University of Colorado Boulder

Principles of Embedded Software

MCUXpresso IDE Installation Guide

Created by:

Shreya Chakraborty (SA for the course during Fall ’19)

Edited by:

Saket Penurkar

Rakesh Kumar

Date: 09/15/2020

INDEX

Introduction to IDE and KL25Z

What is an IDE?

An integrated development environment (IDE) is a software suite that consolidates all tools required to write and test software. As developers we use numerous tools throughout the software development process. An IDE includes all these tools such as text editors, compilers, debugger, code libraries such as vendor specific APIS, test platforms and so on. Without IDE we would have had to manage all these tools ourselves. Hence an IDE can improve productivity as we do not spend time deciding which tools to use. It also standardizes the development process. An IDE can help us in the following ways:

1. **Syntax Highlighting:** IDE knows the syntax of your language and prevents the syntactical errors by highlighting it, saving you some debugging time.
2. **Autocomplete:** It can anticipate what you type next, you do not need to write the entire name of the function, it automatically pops up. This saves keystrokes so the programmer can focus on logic in their code.
3. **Builds executables:** You do not need to make a Makefile. It automatically builds the code for you saves it in relevant directories in your project.
4. **Debugging:** You do not need external debuggers, logic analyzers or oscilloscopes, to see what is exactly happening with the code. The IDE provides you with a detailed debugging capability allowing you to use breakpoints, watchpoints, view variables, register values, disassembly (more on this later) and so on.
5. Version Control, Multiple language support, multiple board support, refactoring etc.

A little about MCUXpresso:

It is an Eclipse based IDE for NXP MCUs based on Arm Cortex-M cores, including LPC and Kinetis microcontrollers and so on. Provides advanced editing, compiling, and editing with MCU-specific debugging views, code trace, and profiling, includes pin, clock and peripheral tools, support for FreeRTOS etc. More info in this link

For a detailed guide of MCUXpresso IDE visit this link.

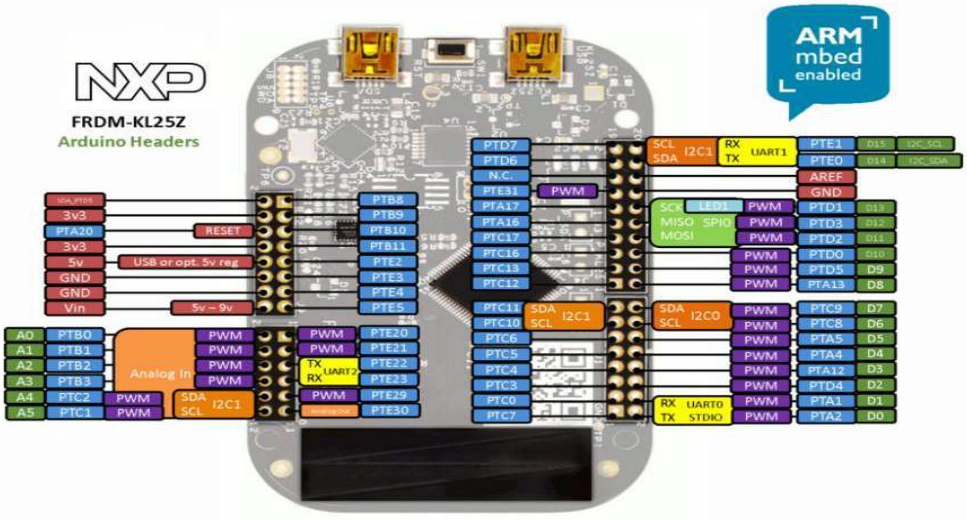
About FRDM KL25Z:

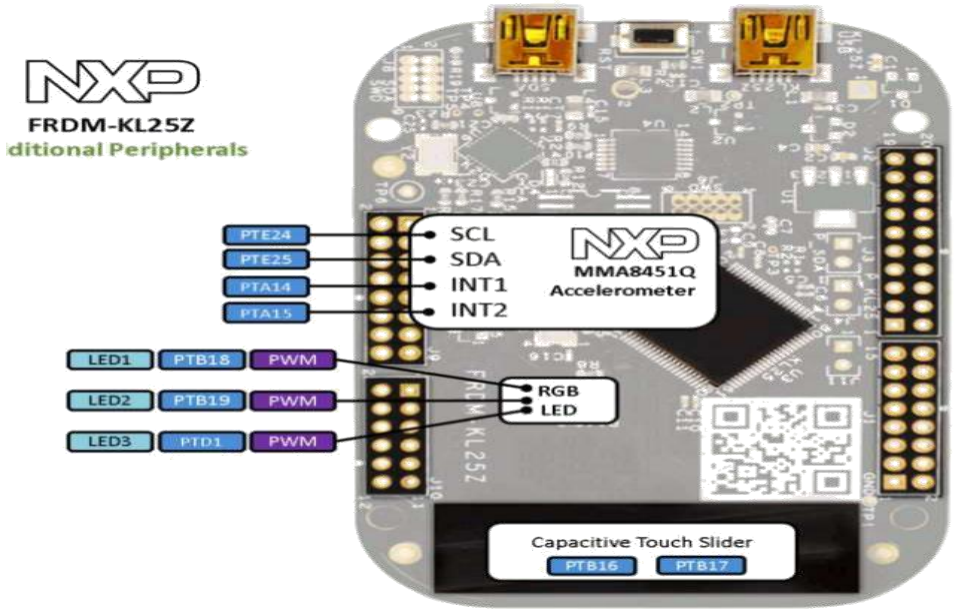
Specification:

Microcontroller	MKL25Z128VLK4 MCU ▪ 48 MHz ▪ 128 KB flash ▪ 16 KB SRAM ▪ 80LQFP
Sensor	▪ Capacitive touch "slider" ▪ MMA8451Q accelerometer
Debug	▪ Sophisticated OpenSDA debug interface ▪ Open-source data logging application provides an example for customer, partner and enthusiast development on the OpenSDA circuit ▪ P&E Multilink interface provides run-control debugging and compatibility with IDE tools
Connectivity	Easy access to MCU I/O
Tools & OS Support	▪ Arm® Mbed™ enabled ▪ Supported by Zephyr® OS ▪ Mass storage device flash programming interface (default) – no tool installation required to evaluate demo apps
User Components	Tri-color LED

and lots of interfaces including USB Host, USB Device, SPI, I2C, ADC, DAC, PWM, Touch Sensor and other I/O interfaces.

