EC544: Timer Lab

Lets get a development environment up and running!

Note that some of the links and images may have changed from what you may observe on the website, but the overall flow is the same. For class discussion, join the corresponding Lab channel on Slack.

This assignment is in 3 parts that cover installing, running a demo project, and finally doing the lab assignment.

Part I

As we have been discussing in class, we are going to be using the MCUXpressoIDE, which is the Integrated Development Environment that lets you connect to the FRDM-K64K board.

This is a powerful, yet a completely free software. You will need to download it, configure it with the proper FRDM board setting, and then create a project for this assignment.

And so, in order:

1. Download and install MCUXpressoIDE from:

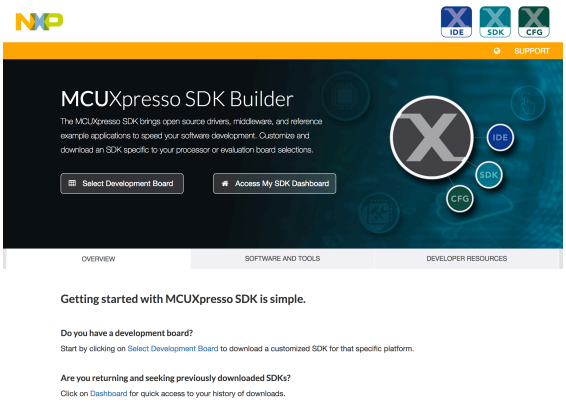
https://www.nxp.com/support/developer-resources/software-development-tools/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE?tab=Design_Tools_Tab

This software runs on Windows and Mac, and will require a registration on their website for download.

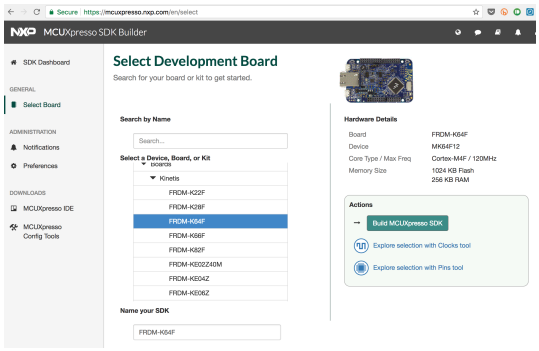
2. Configure the IDE with the SDK Builder:

2.1 Go to https://mcuxpresso.nxp.com/en/welcome

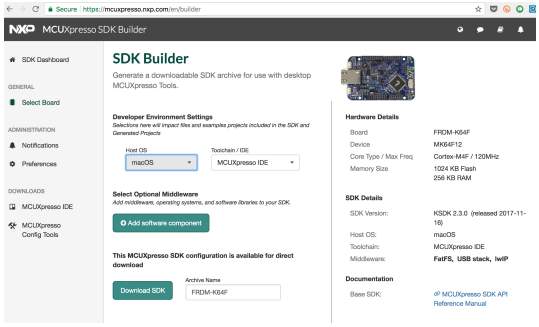
2.2 You will see the web page below:



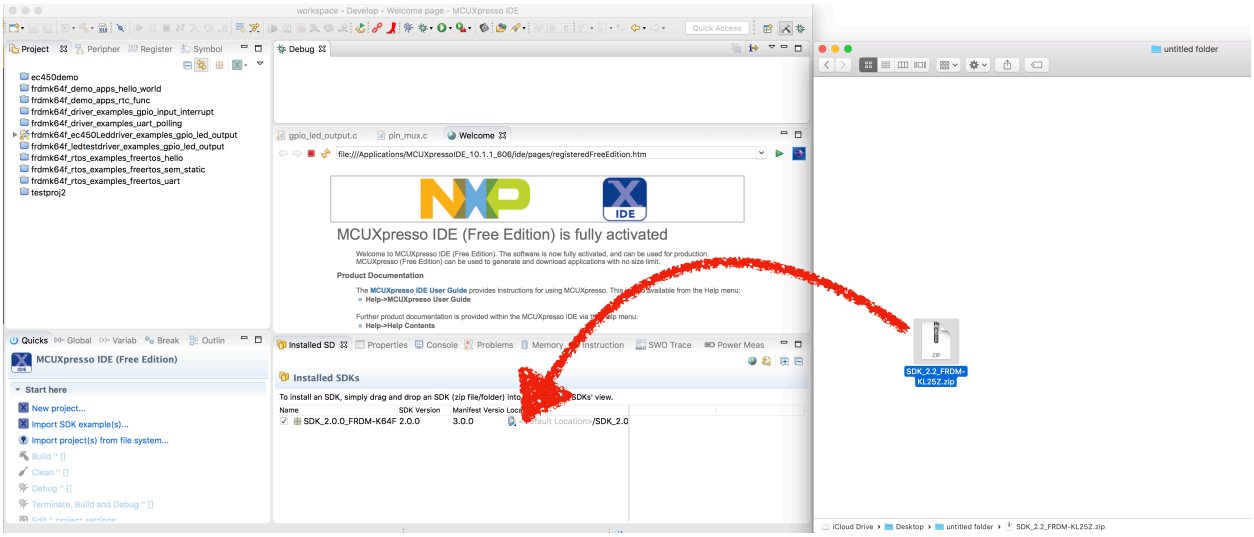
2.3 Click on “Select Development Board” and choose your FRDM board



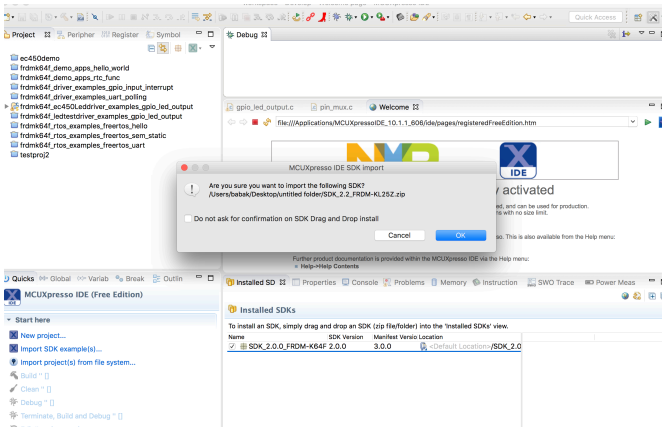
2.4 Click on “Build MCUXpresso SDK”, select the appropriate OS, and then “Download SDK”



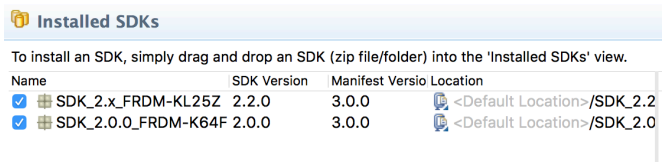
3. Start MCUXpresso, then drag and drop the downloaded SDK into the “Installed SDK” part of the IDE (see flash in image below):



Doing the step above for the FRDM-KL25Z SDK will result in the following image:



Hit ok, and you will now have a ready to use IDE for the second part of this assignment!



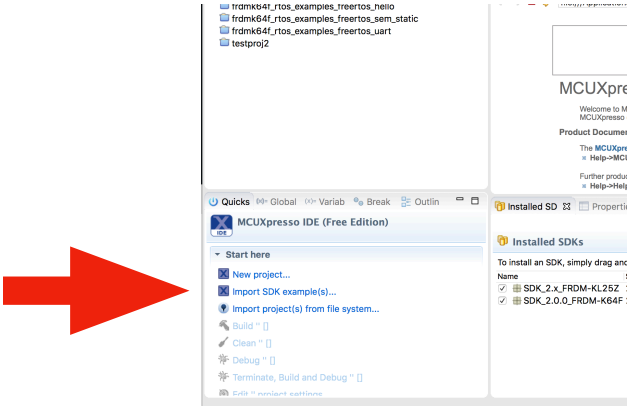
Part II

Let's test out a demo project, just to make sure that everything is properly installed.

