

# University of Colorado Boulder

## PRINCIPLES OF EMBEDDED SYSTEMS

### Getting Started with KL25Z on MCUXpresso IDE (Install Only)

-By Shreya Chakraborty 9/10/19

#### INDEX

Topics	Page No
1. INTRODUCTION TO IDE AND KL25Z	2
2. MCUXpresso Installation instructions for KL25Z	5
3. Setting up your FRDM KL25Z	11
4. How to Unbrick your board?	16

#### 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's 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 ->(<https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE>)

For a detailed guide of MCUXpresso IDE visit the link -> [https://www.nxp.com/docs/en/user-guide/MCUXpresso\\_IDE\\_User\\_Guide.pdf](https://www.nxp.com/docs/en/user-guide/MCUXpresso_IDE_User_Guide.pdf)

## A little about FRDM KL25Z

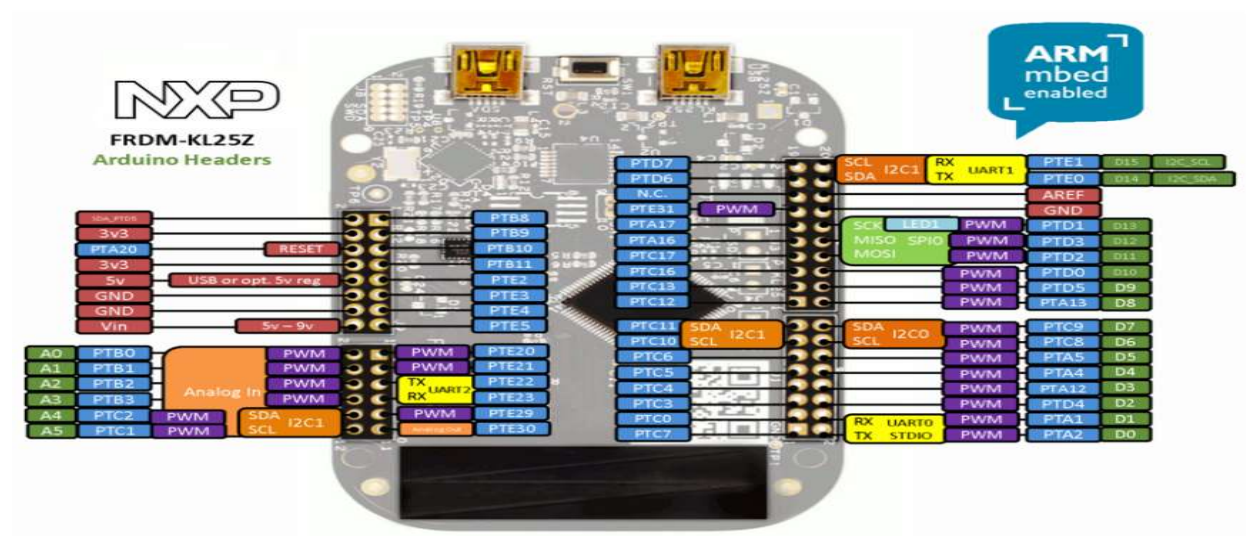
Originally developed by Freescale for their Kinetis L series, was acquired by NXP semiconductors in 2015. It's built on built on 32 bit ARM® Cortex™-M0+ processor running at 48MHz.

