**Getting Set Up with STM32F769I-DISCO**

**Part 1: IDE Installation**

1. Go here:

<http://www.openstm32.org/System+Workbench+for+STM32>

Register with OpenSTM32.org, and then go back to that page once validated & logged in.

1. Click “Installing System Workbench for STM32” in the “Table of Contents,” and then click “Installing System Workbench for STM32 with installer” (if you’re familiar with Eclipse and already use it, go ahead and use “Installing System Workbench for STM32 from Eclipse” if you want, however that option will not be supported here).
2. Click “Downloading the System Workbench for STM32 installer” and open it in a new tab, and download the appropriate file for your operating system (Windows 8/10 users, use the Windows 7 installer; it’s compatible).
3. Run the executable/extract the archive and run the executable (depending on whether you downloaded an executable or an archive; it doesn’t matter). Install to the default location, as STM32 Workbench has issues with spaces in folder names if you aren’t careful.

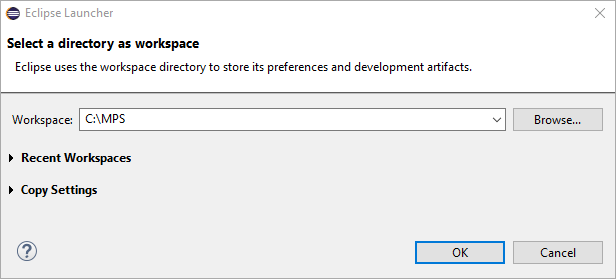
Note to Windows users: Make sure the STLINK driver is checked for installation, otherwise nothing will work when you try to upload your code!

1. Once installed, run STM32 Workbench, and move on to the next section.

**Part 2: Set up the Workspace for MPS**

1. When running STM32 Workbench for the first time, you will be prompted for a workspace. For your workspace, select the folder you’ll be using for MPS labs.[[1]](#footnote-1)

Notice: Better not to have spaces in any folder name, unless you’re willing to futz around with various file directory settings. In other words, C:\MPS\_STM32 and /Users/[name]/Documents/eclipse-labs are safe, but C:\Program Files\... may not be.

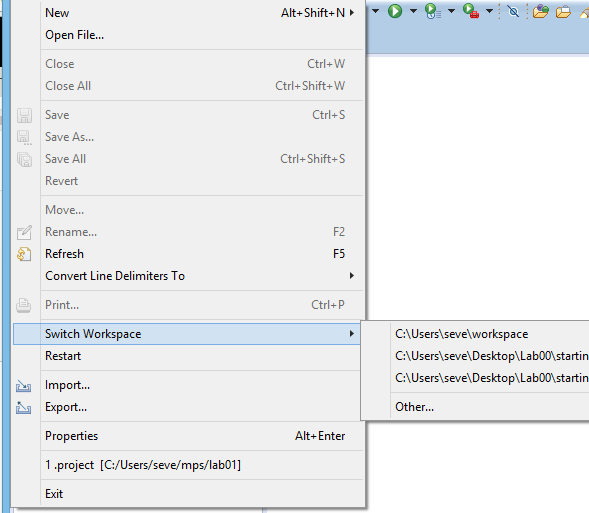


**Figure 1: Selecting a Workspace**

1. Check “Use this as the default and do not ask again” option (not shown in above figure) and click “OK.” You can now go on to Part 3.

*If for some reason you are not prompted for a workspace on first launch:*

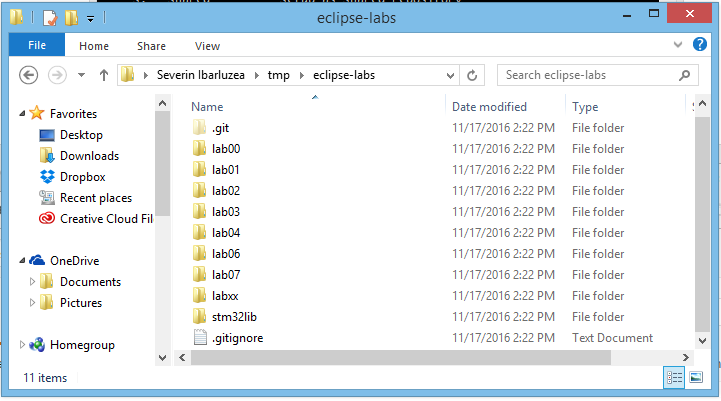
1. Open the IDE. Click File 🡪 Switch Workspace 🡪 Other...



