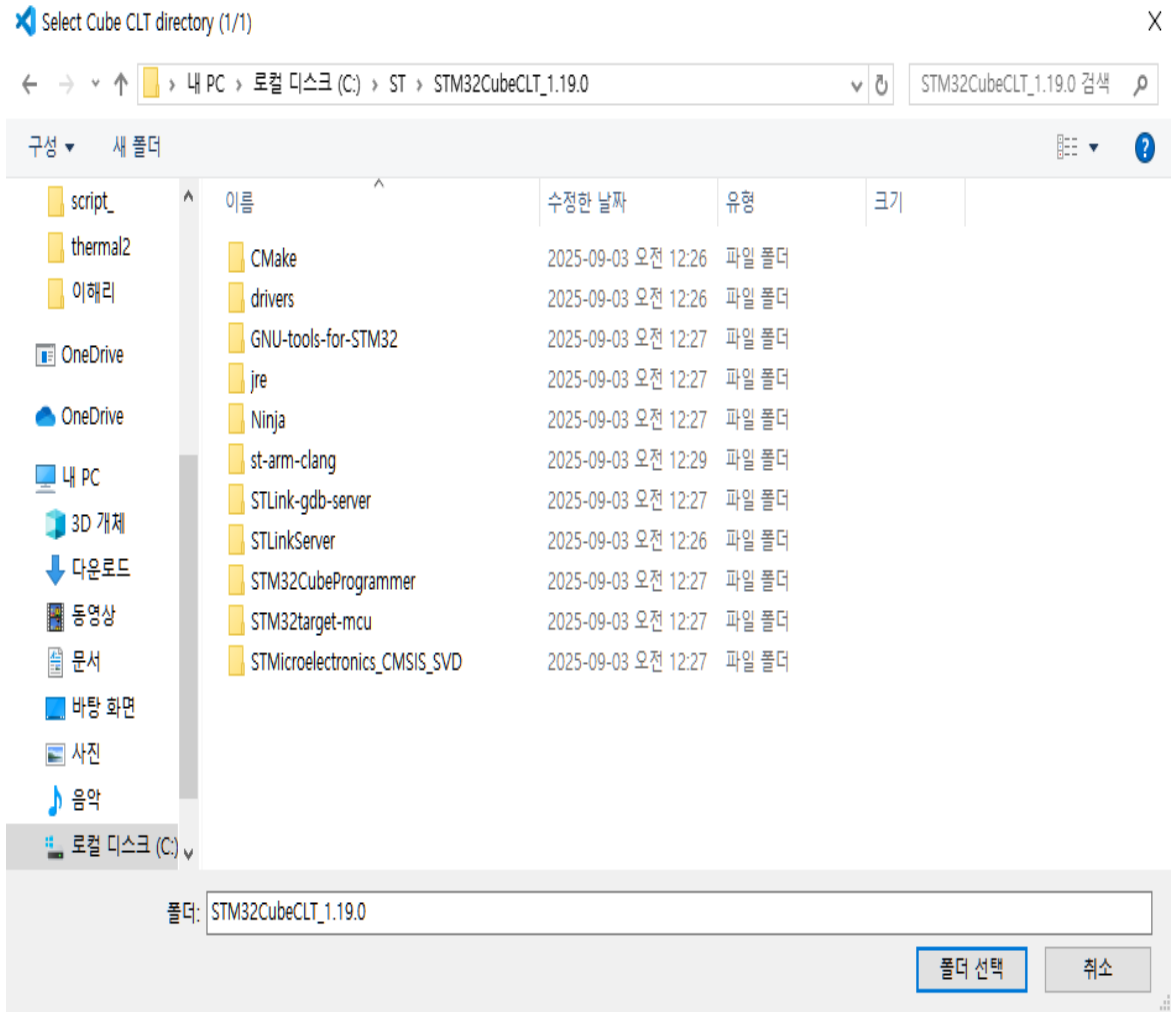
- ✓ In VS Code
  - In menu "STM32 VS Code Extension" Extension, we have to import CMake project but sometimes it does not work at first
  - Similar to reload, search for >cubecIt and configure the path





