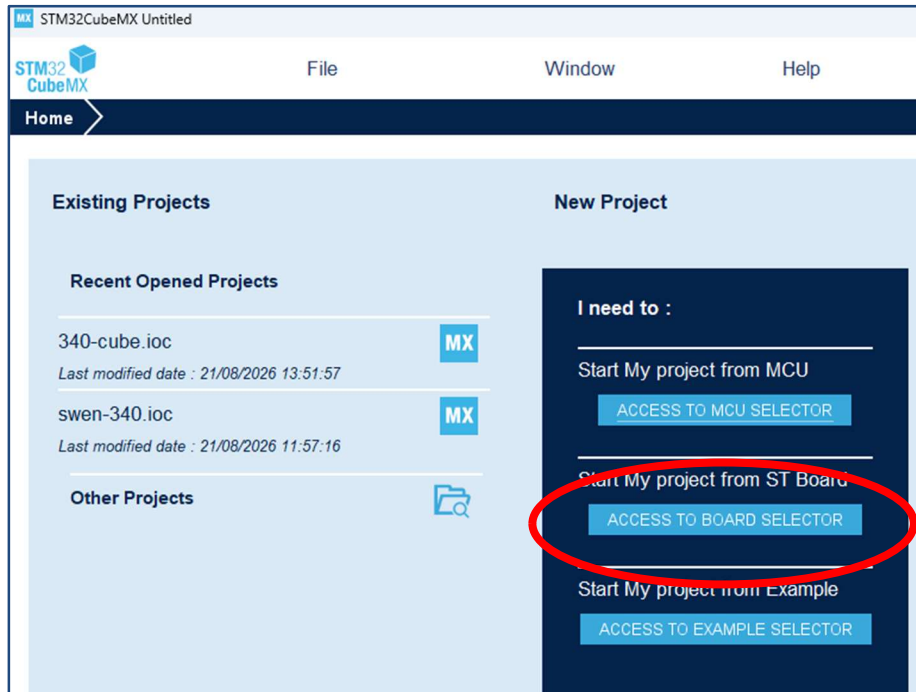
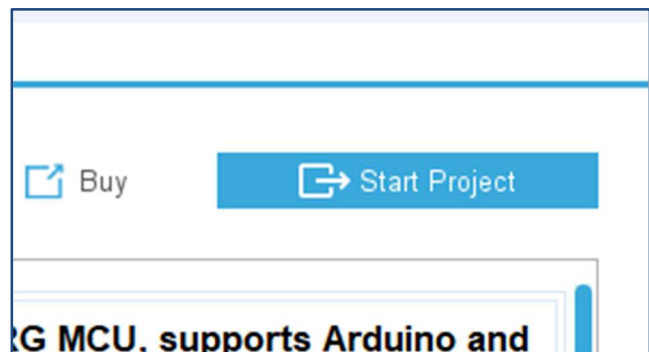
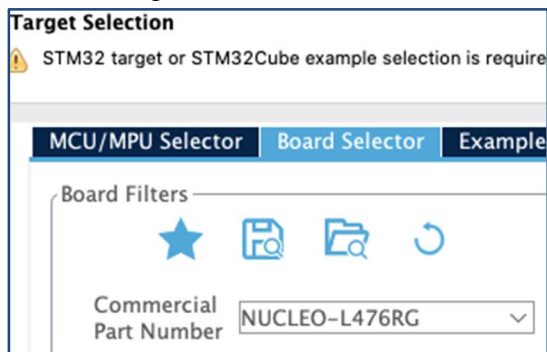


Create our first project

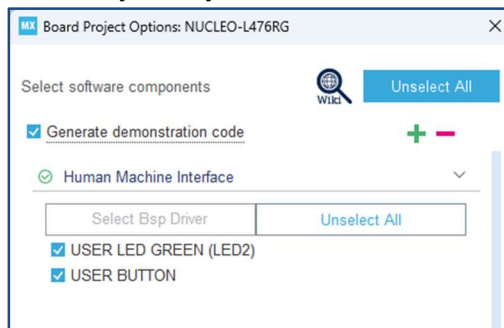
1. Open STM32CubeMX and use:
Start My project from ST Board
ACCESS TO BOARD SELECTOR