**Figure 2: File 🡪 Switch Workspace 🡪 Other...**

1. Select the folder that you’ll be using for MPS labs.[[2]](#footnote-2) An example called “eclipse-labs” is shown in Figure 3.

As mentioned before, try to avoid folder names with spaces, unless you’re willing to mess around with some settings since STM32 Workbench [on Windows, at least] doesn’t like folders with spaces in them without quotes.



**Figure 3: Sample MPS Main Folder**

The IDE should now restart in the MPS folder you gave it.

1. IMPORTANT:

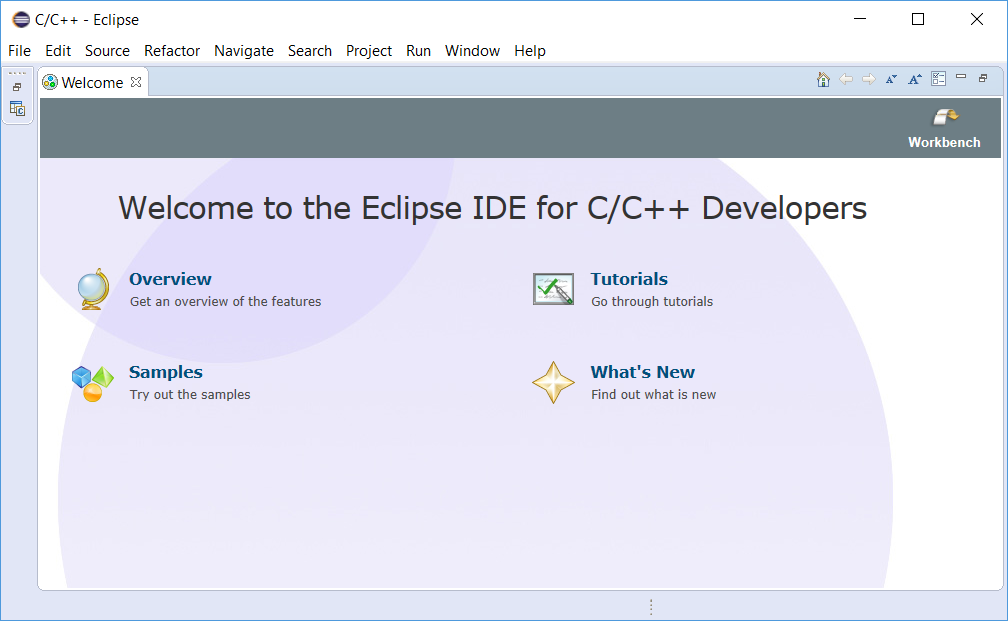
Before moving on, make sure that you have the “stm32lib” folder in your workspace folder as in the above screenshot. This is the folder that contains all the headers and source files for the STM32 board, and all the Workbench labs have been configured to expect this folder at the base of your workspace. Your code simply won’t compile without it.

If you don’t have the stm32lib folder, you need to get it from either <https://github.com/rpi-mps/Workbench-Labs> or your professor’s file hosting area (if it is being hosted there).

Once you’re all set with the stm32lib folder, you can go on to Part 3.