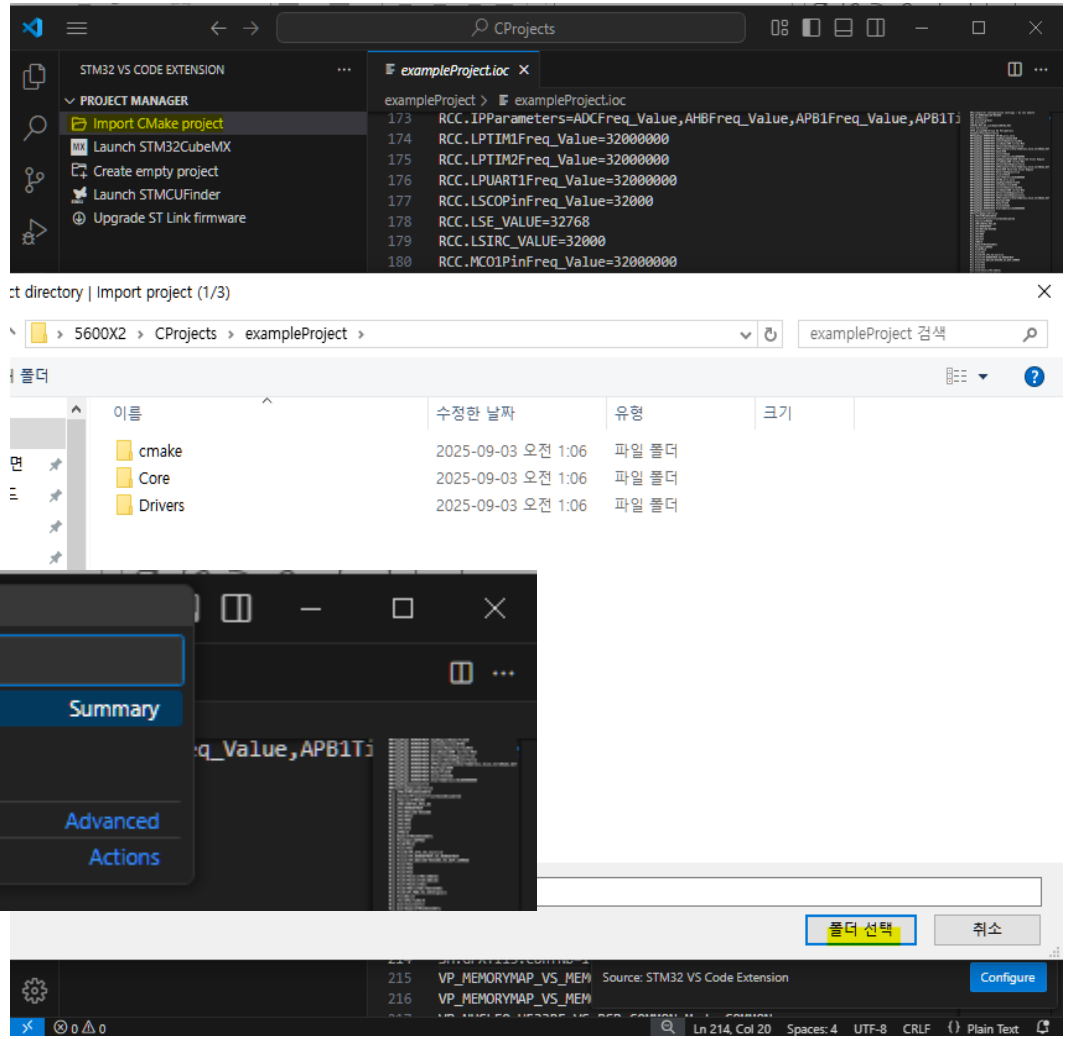
Make a Project with CubeMX

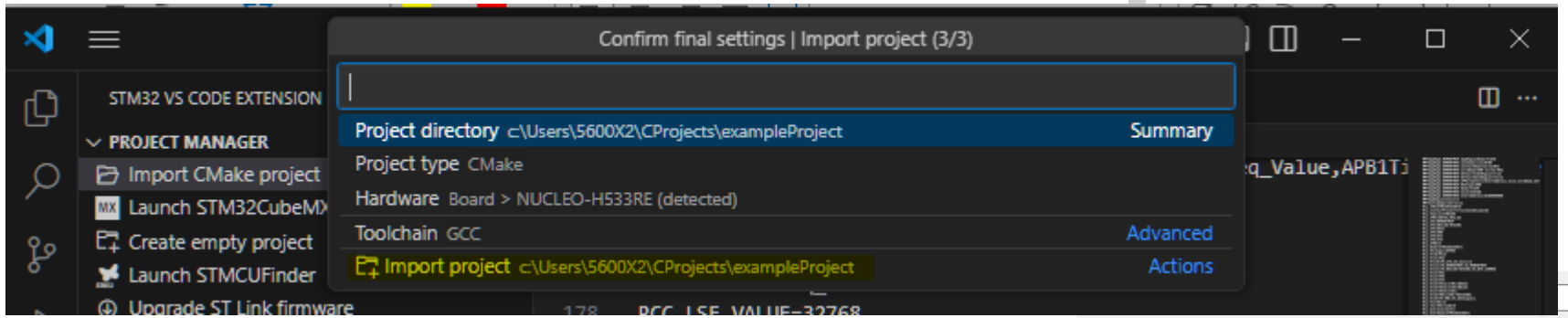
- ✓ In VS Code
  - If you installed CubeCLT successfully,
    you can select C:/ST/STM32CubeCLT_X.XX.X
    (Linux/MacOS : auto-detected)



Make a Project with CubeMX

- ✓ CubeCLT Configured
  - Now you can import your project and VS Code will complete the remaining setup that CubeMX did not

