

Tutorial 3: Compiling, and Deploying your Code

Instructor:

Dr. Carl Latino

carl.latino@okstate.edu

Graduate Teaching Assistants:

Francisco E. Fernandes Jr.

feferna@okstate.edu

School of Electrical and Computer Engineering

Oklahoma State University

Fall 2019





Compiling your code with STM32CubeIDE

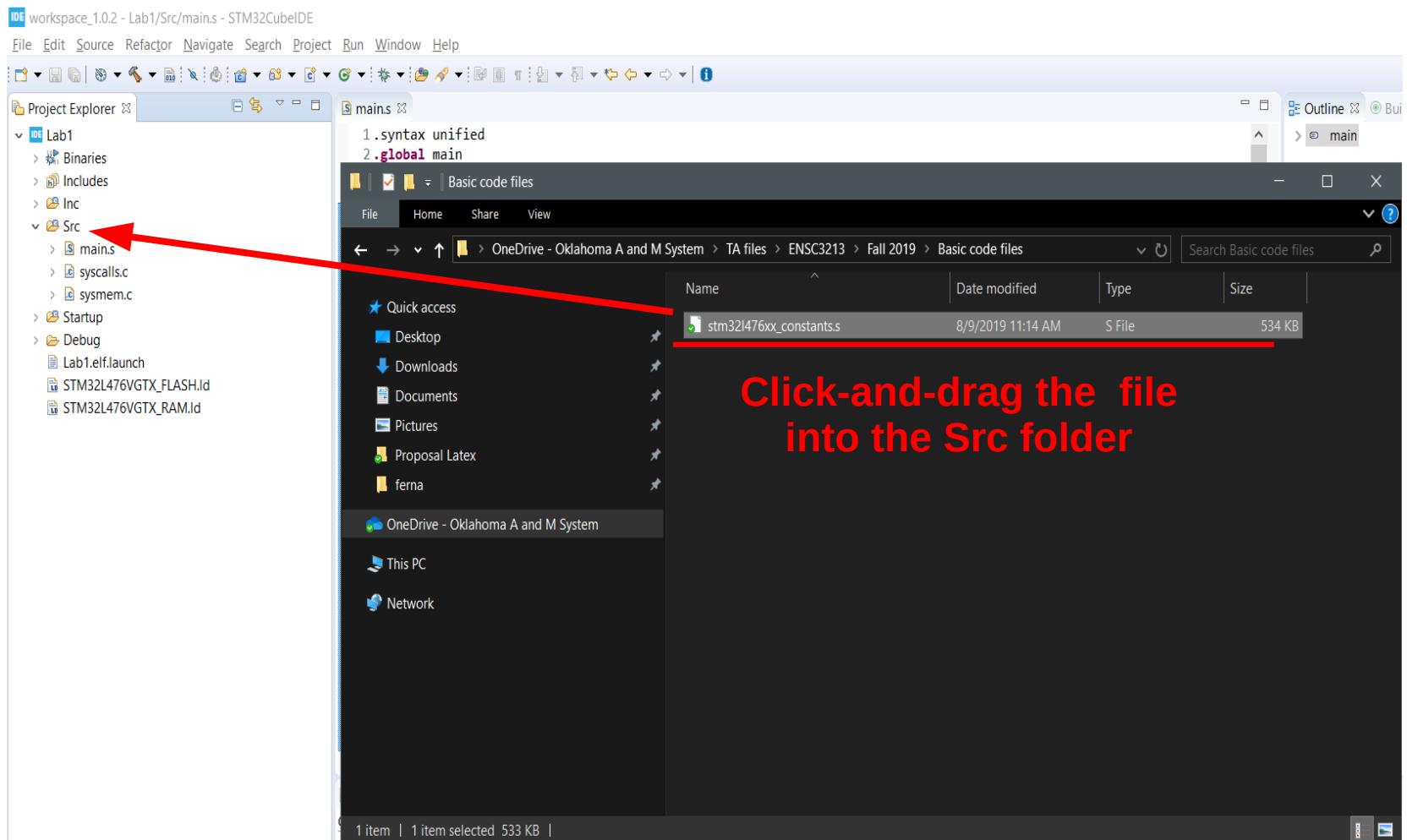
- Remember to always add the following three lines of code to the top of your **main.s** file:

```
.syntax unified
.global main
.include "stm321476xx_constants.s"
```