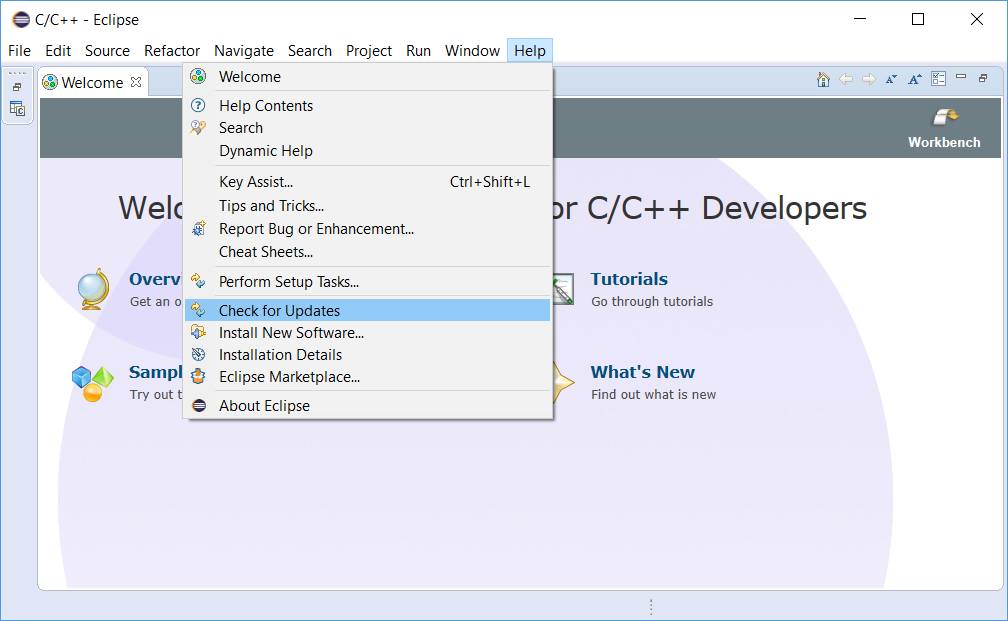
**Part 3: Updating the IDE**

STM32 Workbench needs to be updated to ensure it has the correct files and configuration settings for the STM32F769I-Discovery kit. Upon launching the IDE, you should see the below interface (Figure 3):



**Figure 4: STM32 Workbench Welcome Screen**

Now click Help 🡪 Check for Updates to update your IDE.



**Figure 5: Help --> Check for Updates**

Once updated, from here on you’ll only need to do Parts 4 and 5 for the labs.

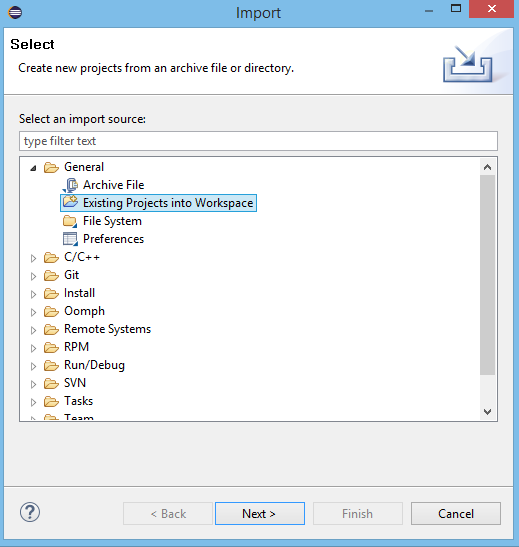
**Part 4: Importing Labs**

1. Download the lab folder for the lab you’re doing from:

<http://www.rpi.edu/dept/ecse/mps/mps_handouts.html>

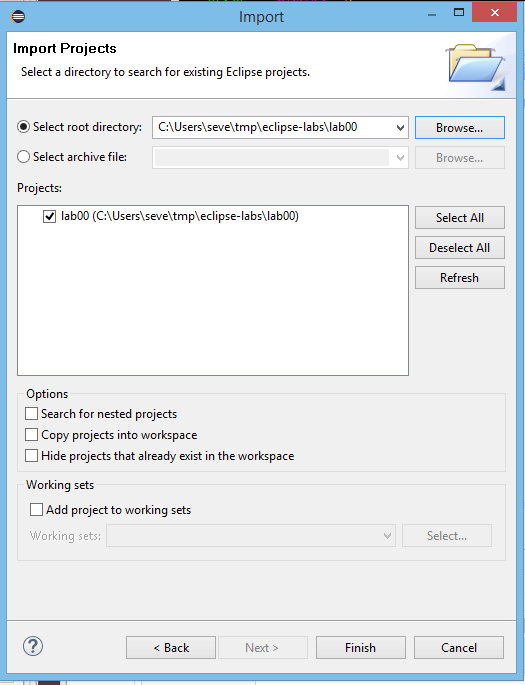
And extract it to the folder that you will be using for MPS labs. In this example, Lab00 ANSI Terminal will be used.