### Specification:

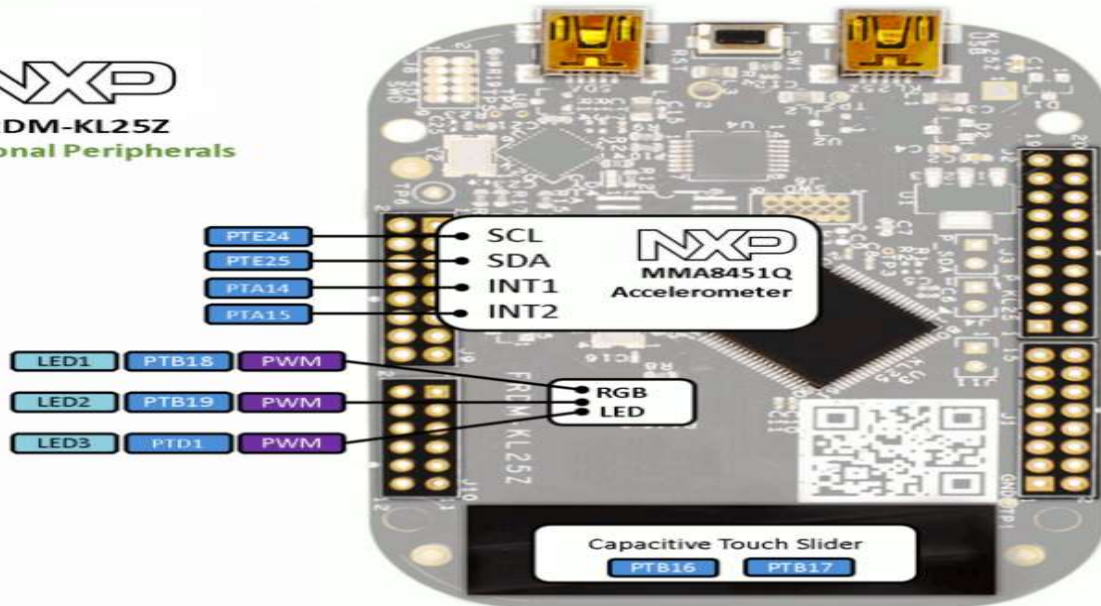
Microcontroller	MKL25Z128VLK4 MCU
	<ul style="list-style-type: none"> <li>■ 48 MHz</li> <li>■ 128 KB flash</li> <li>■ 16 KB SRAM</li> <li>■ 80LQFP</li> </ul>
Sensor	<ul style="list-style-type: none"> <li>■ Capacitive touch "slider"</li> <li>■ MMA8451Q accelerometer</li> </ul>
Debug	<ul style="list-style-type: none"> <li>■ Sophisticated OpenSDA debug interface</li> </ul>
	<ul style="list-style-type: none"> <li>■ Open-source data logging application provides an example for customer, partner and enthusiast development on the OpenSDA circuit</li> </ul>
	<ul style="list-style-type: none"> <li>■ P&amp;E Multilink interface provides run-control debugging and compatibility with IDE tools</li> </ul>
Connectivity	Easy access to MCU I/O
Tools & OS Support	<ul style="list-style-type: none"> <li>■ Arm® Mbed™ enabled</li> </ul>
	<ul style="list-style-type: none"> <li>■ Supported by Zephyr® OS</li> </ul>
	<ul style="list-style-type: none"> <li>■ Mass storage device flash programming interface (default) – no tool installation required to evaluate demo apps</li> </ul>
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 :



**NXP**  
**FRDM-KL25Z**  
 Additional Peripherals



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 -> <https://www.seeedstudio.com/document/pdf/FRMD-KL25Z.pdf>

Link for the reference manual/data sheets -> <https://www.nxp.com/docs/en/reference-manual/KL25P80M48SF0RM.pdf>

Let's get started!

# MCUXpresso Installation instructions for KL25Z

Go to this link and click download

<https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE>

**Overview**

The MCUXpresso IDE brings developers an easy-to-use Eclipse-based development environment for NXP® MCUs based on Arm® Cortex®-M cores, including LPC and Kinetis® microcontrollers and i.MX RT crossover processors. The MCUXpresso IDE offers advanced editing, compiling and debugging features with the addition of MCU-specific debugging views, code trace and profiling, multicore debugging, and integrated configuration tools. The MCUXpresso IDE includes a

[More](#)

**Features**

- A free-of-charge, code size unlimited, easy-to-use IDE for Kinetis and LPC MCUs, and i.MX RT crossover processors
- Advanced editing, compiling and editing with MCU-specific debugging views, code trace, and profiling
- Integrated configuration tools, including pins, clocks and peripheral tools
- Industry-standard GNU toolchain with a choice of

[User Guide](#) [Download](#)

**MCUXpresso-IDE Block Diagram**

MCUXpresso IDE BLOCK DIAGRAM

The diagram shows the MCUXpresso IDE architecture. At the top is the 'MCUXpresso IDE' layer, which is an 'Eclipse Framework for C/C++' that is 'extendable with many plugins'. Below this is the 'Integrated MCUXpresso Config Tools - Pins, Clocks, Peripherals' layer. At the bottom are four specific tool components: 'Quickstart', 'Support for', 'Peripheral', and 'Power'.

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

Next Click on MCUXpresso IDE under current .