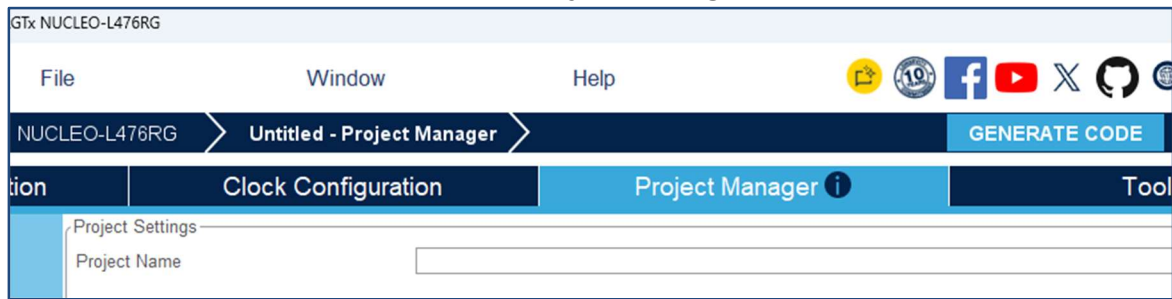
2. Pick our specific microcontroller (NUCLEO-L476RG) from the Board Selector tab. Select the resulting board and then select **Start Project**. *Start Project only becomes available after you select the board in the lower right.*



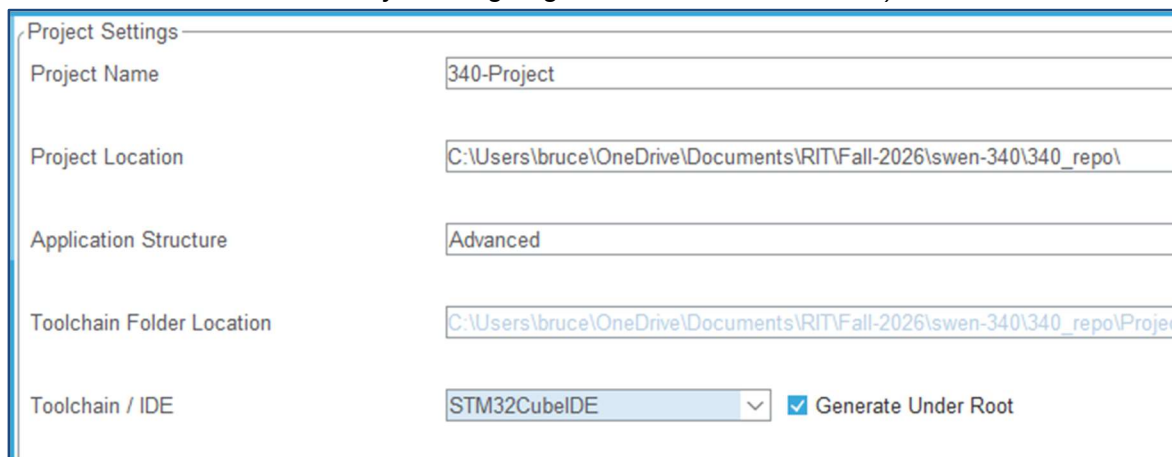
3. On the Board Project Options dialog, it doesn't matter what you select but we'll use the demonstration code very briefly in class.



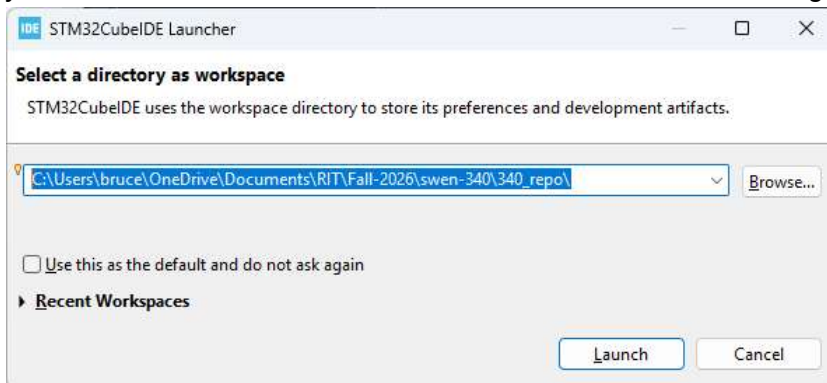
4. Once STM32CubeMX loads, select the **Project Manager** tab.