Pinout:





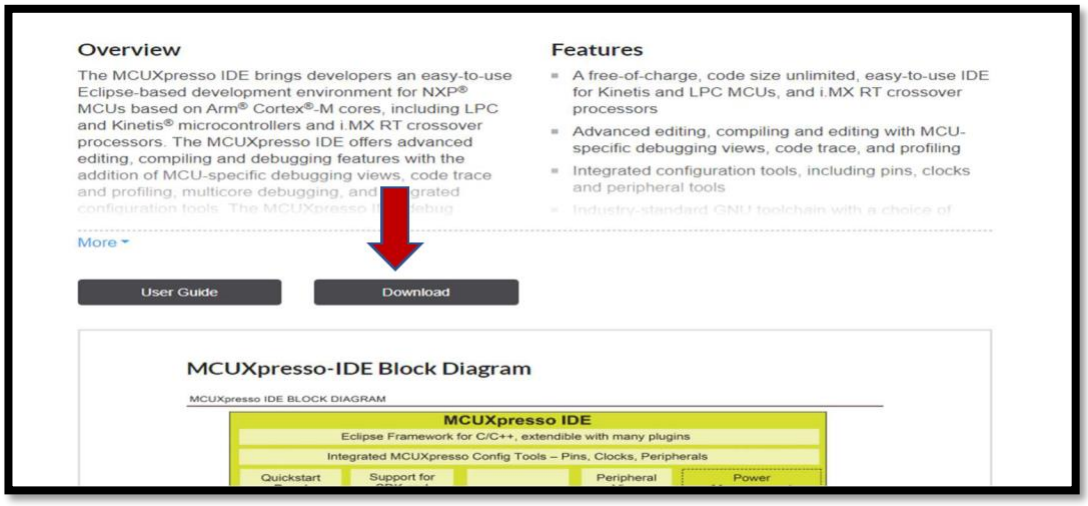
Notice the pinouts given. You can see that majority of the pins are multiplexed! They have 2 or more functionality. We need to choose which functionality to use.

Link to the User Manual.

Link for the reference manual/data sheets.

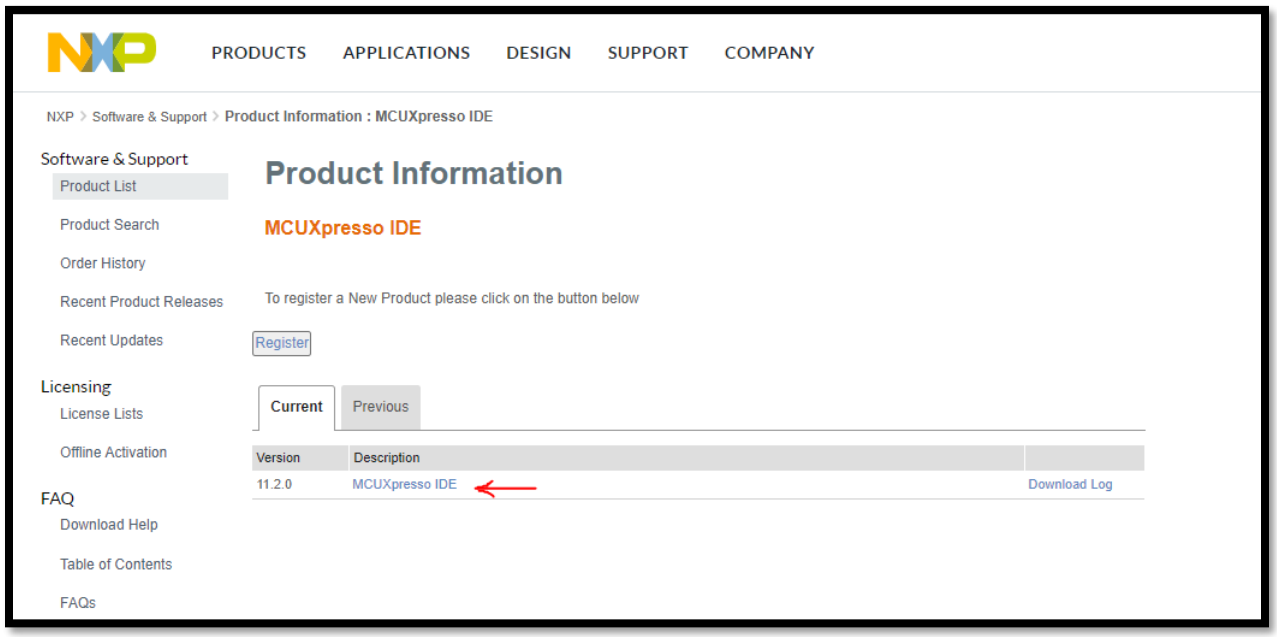
MCUXpresso installation instructions

Click this link to download the setup. User will be directed to the page as shown below:



This will prompt you to register if you are a first-time user, sign in if you already have an account.