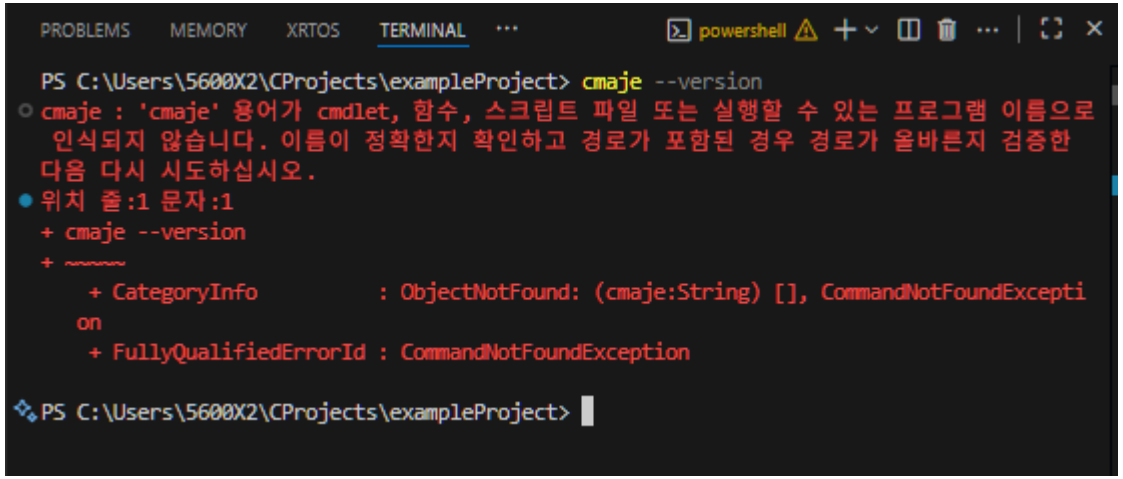


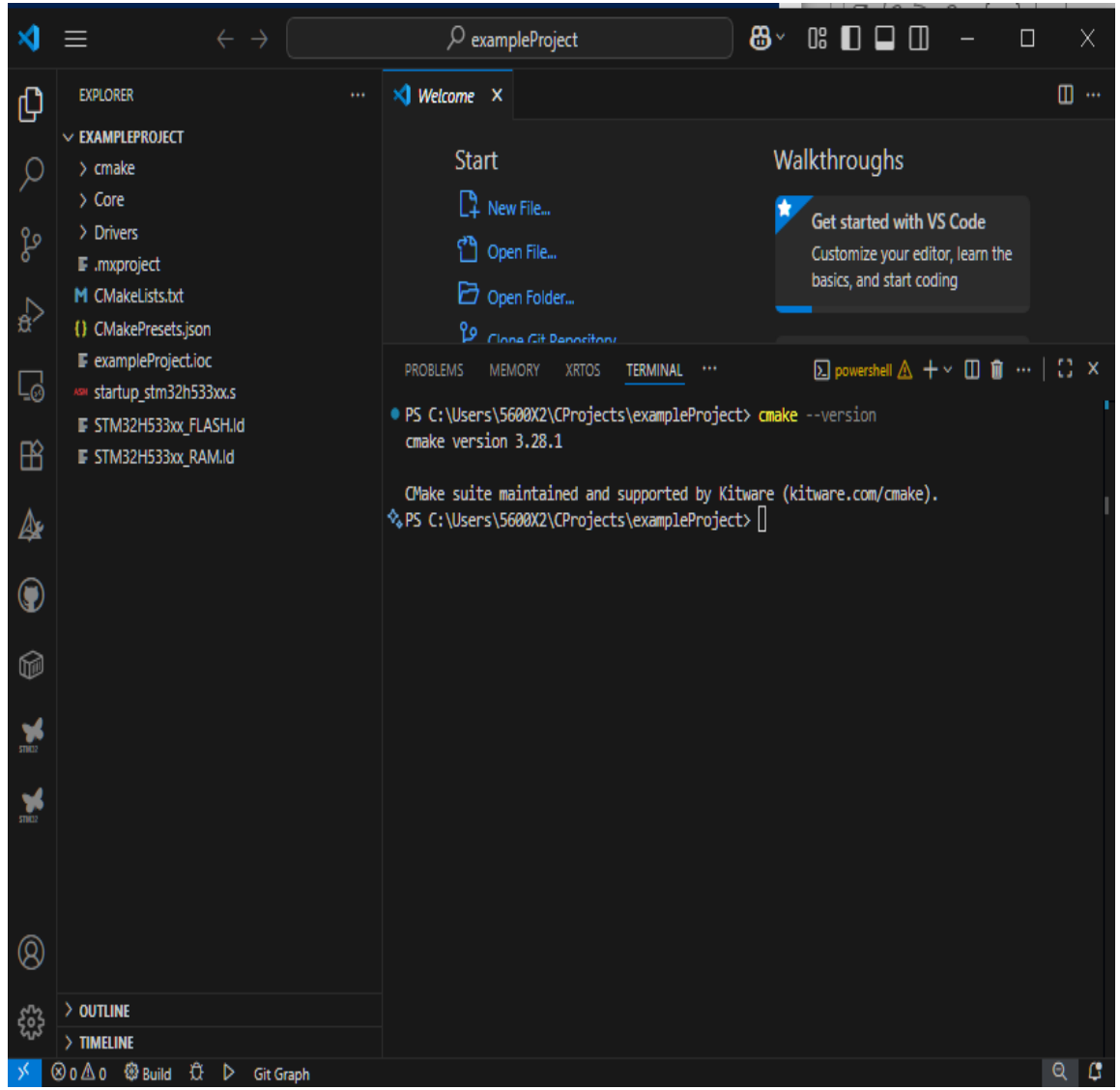


Make a Project with CubeMX

✓ Import CMake Project