1. Go to File 🡪 Import, expand “General,” and click “Existing Projects into Workspace.”



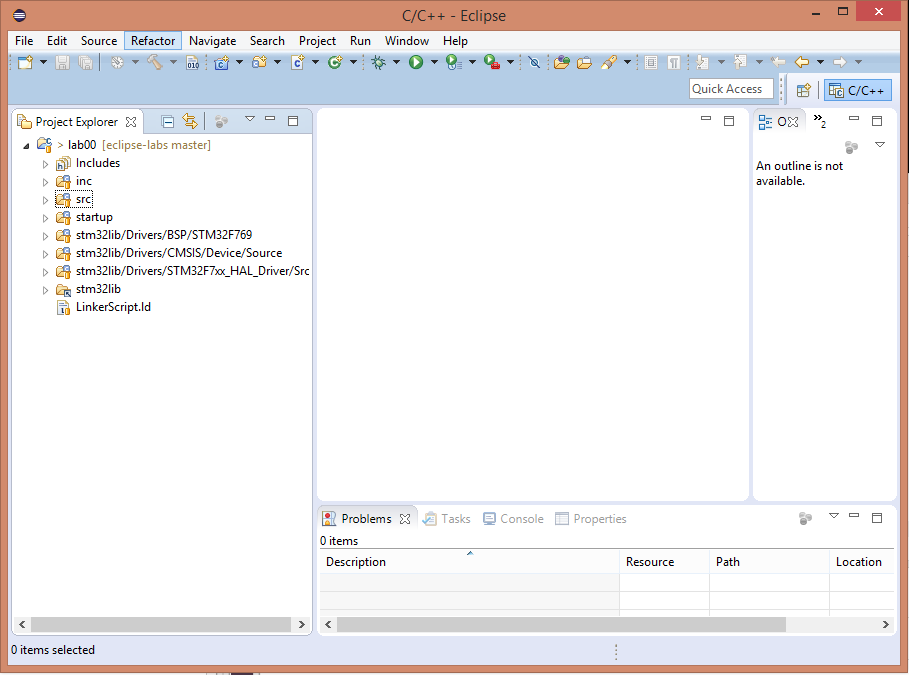
**Figure 6: Import Existing Projects into Workspace**

1. On the next dialog, click “Browse” next to “Select root directory” and select the lab folder you’re using for the specific lab (e.g. lab00, lab02, etc. from Figure 3).
2. The dialog should look something like Figure 6 once you’ve selected your lab:



**Figure 7: Import Lab Root Directory**

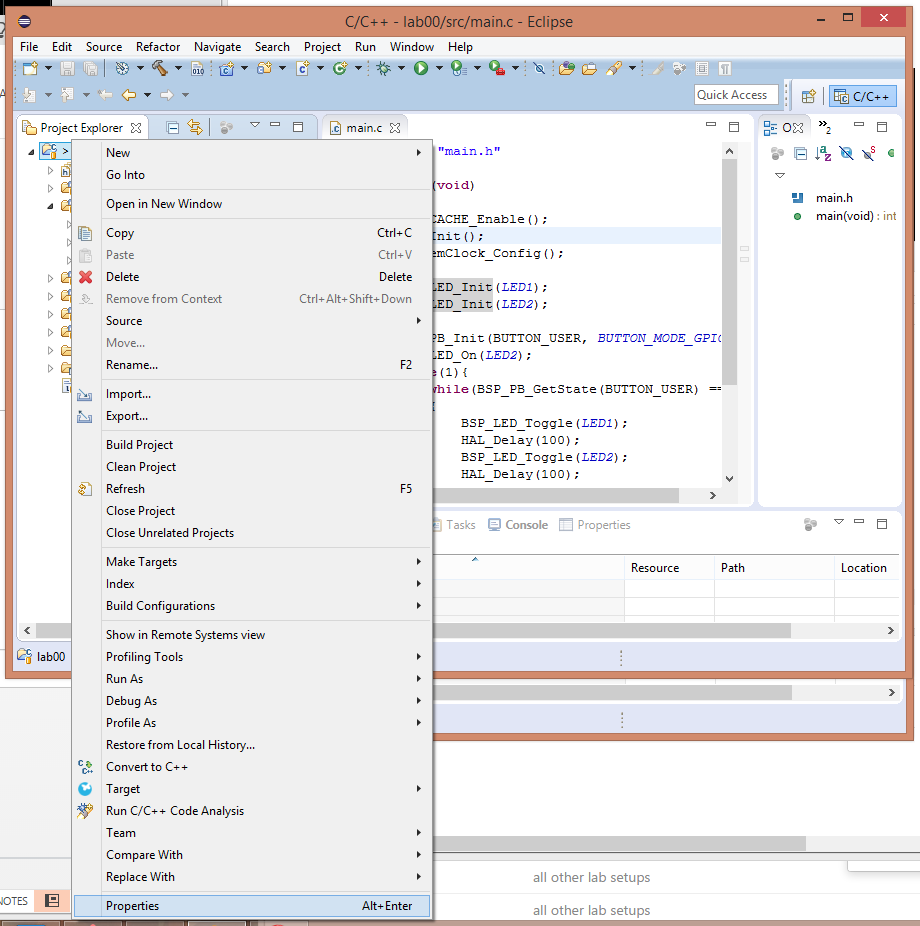
1. Click “Finish” and, if applicable, close the welcome screen (via a little “x” in its tab; you can optionally uncheck “Always show Welcome at start up” in the bottom right of the screen before closing it). STM32 Workbench should now look like Figure 7:



**Figure 8: STM32 Workbench in Ready to Use Form**

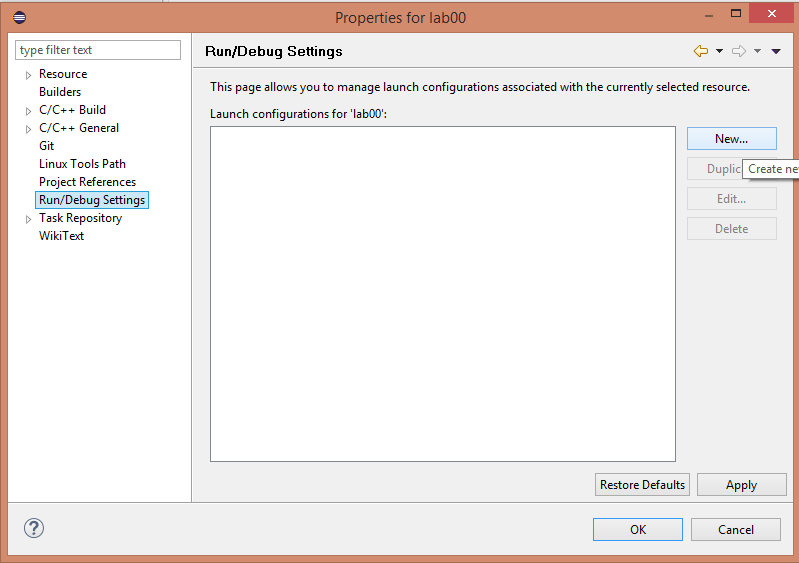
**Part 5: Building and Uploading Labs**

1. To build a lab, open your main.c (or any file in the project you want to build, really) and click the hammer icon at the top of the screen. Any errors will be displayed on the bottom. If you get any errors when trying to compile Lab01, you’ve missed a step (it’s a working code example). Otherwise, fix any errors and try again.
2. To upload your code after compiling, right-click on the root directory of the project and click “Properties.”



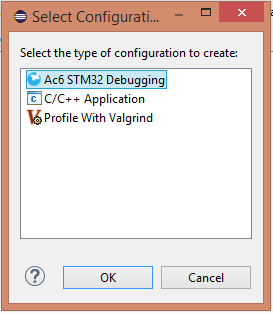
**Figure 9: Open Lab Properties**

1. From the lab properties dialog, click “Run/Debug Settings” and click “New…”



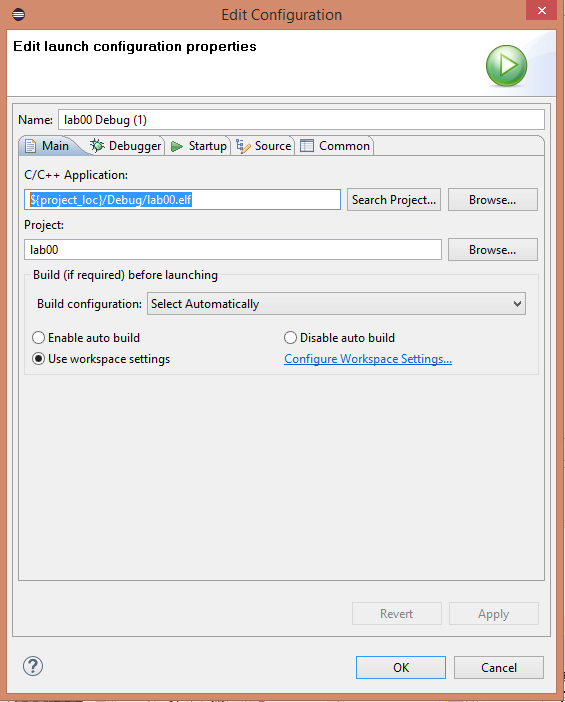
**Figure 10: Run/Debug Settings Dialog**