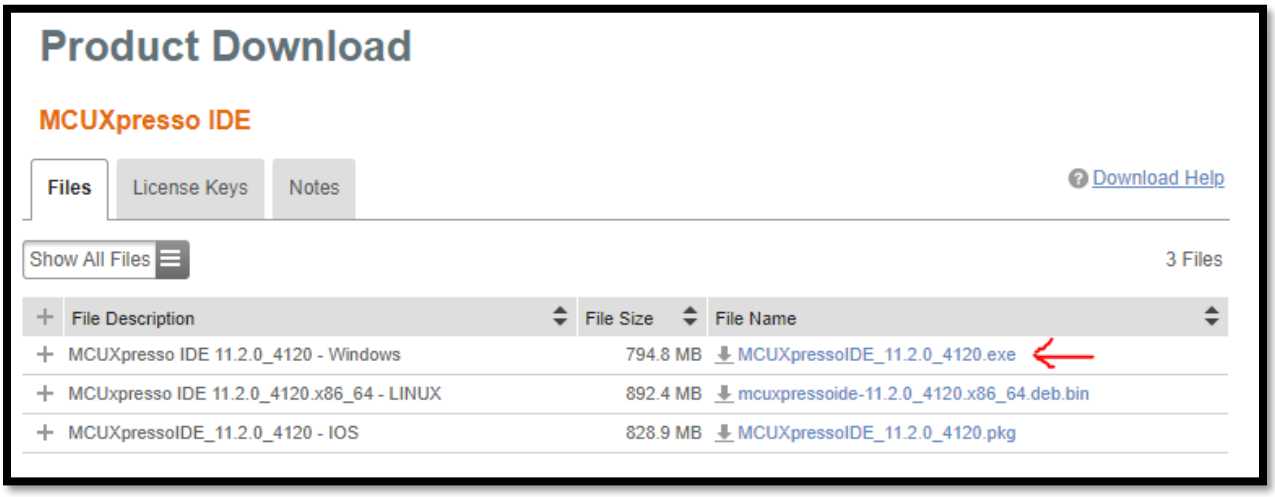
Once the user is successfully logged-in, click on the 'MCUXpresso IDE' as shown below:



Agree to the software terms and conditions, and then select the package based on your operating system.

For our purposes we will download for windows.

Click on the 'MCUXpressoIDE_11.2.0_4120.exe' (for windows) as shown in the figure below:



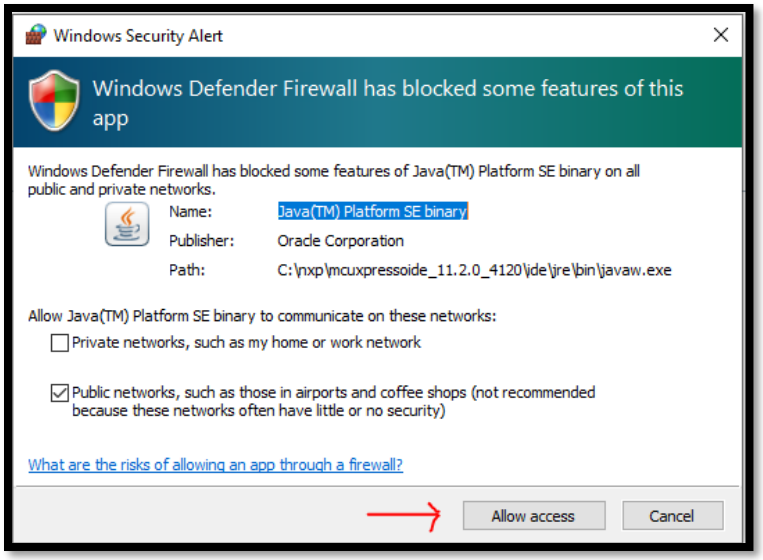
It will take a while to download the exe file. When you click on the exe, it will take you to IDE setup. Once the download is complete, run the executable as an administrator (to start the installation process). Agree to the terms and conditions and proceed with the steps as it directs! (User is advised not to change anything in between during the installation)

Once the installation is complete, user will observe a figure as shown below:

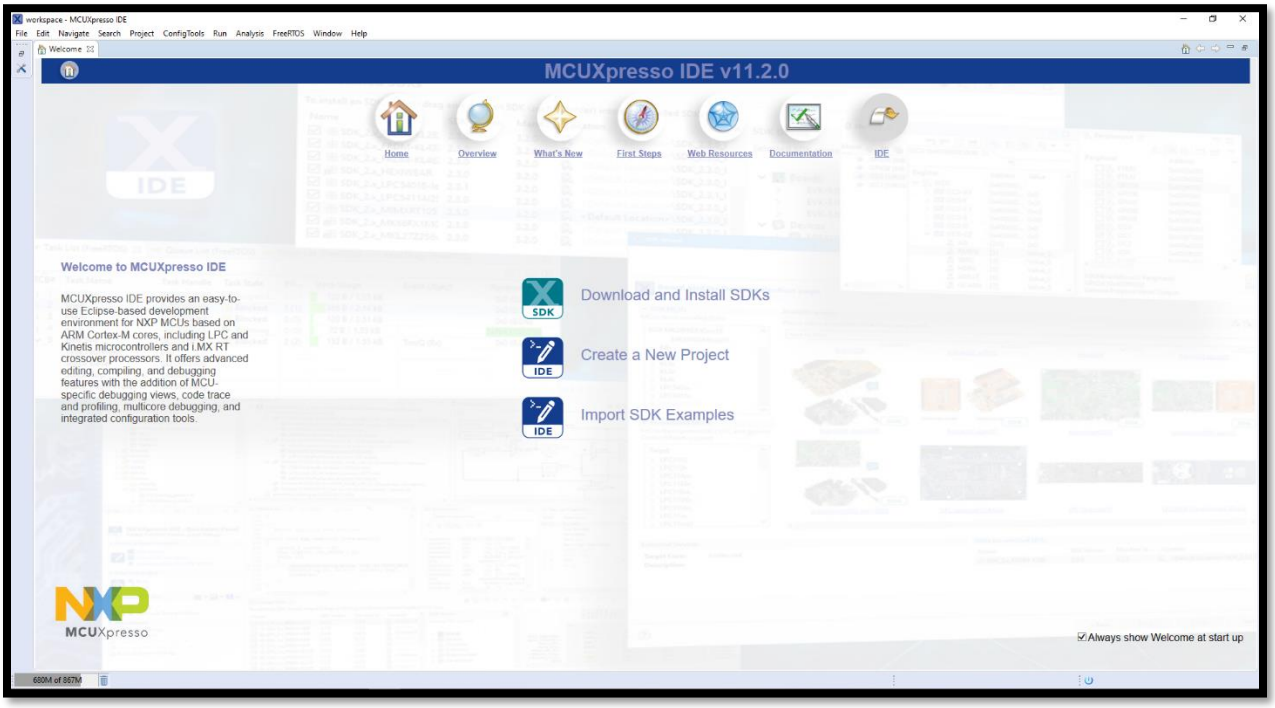
Click 'Finish' and keep the checkboxes marked if the user wants to launch the IDE, display the documentation, and read the user guide. User is recommended to keep the checkboxes marked!