Compiling your code with STM32CubeIDE

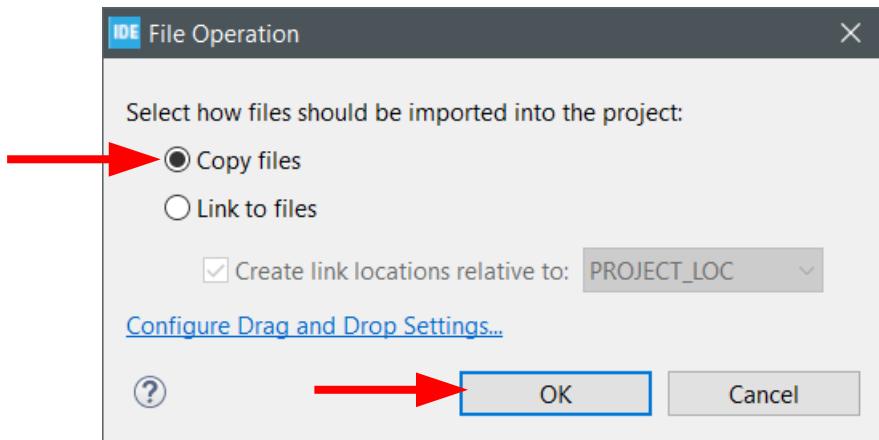
- You will also need to add the supplied **stm32l476xx_constants.s** file (this file can be downloaded on Canvas) to your **Src** folder!





Compiling your code with STM32CubeIDE

- You will also need to add the supplied **stm32l476xx_constants.s** file (this file can be downloaded on Canvas) to your **Src** folder!