[Products](#) [Applications](#) [Design](#) [Support](#) [About](#)

[NXP > Software & Support > Product Information : MCUXpresso IDE](#)

**Software & Support**

- [Product List](#)
- [Product Search](#)
- [Order History](#)
- [Recent Product Releases](#)
- [Recent Updates](#)

**Licensing**

- [License Lists](#)
- [Offline Activation](#)

**FAQ**

- [Download Help](#)
- [Table of Contents](#)
- [FAQs](#)

**Product Information**

**MCUXpresso IDE**

To register a New Product please click on the button below

[Register](#)

Current

Previous

Version	Description	Download Log
11.0.1	MCUXpresso IDE	<a href="#">Download Log</a>

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.

# Product Download

## MCUXpresso IDE

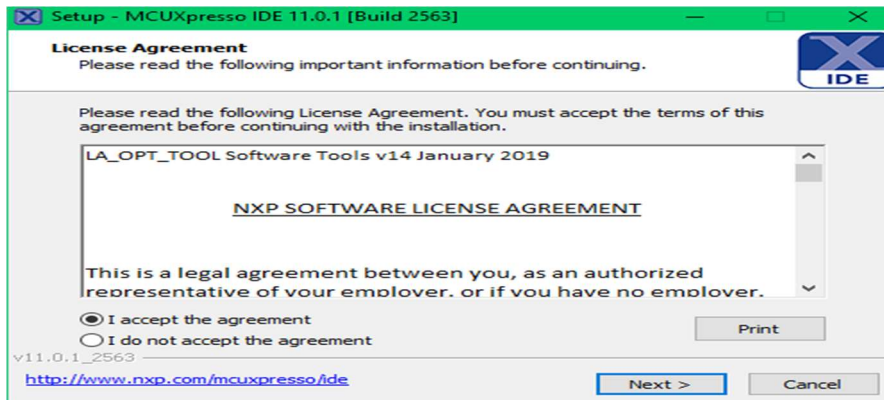
Files License Keys Notes

[Download Help](#)

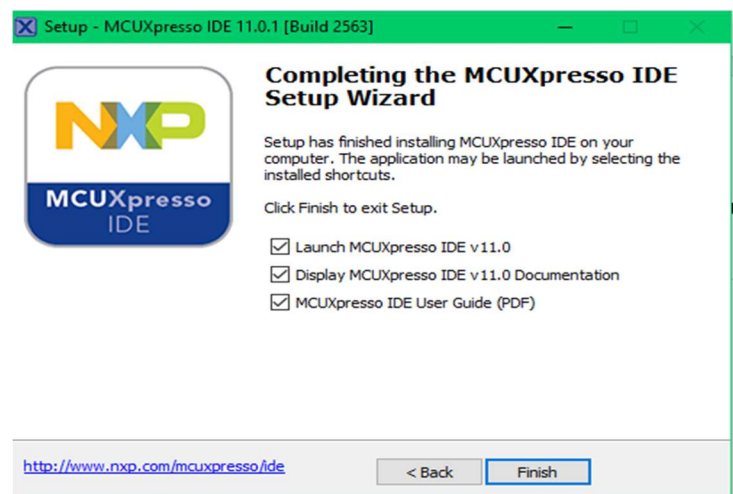
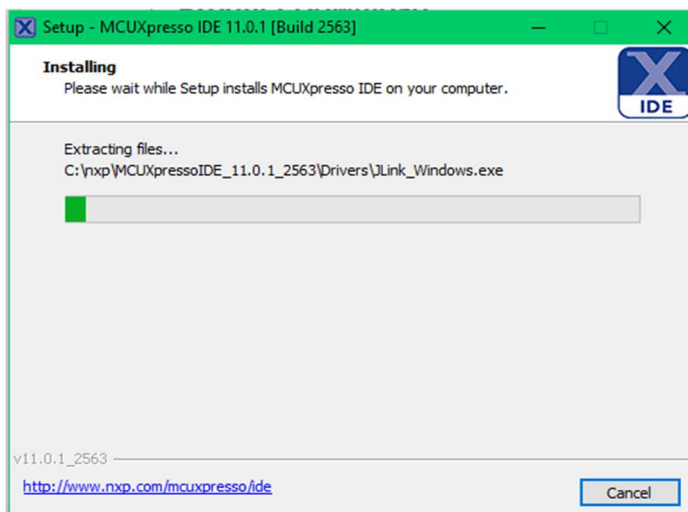
Show All Files 3 Files

+	File Description	File Size	File Name
+	MCUXpresso IDE v11.0.1 - Linux	818.3 MB	<a href="#">mcuxpressoide-11.0.1_2563.x86_64.deb.bin</a>
+	MCUXpresso IDE v11.0.1 - MAC	782.9 MB	<a href="#">MCUXpressoIDE_11.0.1_2563.pkg</a>
+	MCUXpresso IDE v11.0.1 - Windows	741.4 MB	<a href="#">MCUXpressoIDE_11.0.1_2563.exe</a>

It takes around 5 mins to download the exe file. When you click on the exe, it will take you to IDE setup.