On successful installation, you see the MCU user guide automatically open up. For any security alert that you may come across during the installation process, allow access.

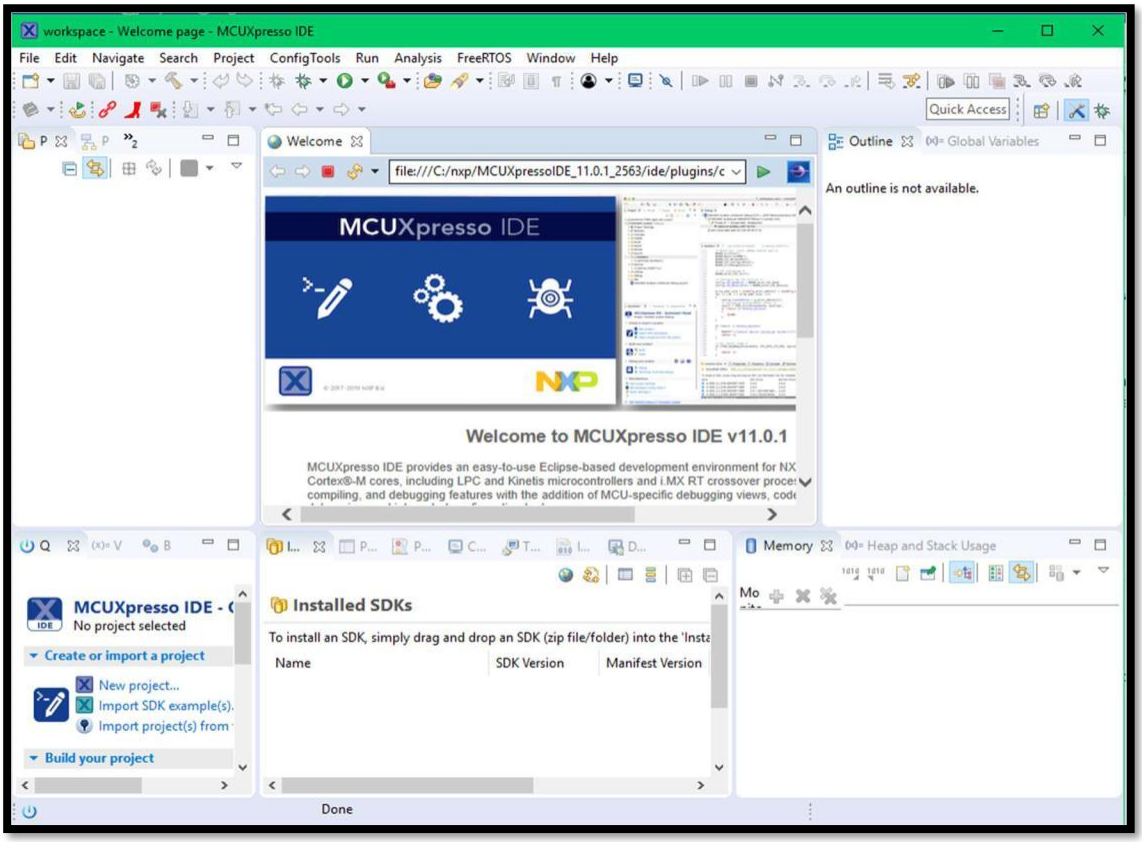
User may see the security alerts, as shown below! For any security alerts, user should allow the access.



Once, everything is ready, User will see a screen, as shown below: (this is a welcome screen, close this welcome window by clicking the ‘X’ on the top left-hand corner)

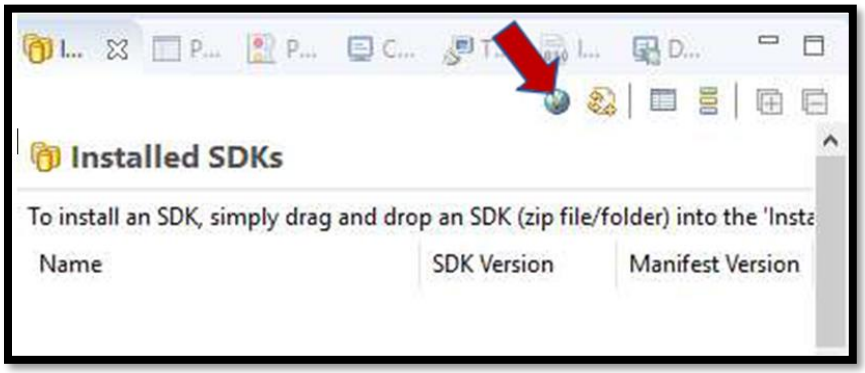


Once the welcome screen is closed, the user will get the IDE perspective! (as shown below)



At this point, our IDE setup is complete! Now, it is time to get the SDK!