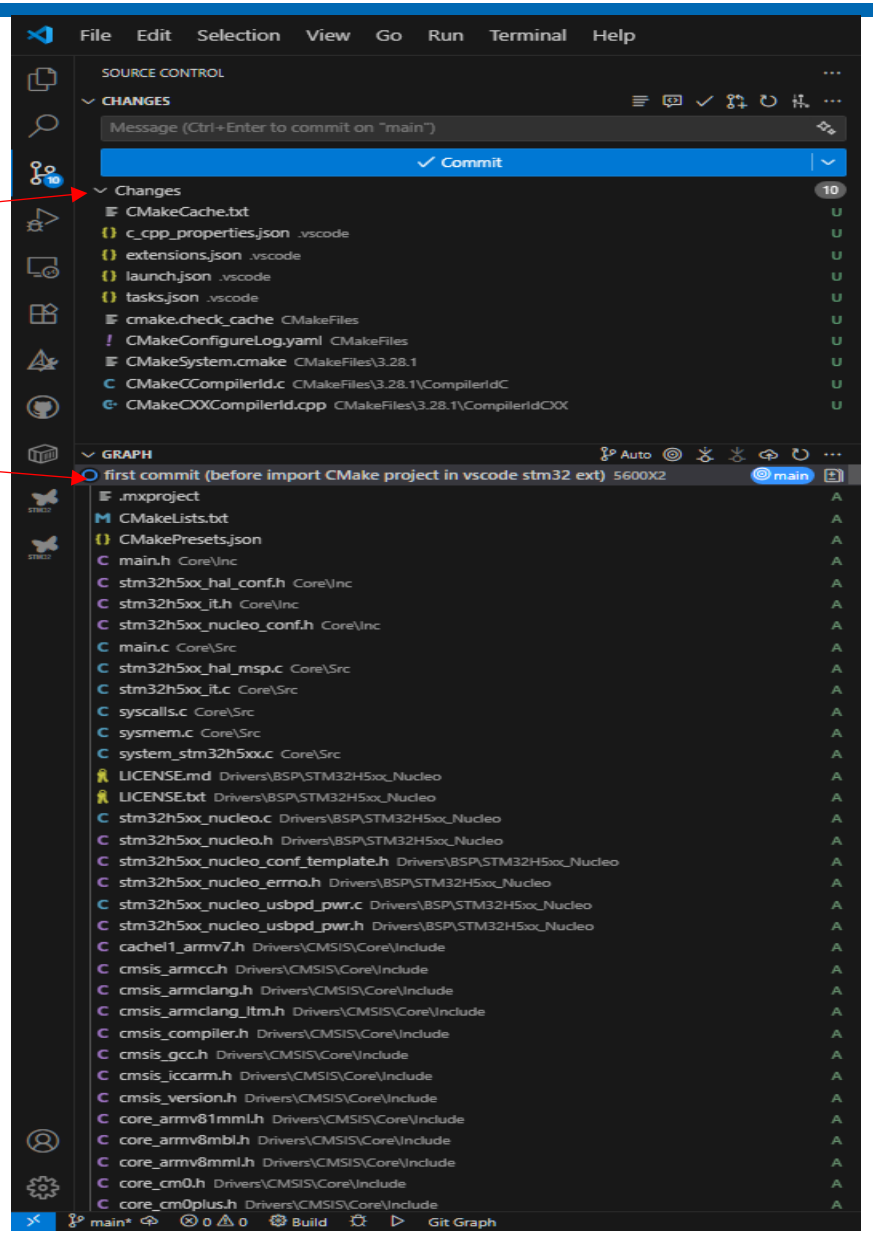
- If “cmake --version” command not works, quit and re-open vscode





Build and Download into MCU