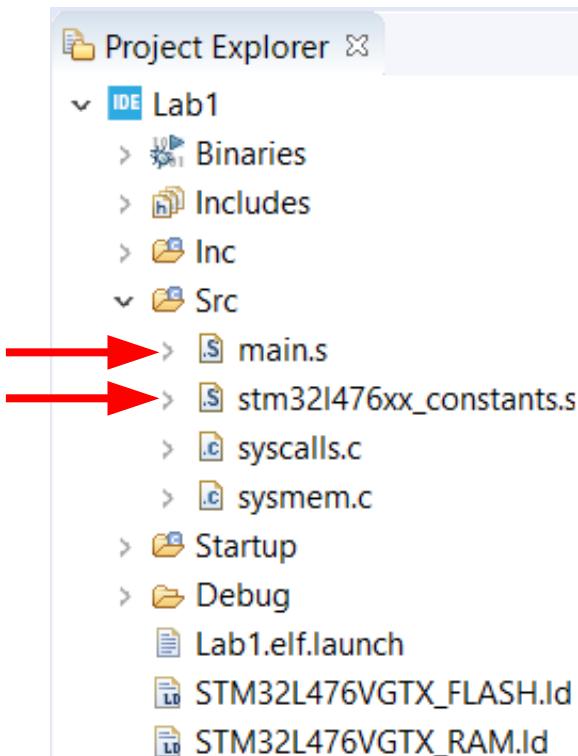


Select **Copy files**, and
click on **OK**



Compiling your code with STM32CubeIDE

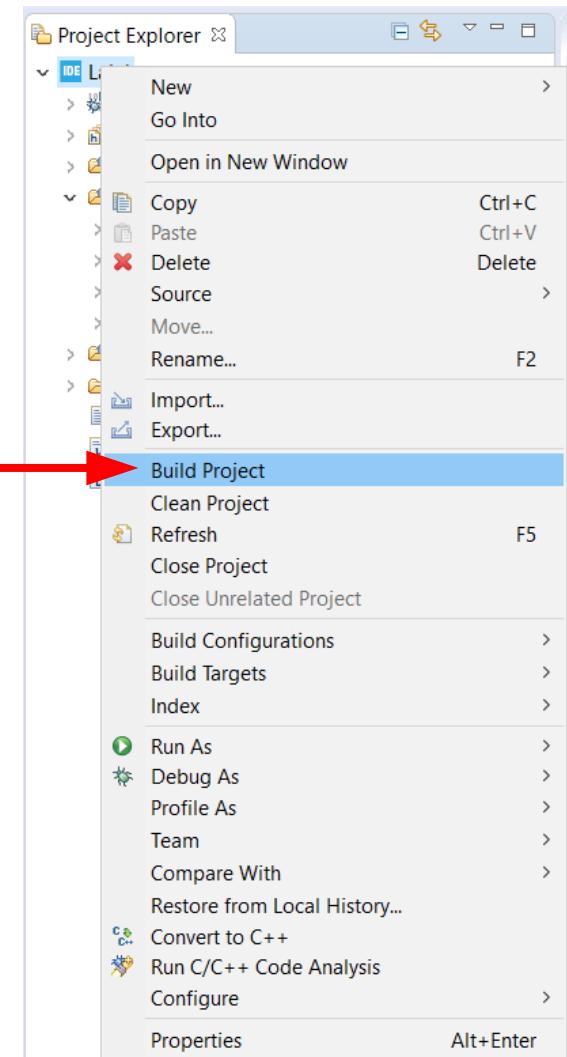
- Once you have YOUR **main.s** file and the **stm32l476xx_constants.s** file, you will be able to compile your code.



**Make sure you have
these two files before
compiling!**

Compiling your code with STM32CubeIDE

- To compile your code, right-click on your project name and select **Build Project**.





Compiling your code with STM32CubeIDE

- If your code is compiled correctly, you will see the message indicated by the arrow with **zero errors**.