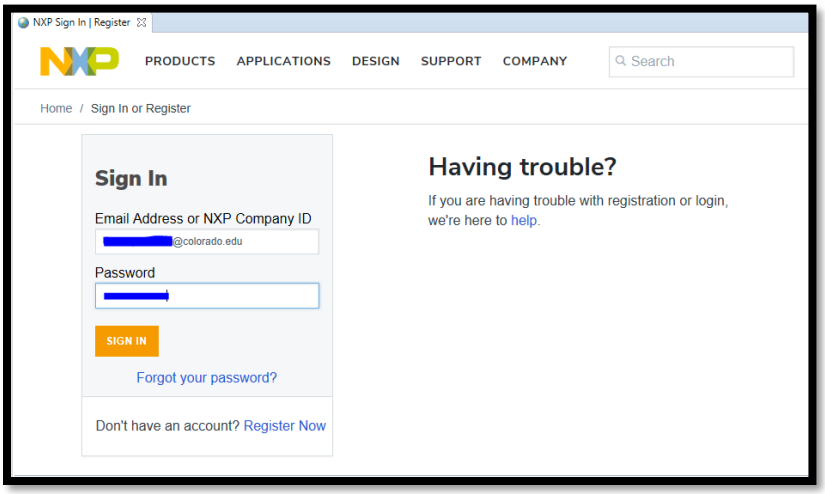
We can observe that there are no SDKs under installed SDKs in the above example. To download it, go to the tiny blue globe icon on the right of installed SDKs, as shown below:



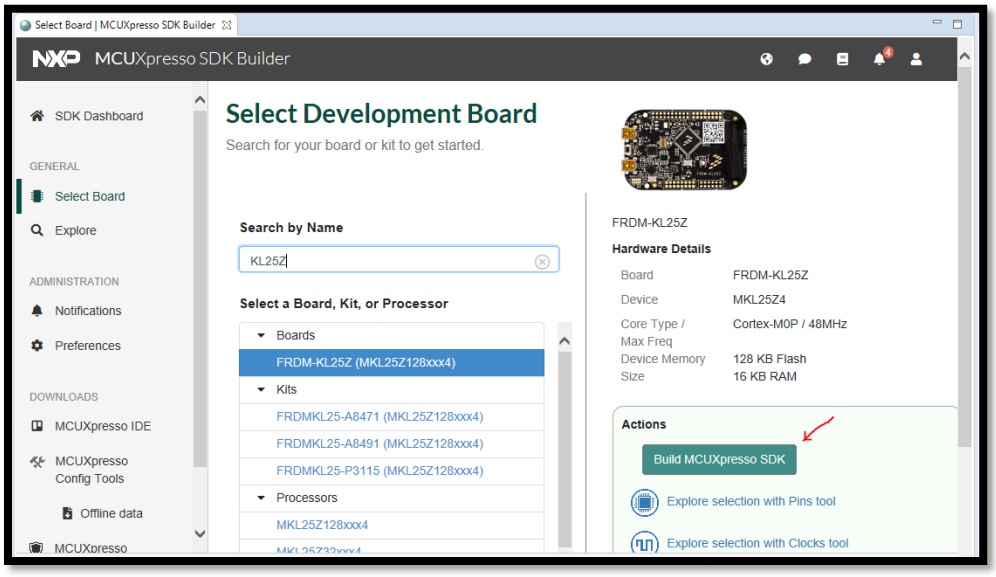
Once the globe icon is clicked, a window will show up, as shown below:

If this window does not pop up, click on this link.

Click on the 'Select Development Board' option and then the user will be asked to sign in (as shown below). Use the credentials you had used initially!

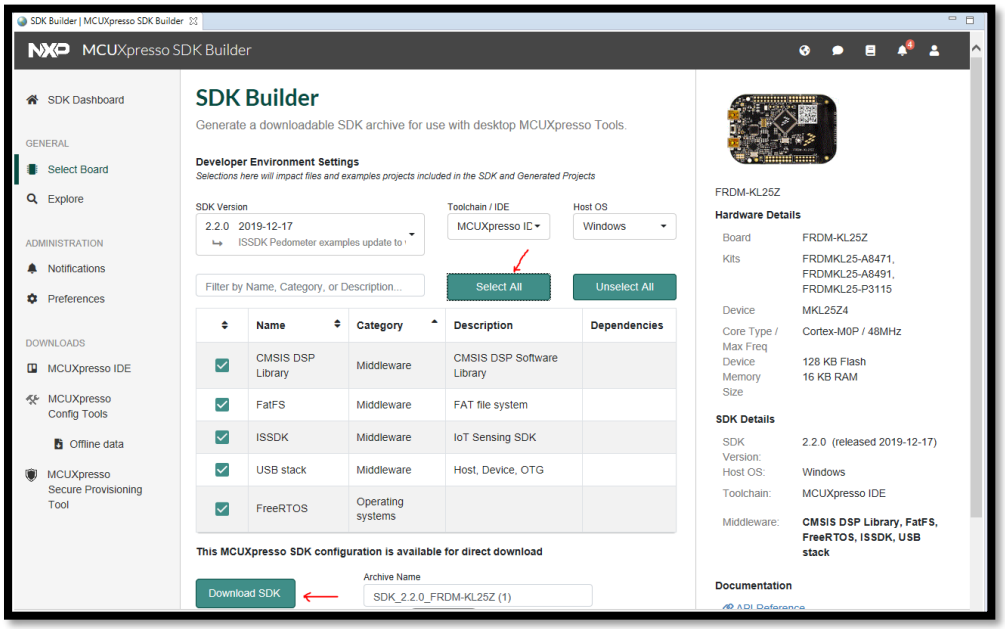


Once the sign-in is successful, the user will see a display as shown below:



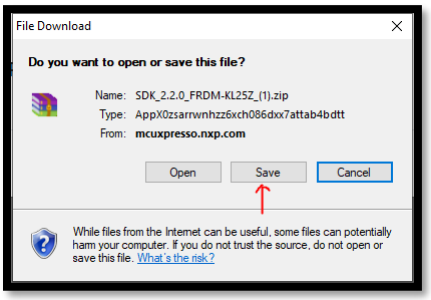
Type ‘KL25Z’ in the ‘Search by Name’ tab, select ‘FRDM-KL25Z (MKL25Z128xxx4)’ under the ‘Boards’ label and then click ‘Build MCUXpresso SDK’ (as shown in the figure above)

The user will now see an SDK builder window as shown below:



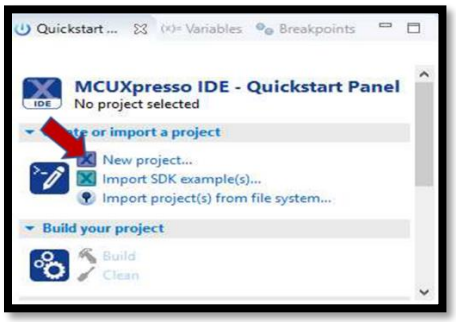
Click ‘Select All’ followed by ‘Download SDK’ (The CMSIS DSP Library, FATFS etc. middleware might not be useful now however if user wants to use this middleware in future projects, its recommended to ‘Select All’). Please make sure ‘FreeRTOS’ is checked!