1. Select “Ac6 STM32 Debugging,” as per Figure 10:



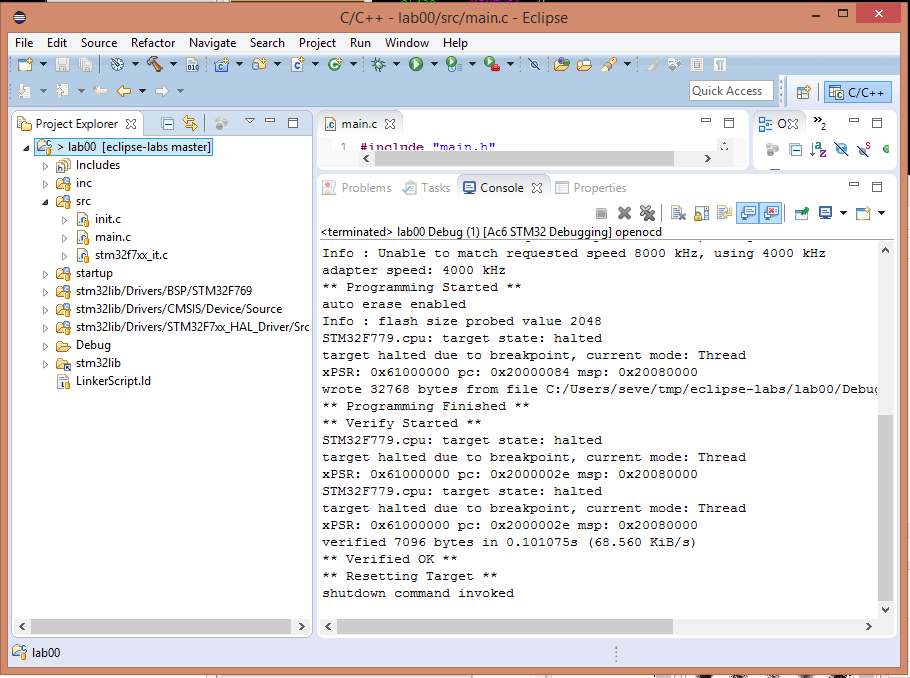
**Figure 11: Ac6 STM32 Debugging**

1. In the next dialog, either use “Search Project” or “Browse” to the “Debug” folder inside the lab folder to add the path to the compiled .elf file. Be aware that “Search Project” doesn’t always work. The dialog should look very similar to Figure 11 when done:



**Figure 12: Pointing STM32 to the Compiled Program**

1. Now click the “Debugger” tab and click “Show generator options” under the “Configuration Script” section. Change the Reset Mode to “Software System Reset.” There appears to be an issue that sometimes prevents uploading code to the DISCO board using hardware resets with OpenOCD, so consider this step necessary until that is fixed.
2. Finally, click “OK” and use the green arrow (it looks like a “Play” button) at the top of the IDE to upload and run your program. You will only need to set this Run Configuration setting once per lab, after which you can just use the Play button.
   1. If the “OK” button does not work (due to a bug in Eclipse), you can get to a non-buggy version of the same options by clicking the dropdown arrow next to the Play button and going to “Run Configurations.” From there, right click “Ac6 STM32 Debugging” and select “New,” and redo steps 5 and 6. You’ll need to use this method instead of steps 2-4 to get to these options until the bug gets fixed in an Eclipse update.
3. [Optional] Alternatively, click the green bug next to the Play button instead of using the Play button to upload your code and enter the step-by-step debugger (that steps through your C code!).



**Figure 13: Successfully Uploading Your Program Looks Like This**

Note: If you get an “\*\* Unable to reset target \*\*” error when uploading, make sure you selected the “Software System Reset” option in step 6.

If you did that and the error still occurs, you probably crossed a wire or tapped something conductive somewhere sensitive and one of the STM32 chips on the board freaked out (the one responsible for USB debugging seems to be less resilient than the main chip). Simply unplug and re-plug the Discovery board and then try again.

Any other error means you probably missed a step above.

1. If you’re coming from the build script system (you’ll know if you are), this should be the folder that contains Backend, Docs, etc. Unfortunately, you’ll have to remove spaces in your folder names if you want to use STM32 Workbench (for Windows, at least). [↑](#footnote-ref-1)
2. See footnote 1 above. [↑](#footnote-ref-2)