- Now, you can send your code to the **development board**.

The screenshot shows the STM32CubeIDE interface with the following details:

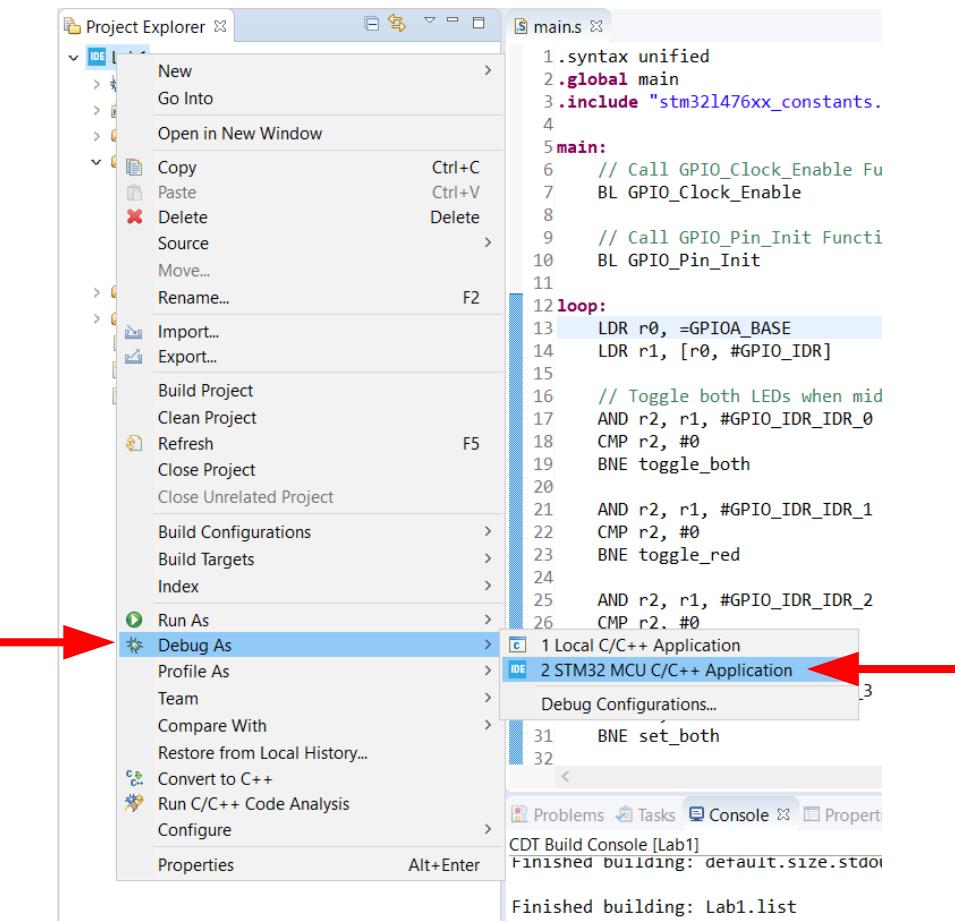
- Project Explorer:** Shows the project structure for "Lab1" with files like "main.s", "stm32l476xx_constants.s", and "syscalls.c".
- Code Editor:** Displays the assembly code for "main.s".
- Build Analyzer:** Shows the build log: "Finished building: Lab1.list" and the message "11:30:58 Build Finished. 0 errors, 0 warnings. (took 723ms)".
- Build Analyzer (Memory Details):** A table showing memory usage:

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000000	0x20018000	96 KB	94.47 KB	1.53 KB	1.60%
ROM	0x08000000	0x08100000	1024 KB	1022.94 KB	1.06 KB	0.10%

A red arrow points to the "Build Finished" message in the build log.

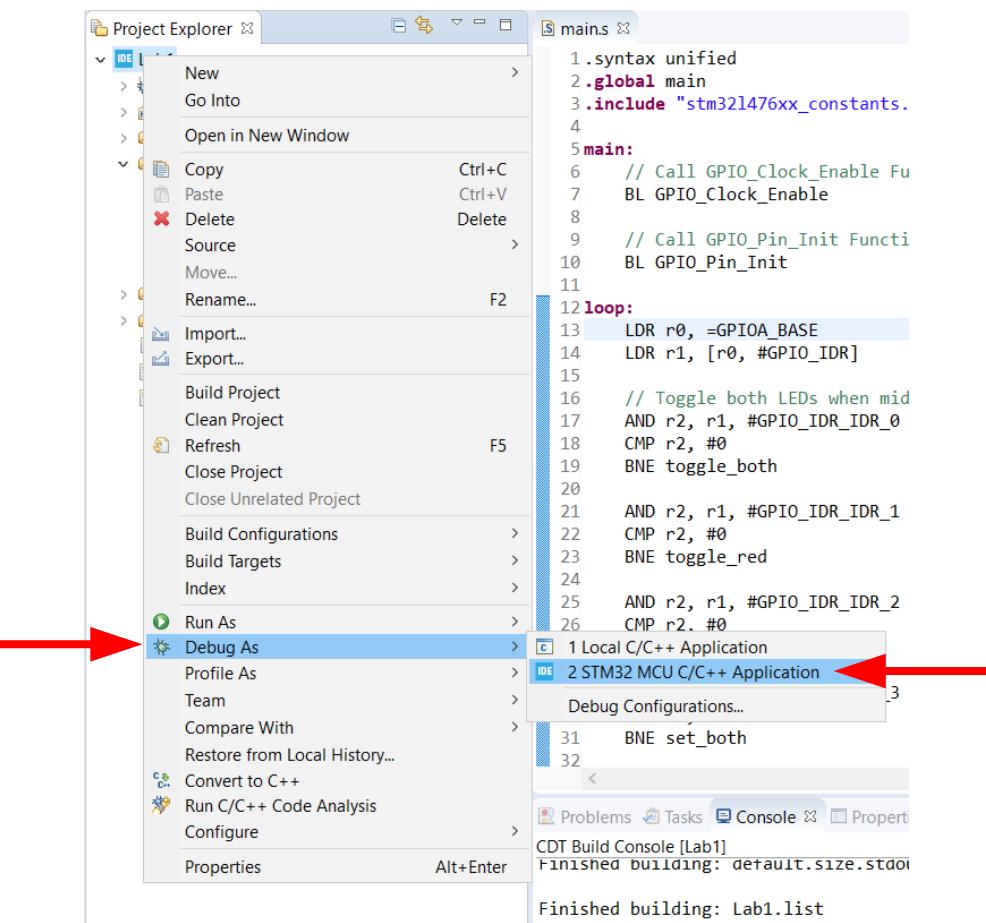
Deploying your code

- The process of sending your code to the development board is also known as **deployment or upload**.
- Right-click on your project name, and select **Debug as → STM32 MCU C/C++ Application**.



Deploying your code

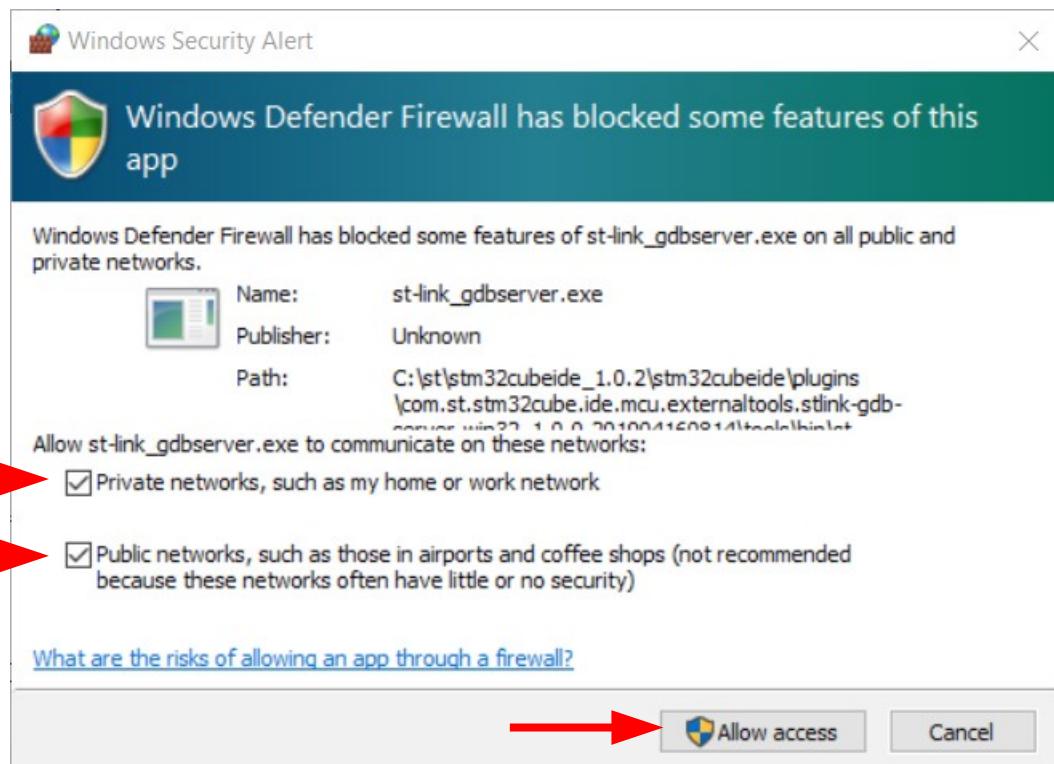
- The process of sending your code to the development board is also known as **deployment** or **upload**.
- Right-click on your project name, and select **Debug as** → **STM32 MCU C/C++ Application**.



Make sure your development board is connected to your PC!