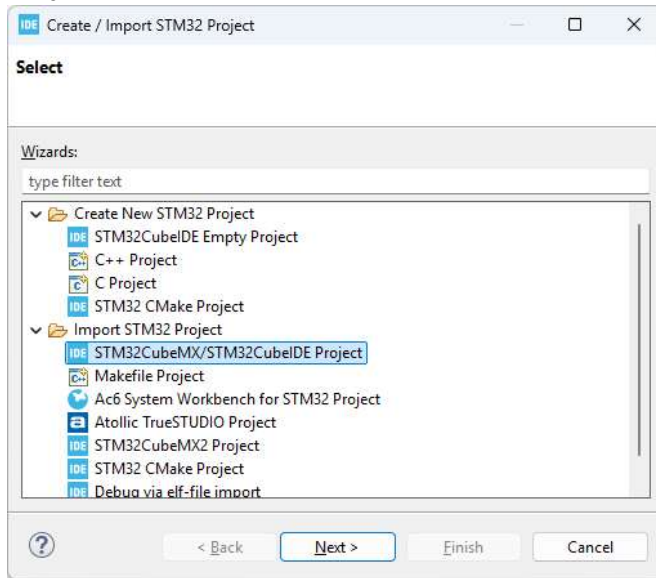
5. Enter a project name, use your course repo for the **Project Location**, and select **STM32CubeIDE** for the **Toolchain / IDE**. Make sure the Generate Under Root option is selected. (Alternatively, you can use the **CMake Toolchain** if you are going to use VS Code Instead.)



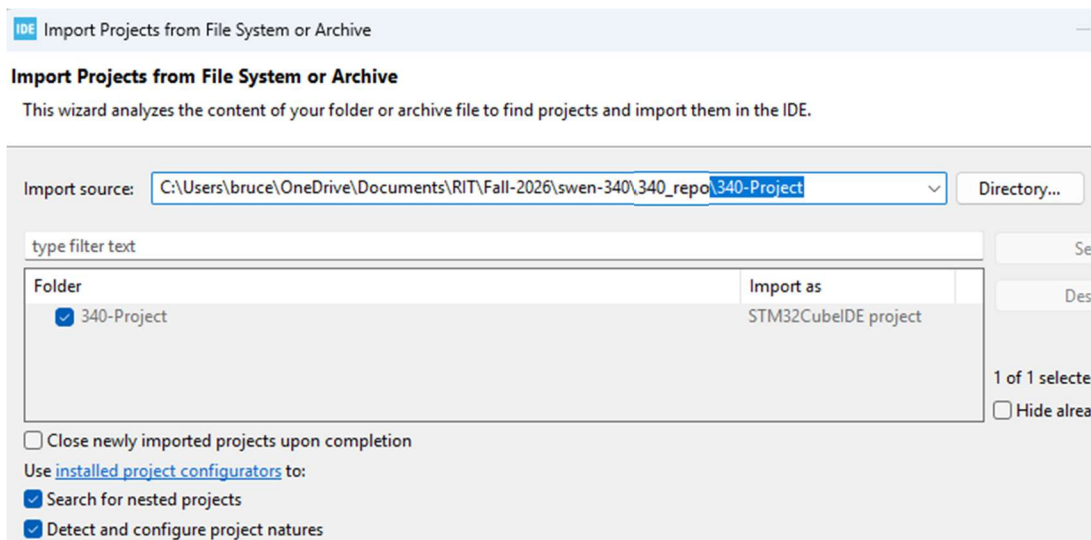
6. Press the GENERATE CODE button in the upper right corner. Accept any prompts.
7. Start STM32CubeIDE. Select the course repo (340_repo) for your workspace. (It is recommended that you do not check the "Use this as the default and do not ask again" checkbox.)



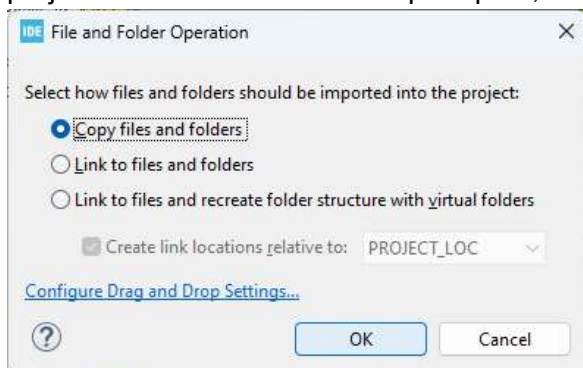
8. Select the Create/Import STM32 Project button and choose the Import32CubeMX/STM32CubeIDE Project option.