After ‘Download SDK’ is clicked, the user may need to agree to some T&Cs. Save the zip file which needs to download, as shown below:

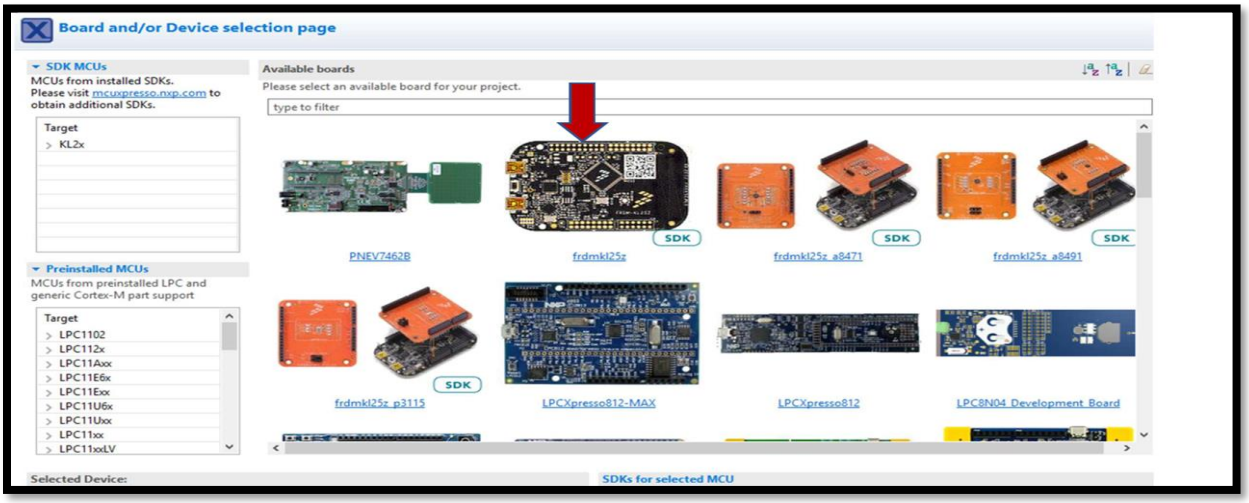


And a zip file will be downloaded! Simply drag and drop the zip file in ‘Installed SDKs’ box. If this does not help, a copy and paste should work always!

To confirm your SDK installation, click on new project option from the quick start panel, or just new icon from the menu bar and select New C/C++ project from under MCUXpresso IDE.



And see that the FRDM KL25Z SDK is added under the list of available boards.



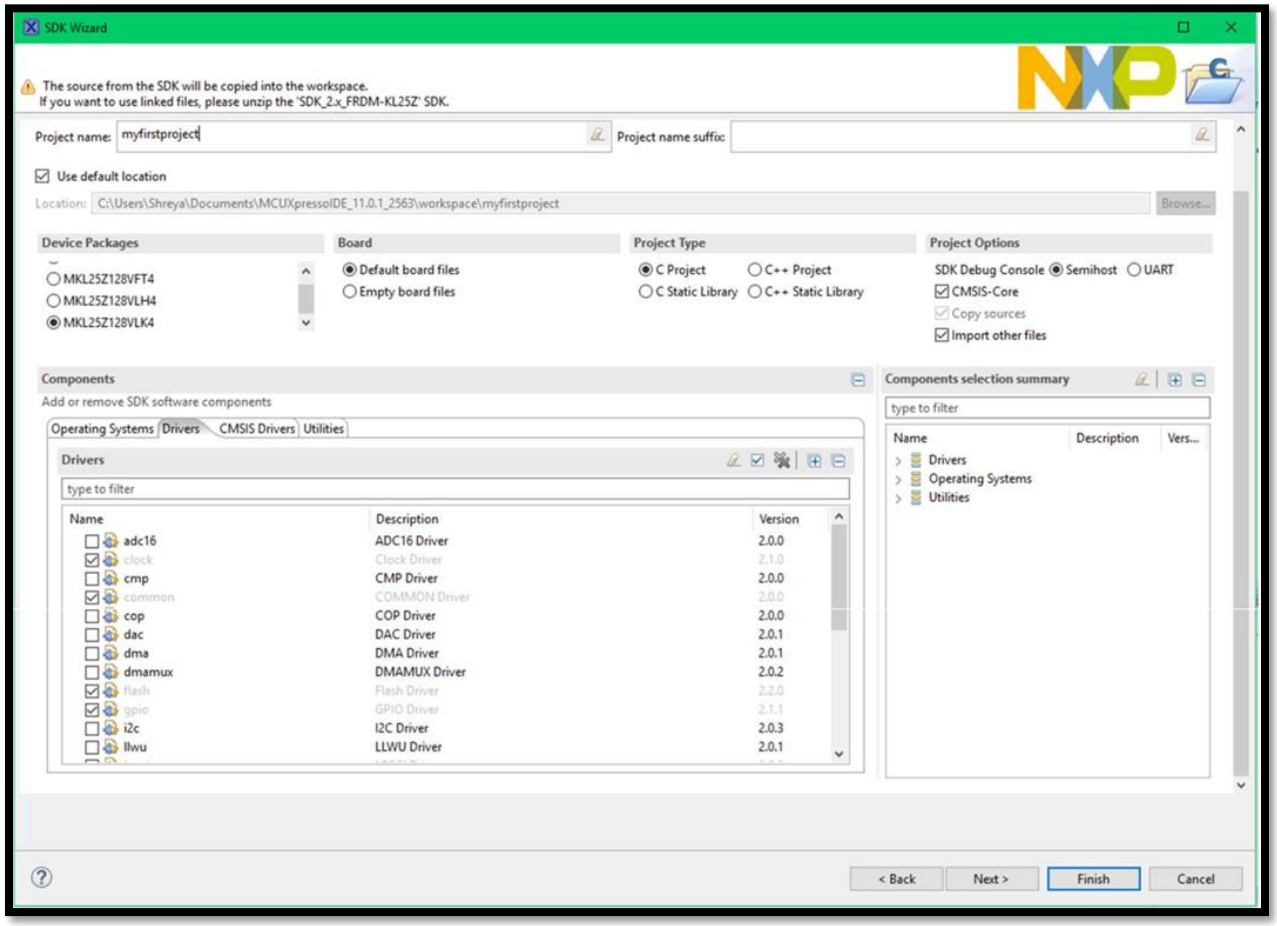
If you see the FRDMKL25Z board then the SDK configuration is successful!
Here the environment setup is done!