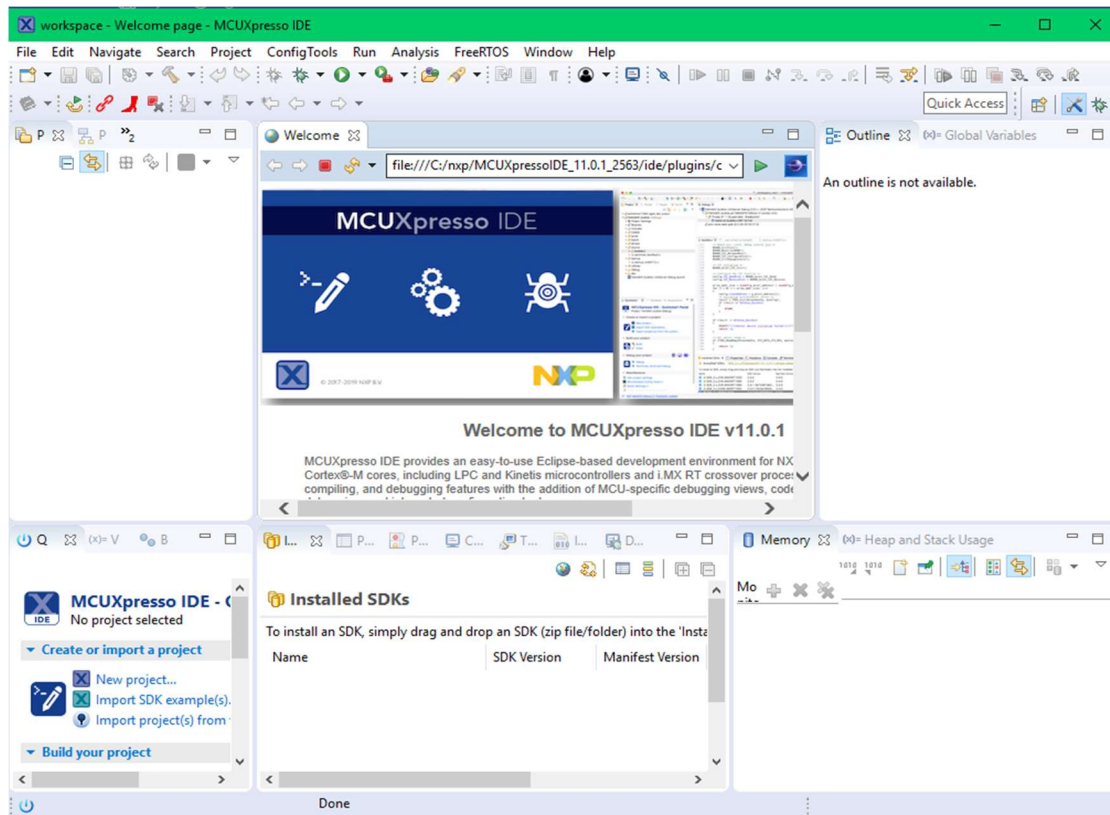


Keep on clicking next without changing anything and then click install and finally finish!!