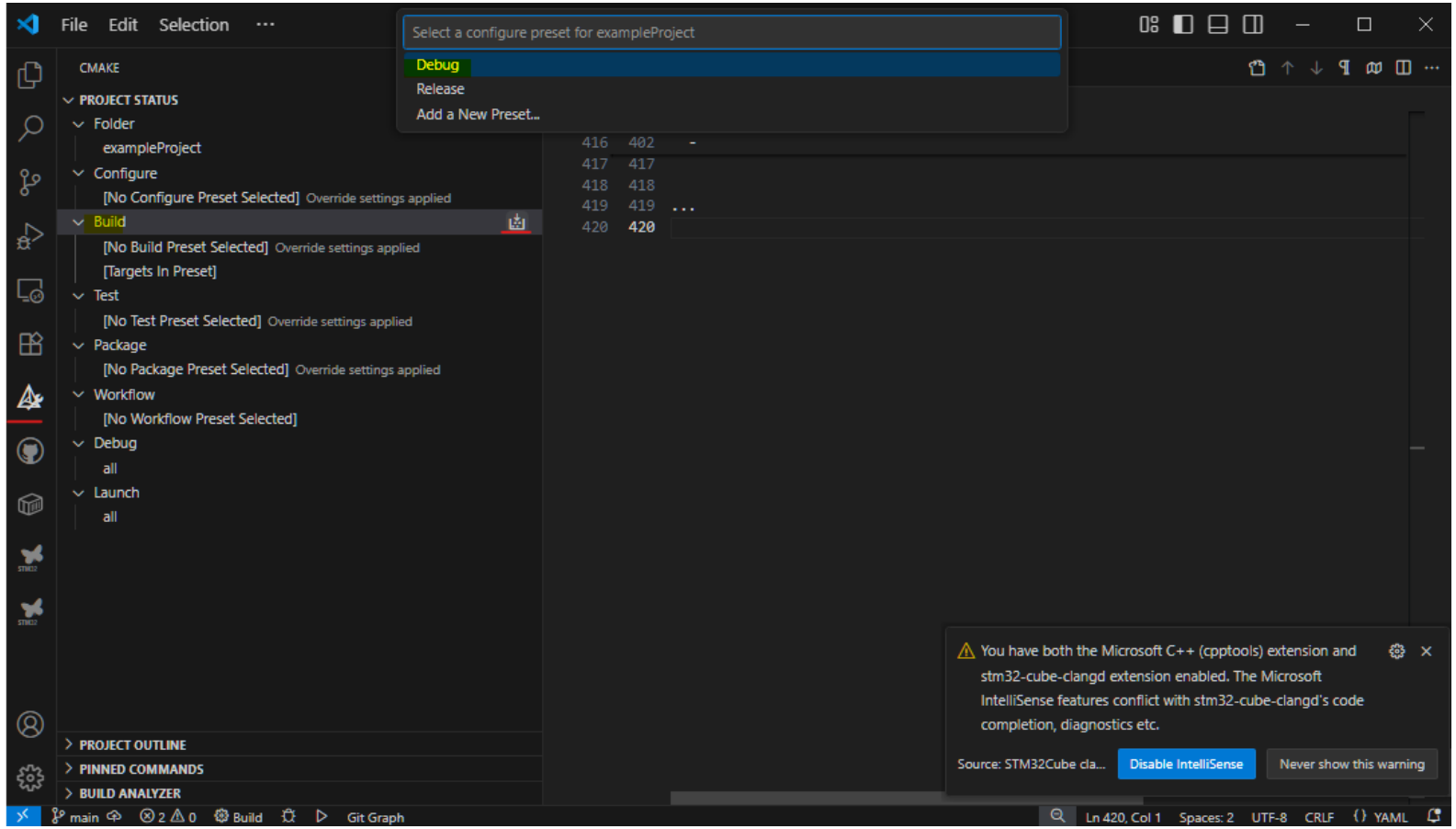
- ✓ CMake Project Imported
  - This is made by VS Code Ext
  - This is made by CubeMX