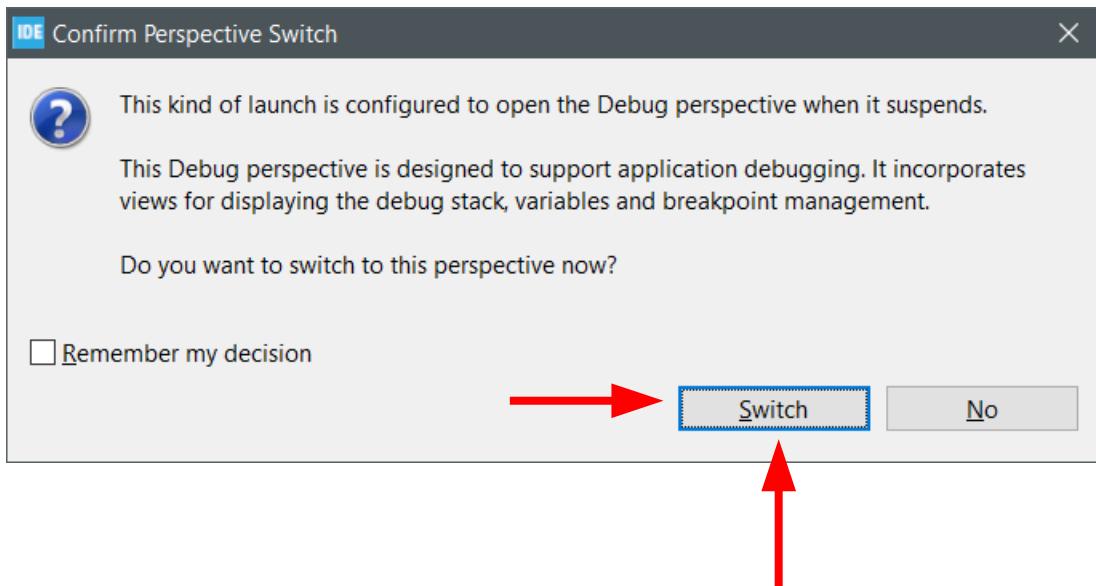
Deploying your code

- If this is your first time deploying your code, the Windows Firewall will ask your permission to connect to the network.
- Make sure to **allow access** in the new screen that will show up!