Getting started with your First Project

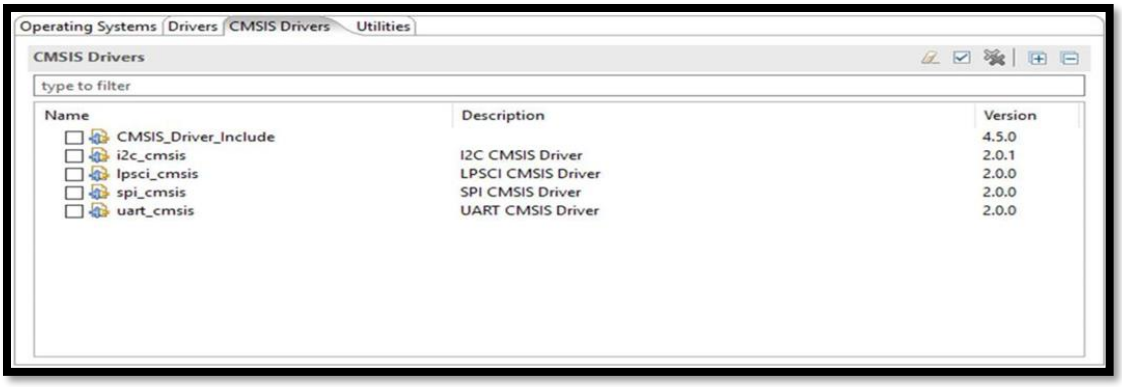
Download the FRDM-KL25Z Quick start package (if you don't have it already) from under 'Documents and Software' in the following this link

For document on 'Getting started with MCUXpresso' use this link.

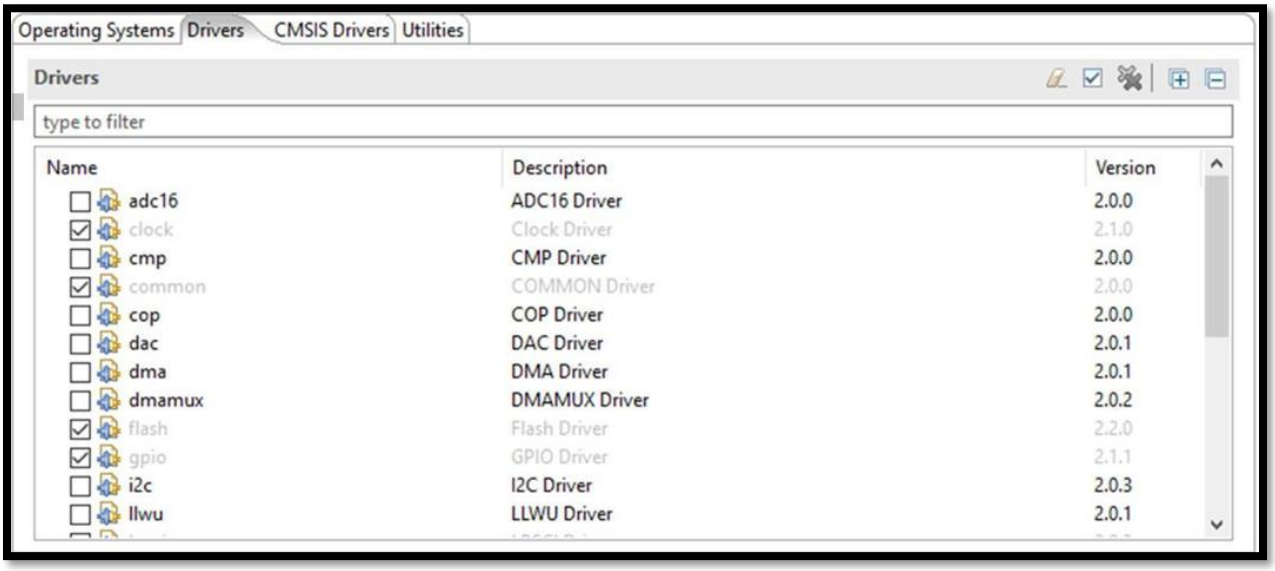
Go to new project button from the QuickStart menu or new from the menu bar. Click on the KL25Z SDK from the board and device selection page then click next.