Build and Download into MCU

✓ Build Test

- Now you can compile whole .c source codes and get .elf binary



Build and Download into MCU

- ✓ Build Test
  - (Compiling messages)
  - They want to disable something, let them do it

PROBLEMS 2 MEMORY XRTOS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filter CMake/Build

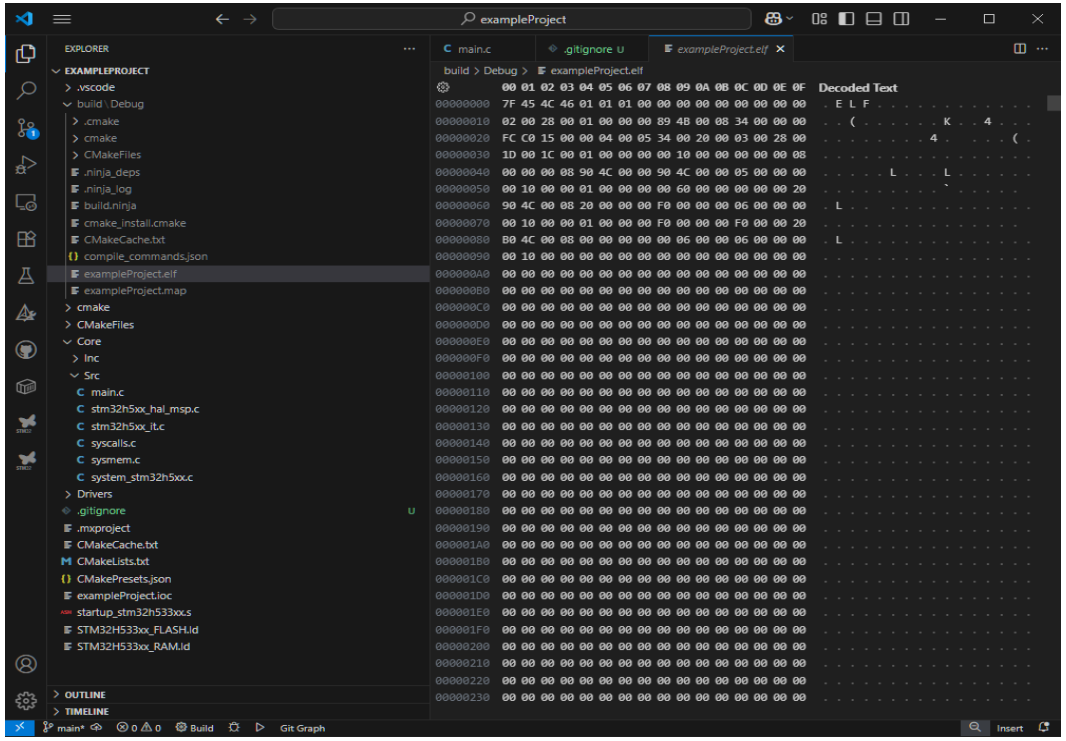
```
[cmake] -- Check for working CXX compiler: C:/ST/STM32CubeCLT_1.19.0/GNU-tools-for-STM32/bin/arm-none-eabi-g++.exe - skipped
[cmake] -- Detecting CXX compile features
[cmake] -- Detecting CXX compile features - done
[cmake] Build type: Debug
[cmake] -- The ASM compiler identification is GNU
[cmake] -- Found assembler: C:/ST/STM32CubeCLT_1.19.0/GNU-tools-for-STM32/bin/arm-none-eabi-gcc.exe
[cmake] -- Configuring done (1.7s)
[cmake] -- Generating done (0.0s)
[cmake] -- Build files have been written to: C:/Users/5600X2/CProjects/exampleProject/build/Debug
[main] Building folder: C:/Users/5600X2/CProjects/exampleProject/build/Debug
[build] Starting build
[driver] NOTE: You are building with preset Debug, but there are some overrides being applied from your VS Code settings.
[proc] Executing command: chcp
[proc] Executing command: C:\ST\STM32CubeCLT_1.19.0\CMake\bin\cmake.EXE --build C:/Users/5600X2/CProjects/exampleProject/build/Debug --
[build] [1/25] Building ASM object CMakeFiles/exampleProject.dir/startup_stm32h533xx.s.obj
[build] [2/25] Building C object CMakeFiles/exampleProject.dir/Core/Src/sysmem.c.obj
[build] [3/25] Building C object CMakeFiles/exampleProject.dir/Core/Src/syscalls.c.obj
[build] [4/25] Building C object CMakeFiles/exampleProject.dir/Core/Src/stm32h5xx_it.c.obj
[build] [5/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.dir/__/__/Drivers/STM32H5xx_HAL_Driver/Src/stm32h5xx_hal_cortex.c.obj
[build] [6/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.dir/__/__/Drivers/STM32H5xx_HAL_Driver/Src/stm32h5xx_hal_flash_ex.c.obj
[build] [7/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.dir/__/__/Core/Src/system_stm32h5xx.c.obj
[build] [8/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.dir/__/__/Drivers/STM32H5xx_HAL_Driver/Src/stm32h5xx_hal_gpio.c.obj
