

## Keil for STM32F3 Discovery

### Connecting board to computer first time

#### 1. Download ST-Link utility

<http://www.st.com/web/en/catalog/tools/PF258168#>

#### 2. Install the program and run it

#### 3. Plug the board into the computer through ONLY the ST-LINK

#### 4. You should be able to connect to the board, if not, could be a problem with the ST-LINK

#### 5. Click on the tab “ST-LINK” and select “Firmware Update”

#### 6. If your board is connected, update the firmware

#### 7. If you get the “DFU Mode” error, unplug the board and hit “Install firmware” right after you plug the board in

Reason I updated the firmware is that I had an issue with Keil not recognizing the ST-Link until updating the firmware (This was with the new board I got yesterday), you can skip this if you want, but if Keil does not detect the ST-Link and the ST-Link Utility does, the firmware could be the issue.

## Intro to Keil

[https://www.youtube.com/watch?v=Z\\_BwDGf7FTs](https://www.youtube.com/watch?v=Z_BwDGf7FTs)

Firstly, this youtube video does cover quite a bit of setting up Keil, and can probably explain 80% of the process faster than I can type all this. Most of what I am writing is what you need to do differently in some manner.

1. Download Keil, “MDK-ARM v4”

<https://www.keil.com/download/product/>

2. Download firmware packet for STM32F3 Discovery

<http://www.st.com/web/en/catalog/tools/PF258154>

This not only contains example code, but all libraries of software for the ARM processor, already organized neatly, you will need these libraries later.

Take note of the example projects (demonstration is a good one), as I ask later for you to use one of them to help setup a new project.

3. Unzip the firmware packet “stm32f3discovery\_fw”
4. Click on the unzipped file and copy the folder “STM32F3-Discovery\_FW\_V1.1.0”
5. Paste that folder to wherever you are going to store your Keil projects. You can rename it if you want.
6. Click on the folder and click on the “Project” folder, and create a new folder for your new project

7. I like the way the given examples setup the project folder and sub-folders:

[project name]

-All code files goes here (ie main.c and a few others, see examples/video)

-Copy those files from one of the examples

-Note that those files may be read only, you may need to create a main.c from scratch to be able to edit the file.

[MDK-ARM]

-Keil project file is stored here

[List] (See Video)

[Obj] (See Video)

-Can set for debug to store a hex file here

8. Run Keil

9. If an example project is open, go to the “Project” tab and close the project

10. Create a new project (it is in the “project” tab), follow video

-You don’t need to copy the startup file, see next step

11. Setup the File Extensions

Do not copy the video exactly, but it is still useful

Instead, open up one of the example projects and open the extension tab, setup the extensions the same way as the example project is setup.

This will require a bit of searching the libraries for files, but once all the files linked to the project match the example, then you should be all set

12. Follow the video for setting up the Flash Tools

On the “C++” tab, do not copy the video exactly

Instead, refer to an example project for the “Define:” and for the “Include Paths”

The 1<sup>st</sup> include path is a bit tricky, if an error is caused building the project due to a file not found (happened to me), try adding/removing “..\” to that until it works

### 13. Go to the “Debug” Tab

You can select “Use simulator” when not using the board, this will still debug

If select the board debug, select the “ST-Link Debugger”

Hit “settings” and set the “Port” to either “SW” or “JTAG”.

I believe both ports should work similarly, although I used “SW”

More important is that if either Port is working (it will display “ST-LINK not found” if one or both is not working)

### 14. Go to the “Utilities” tab

Make sure target driver is the “ST-Link”

### 15. Build the project (button below the “open” icon), if using one of the examples, the code should build

If code does not build, double check your settings, especially the “include paths” and the “file extensions”

### 16. Enter the debug mode (option in debug menu)

If running with the board, changes on the board should occur as you step through them (The “Demonstration” project is likely good to see this process)

With all that, you should have a working project on the board

Now, I want to see what happens when creating a master workspace, since if it saves all project settings and applies them to newly created projects, that would make it much much easier for us to use multiple projects to micro-manage and test segments of code.

Only other workaround is to just swap the code in the “main.c” file with whatever you want to debug.

I hope you all don’t have any difficulties with this, or get any errors, if so, just bring your laptops tomorrow.