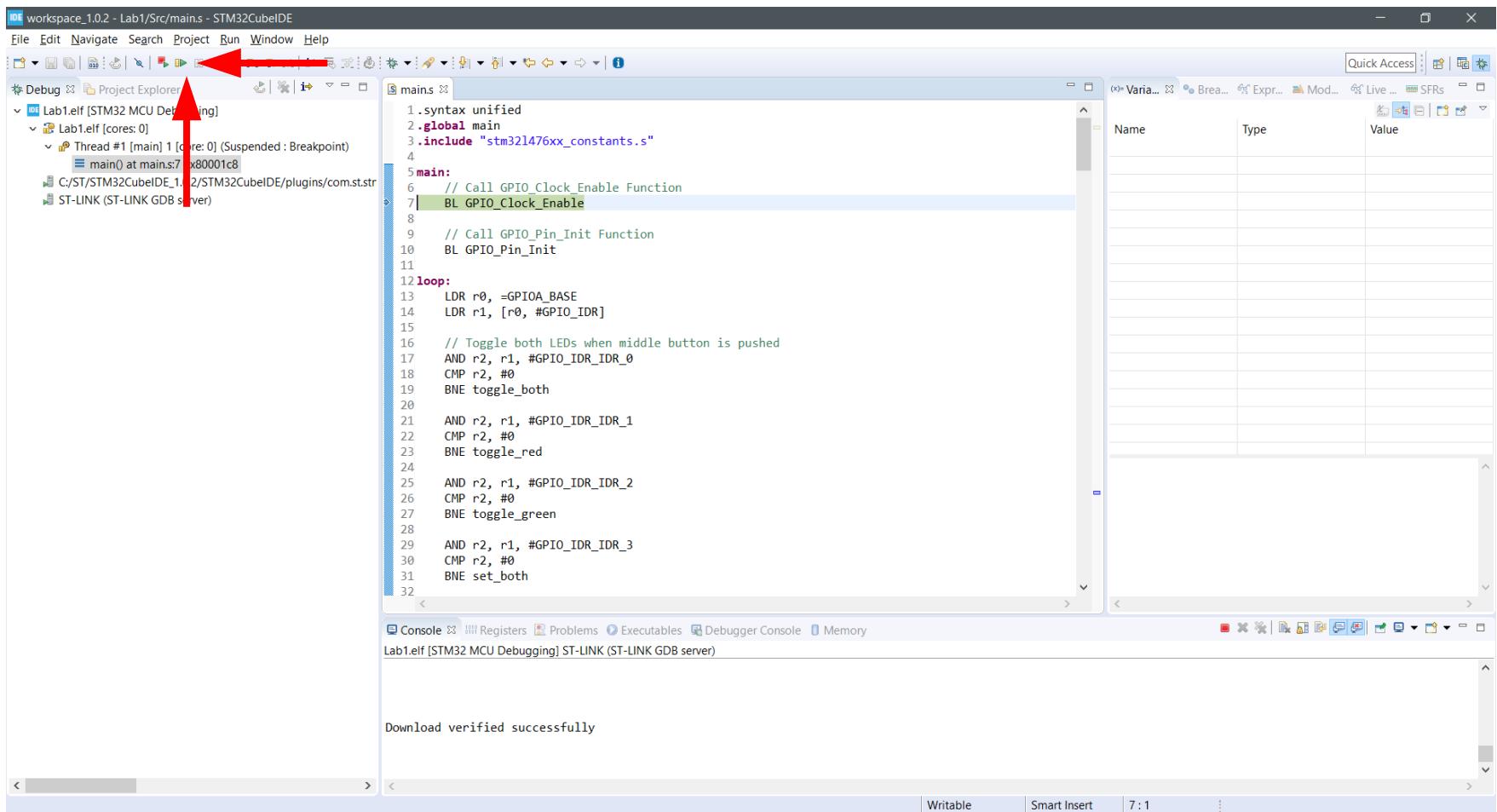
Deploying your code

- A window will pop-up asking if you want to switch to the **Debug** view. Click on **Switch**.
- Don't panic, but your IDE will look different now! See the next pages.



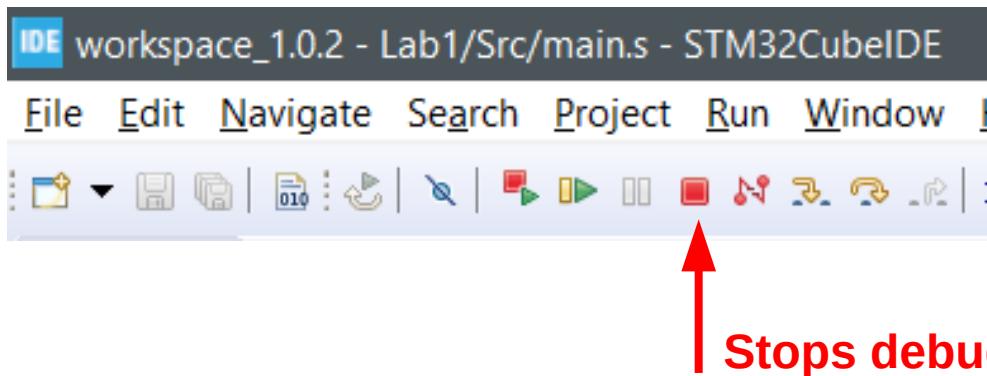
Deploying your code

- This is the **Debug** view of the IDE. Your code is on the **development board**, but it is **not running yet**.
- To run the code, just click in the icon indicated by the arrow. It looks like a green **play** button.



Deploying your code

- Your code is now running in the development board.
- Once you finish testing your code in the board, click on the red square to go out of the debugging mode, and back to the normal coding mode.



- The next tutorial (Tutorial 4) shows how you can use the debugging mode to find errors in your code.