[build] [9/25] Building C object CMakeFiles/exampleProject.dir/Core/Src/main.c.obj
```

⚠ You have both the Microsoft C++ (cpptools) extension and stm32-cube-clangd extension enabled. The Microsoft IntelliSense features conflict with stm32-cube-clangd's code completion, diagnostics etc.

Source: STM32Cube cla... Disable IntelliSense Never show this warning

Build and Download into MCU

- ✓ Build completed messages
  - `[build] Build finished with exit code 0`
  - You can also check the compiled binary (.elf)




```
PROBLEMS   MEMORY   XRTOS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
[build] [8/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.
[build] [9/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers.
[build] [10/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [11/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [12/25] Building C object CMakeFiles/exampleProject.dir/Core/Src/mai
[build] [13/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [14/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [15/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [16/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [17/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [18/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [19/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [20/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [21/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [22/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [23/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [24/25] Building C object cmake/stm32cubemx/CMakeFiles/STM32_Drivers
[build] [25/25] Linking C executable exampleProject.elf
[build] Memory region         Used Size  Region Size  %age Used
[build]             RAM:        1776 B       272 KB      0.64%
[build]           FLASH:       19632 B       512 KB      3.74%
[driver] Build completed: 00:00:01.374
[build] Build finished with exit code 0
```

Build and Download into MCU

- ✓ Download .elf binary into your board
- ✓ (사진추가예정)

Finish

- ✓ You can find my source code and installation video on his Github