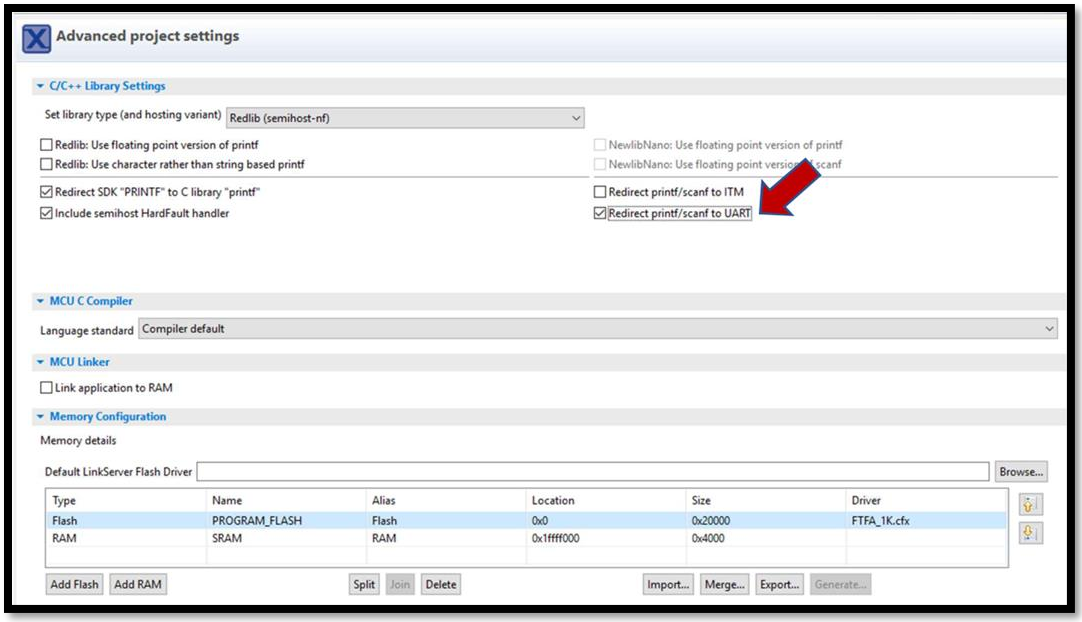
The SDK wizard as shown above allows us to choose operating systems, add remove drivers. The CMSIS (Cortex Microcontroller Software Interface Standard.) is an ARM standard. It is a portable software across all Cortex implementation and is usually written in C/C++.



For the purpose of this project we do not need to include any of the CMSIS drivers.

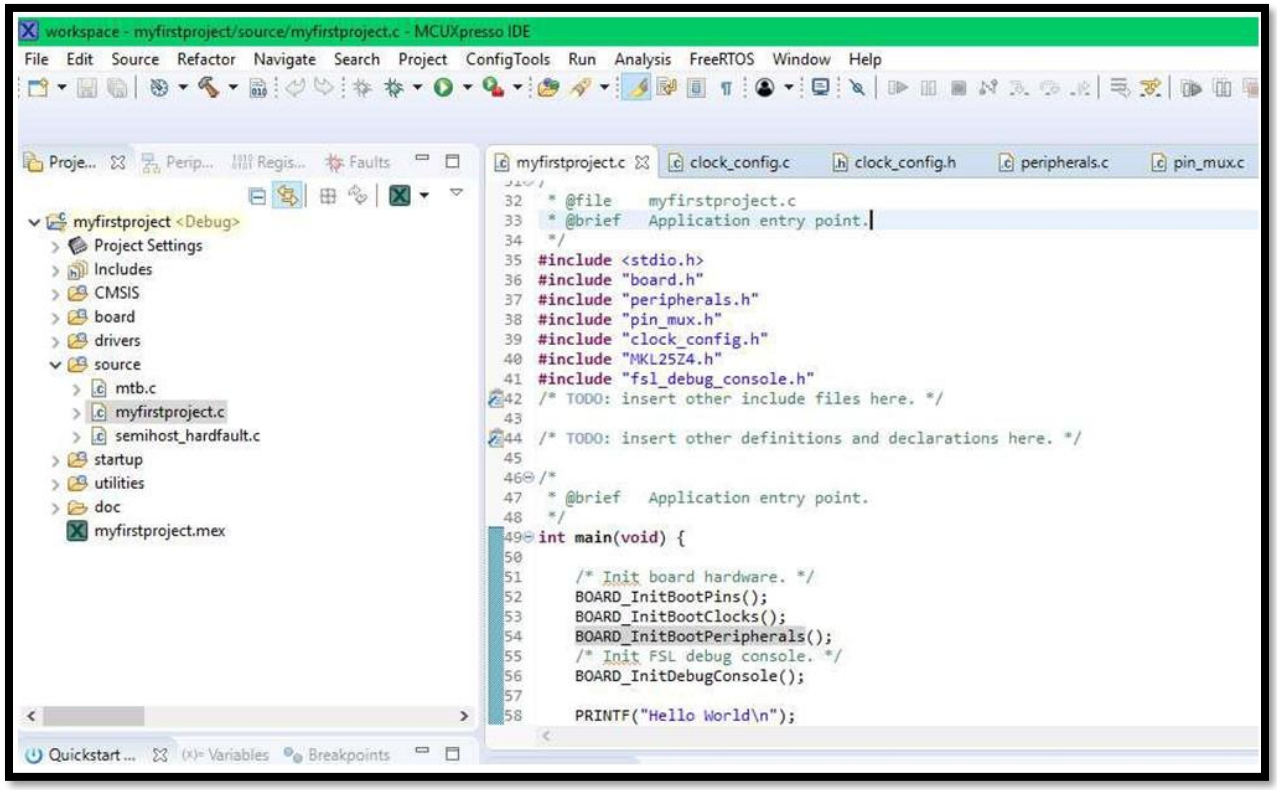


These are the peripheral drivers and are more vendor specific. Make sure you have gpio driver selected. No need to change the rest. Under operating systems, select ‘baremetal’ if it is not already selected. Bare metal programming refers to writing firmware which directly runs on the Target (hardware) without any underlying abstraction such as operating systems. This is what we are going to do in this class. Let the utilities be as it is. Finally give a project name, select the device package – MKL25Z128VLK4, select ‘c project’ and click next.



In the advanced project settings, select 'Redirect printf/scanf to UART' and click finish.

Our project gets generated and we see the following perspective view:



And…..You are all set!