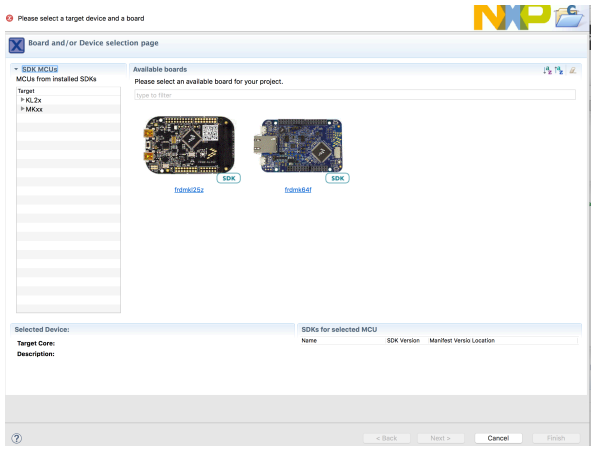
1. Connect the FRDM board to your computer, make sure that you are using the USB port that is labeled OpenSDA! (it will say so on the back of the PCB board, and also you can - and should - read the FRDM board's Users Manual).

   Also make sure that you are using a *data* USB cable (and not one that is designed to only provide power)

2. Click on Import SDK example(s)…

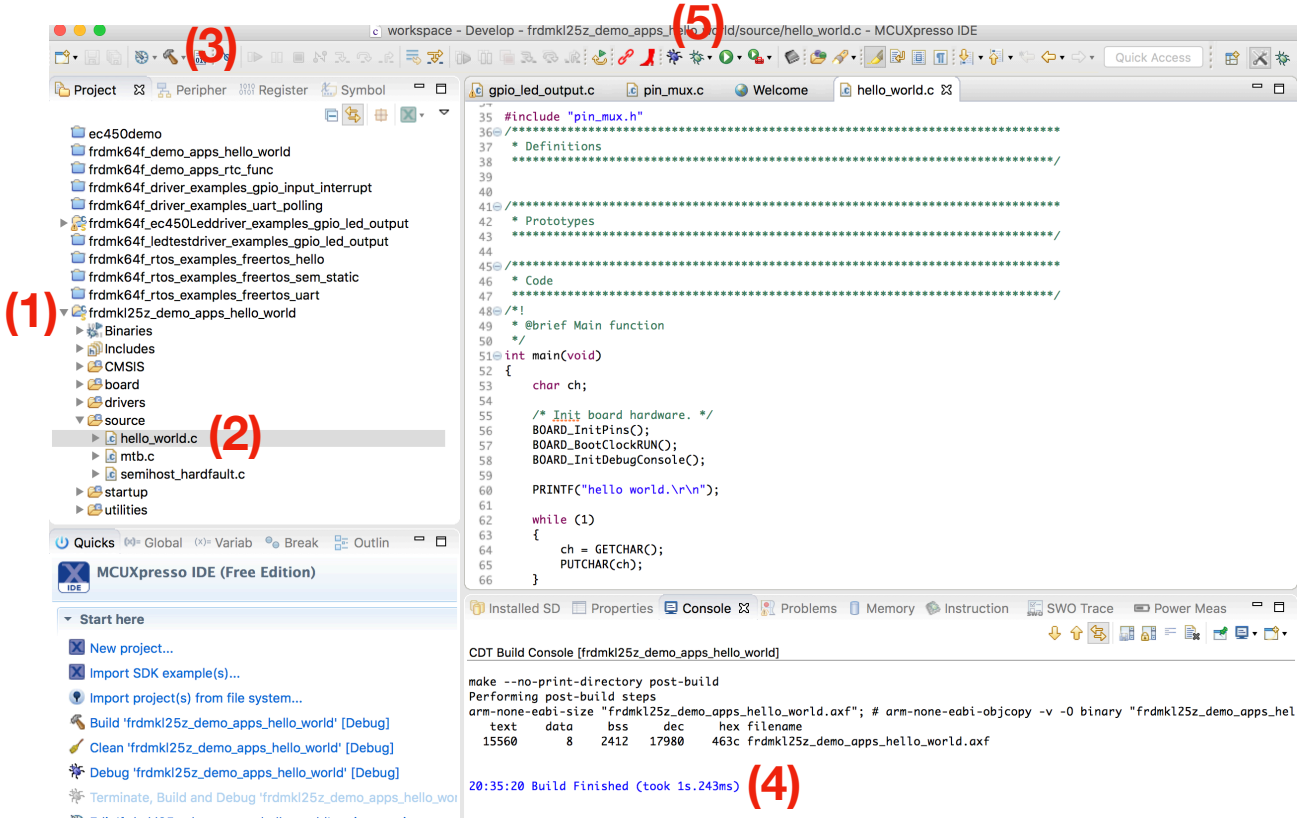


3. You should only see 1 board, but if you see 2, just select the appropriate FRDM board:



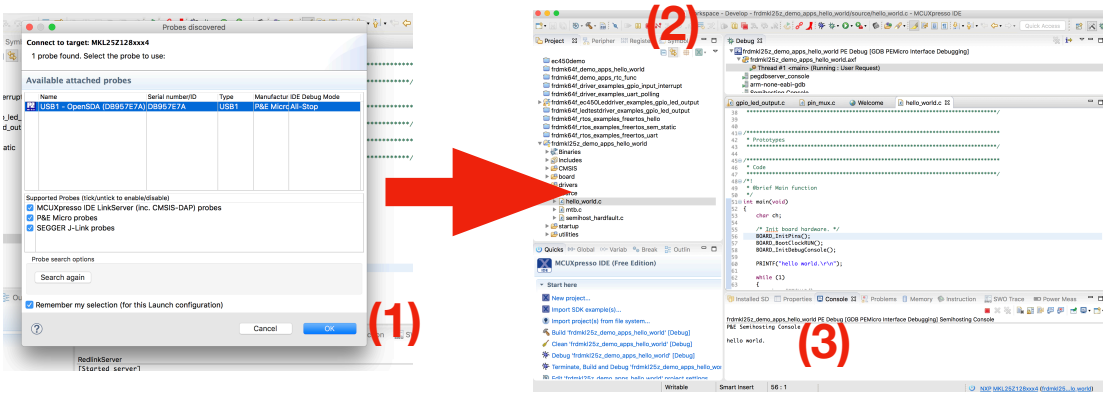
Select the hello_world demo and click on Finish

4. In the project that ensues (see 1 below), you can view the hello_world.c file (2), and click the build (3) button to make sure that every compiles satisfactorily (4). Once this happens, you are ready to program the FRDM board and debug with it! To do so, click on the blue debug button (5)



6. When you click on 5, you will be prompted with the image below, which indicates that the IDE has found and is connected to the FRDM board. Note that some boards will display “P&E”, and some boards will display “Segger”.

Once you hit OK (1) and run the program by clicking on the green play button (2) which is grayed out in this image, you should see the results (3)



Part III

Now we are ready for the lab assignment!

The assignment is to have a simple stopwatch timer, using *delays and printfs*. You need to print out the elapsed time in HH:MM:SS every 5 seconds.

The output should look something like this:

00:00:00
00:00:05
00:00:10
00:00:15

And so on and so forth. Once you have finished the assignment, please take a screen capture of the output and upload it to the assignment folder.

What to hand in: Your code, printed in PDF format, and the answer to the following questions:

Q1: What are your thoughts about using delays and printfs in this lab assignment?
Q2: What are your thoughts about using an ISR (interrupt service routine) to perform this function? (You don’t have to use an ISR, just asking what your thoughts are if you were to).