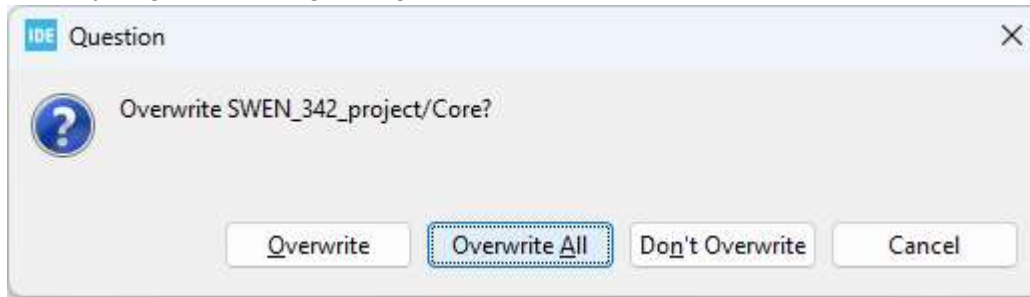
9. Browse to your course repo/project using the **Directory...** button and select the project you just created. Press the **Finish** button.



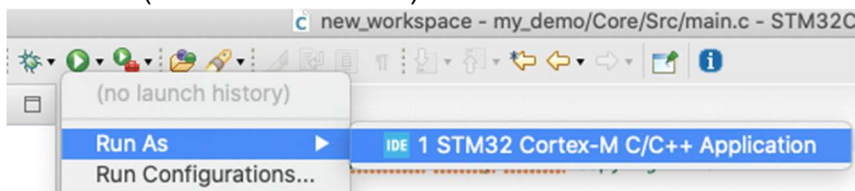
10. Build the project using Project | Build All or the  button. It should build with no errors.
11. Copy **Core** and **Drivers** directories into the project by dragging them directly onto the top item in the project . When prompted, select **Copy files and folders**.



When you get a warning dialog like this, select **Overwrite All**.



12. Build the project. Project > Build Project (hammer icon) (or right click on the project and select from the drop down menu).
13. Select Run or Debug from the icons or the project right-click menu. You may need to select the Run Configurations for the first time (pick what you see below). You may also get an additional **Edit launch configuration properties** dialog concerning your debug or run configuration. If you do get this dialog select OK (the defaults are fine).



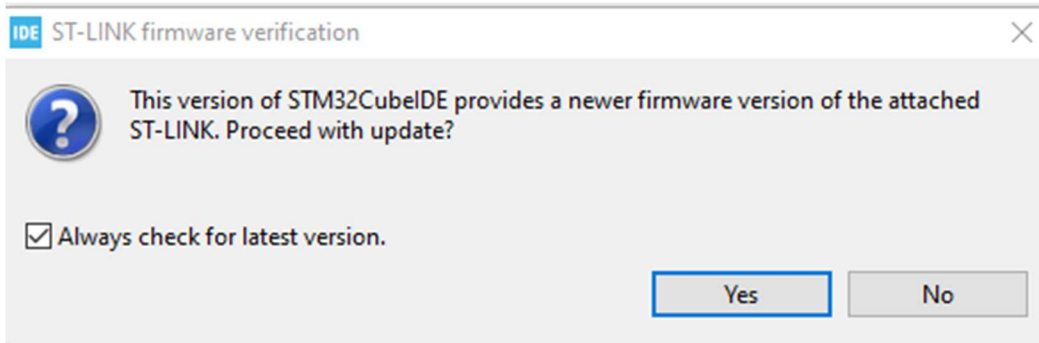
At this point, you have created an executable binary that can run on a board.

- Plug the board into your laptop using a USB cable.
- The board may appear as a file system. You can ignore that.
- You may have to update your ST-Link drivers when you attempt your first download.
- To build, Project>Build - this is the normal thing you will do
The first time through, CubeIDE may ask you if you want to change perspectives. Eclipse (the tool upon which CubeIDE is built) has the notion of perspectives...IDE layouts tailored to different tasks. Writing code and debugging code may use different perspectives. Select 'Switch' and Remember my decision.
- To run/debug, Run>Run As or Run>Debug As (or click Bug Icon).

You can single step your way through the code. You can set breakpoints, examine variables, etc. The goal of this tutorial is to make sure you have a properly functioning development environment, a working board, are able to do simple debugging operations.

Update ST-Link firmware on your microcontroller

When you download software to the board, your IDE may require an ST-LINK update to your NUCLEO board. This is OK. It should happen only once.



Clicking Yes, the STLink Upgrade utility will ask to open the ST-LINK device in update mode.

Open in update mode, do not Change Type, click Upgrade.

Close the ST-LINK Upgrade utility.

If you were building the app for debug, go back and restart those instructions.