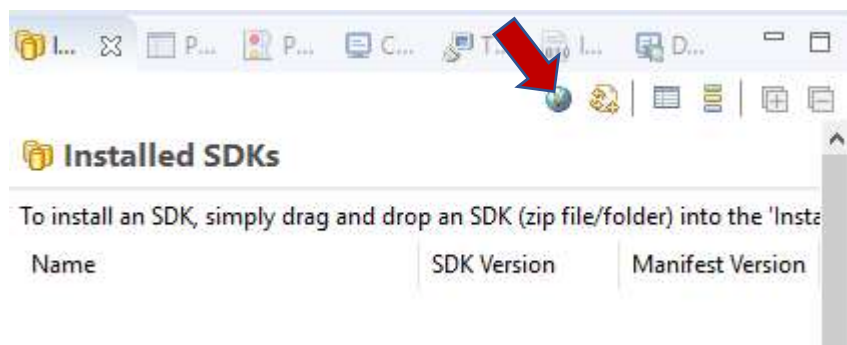




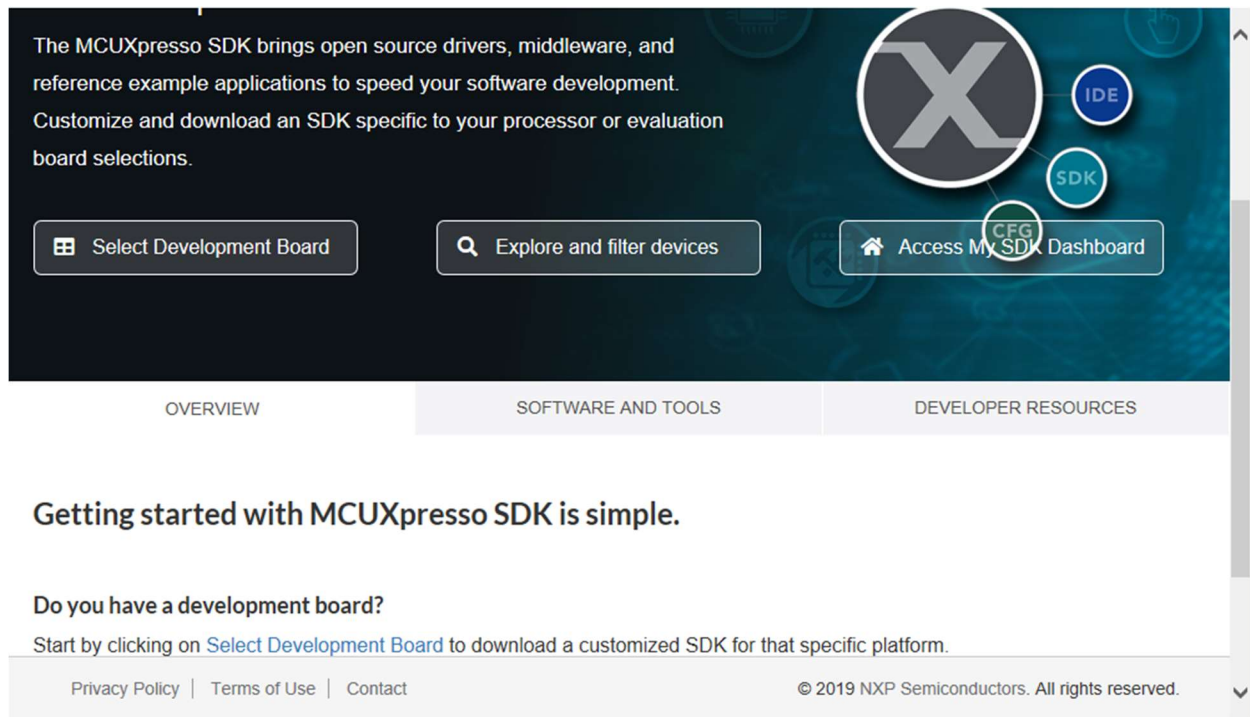
On successful installation, you see the MCU user guide automatically open up. For any security alert that you may come across during the installation process, allow access. And at this point we are done installing.



We have the IDE, now its 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 sdk's



It opens the following tab, alternately you can also go to this link : <https://mcuxpresso.nxp.com/en/welcome>



Click on 'select development board', after signing in with you account, you see the following display:

## Select Development Board

Search for your board or kit to get started.

### Search by Name

### Select a Device, Board, or Kit

- ▼ Boards
  - FRDM-KL25Z
- ▼ Kits
- ▼ Processors
  - MKL25Z128xxx4
  - MKL25Z32xxx4
  - MKL25Z64xxx4

### Name your SDK

Don't use: < > : ; , ' / \ ? \* % \ in the name of your SDK



### Hardware Details

Board	FRDM-KL25Z
Device	MKL25Z4
Core Type / Max Freq	Cortex-M0P / 48MHz
Device Memory Size	128 KB Flash 16 KB RAM

### Actions

**Build MCUXpresso SDK**



Explore selection with Pins tool



Explore selection with Clocks tool

Click on MCUXpresso SDK and it you the following page.



Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.



*Selections here will impact files and examples projects included in the SDK and Generated Projects*

Host OS	Toolchain / IDE
Windows	MCUXpresso IDE
SDK Version	
2.2.0 2017-06-29	

**Add middleware, operating systems, and software libraries to your SDK.**

**+** Add software component

[Download SDK](#)

Archive Name

SDK\_2.2.0\_FRDM-KL25Z (2)

Don't use: `<>.:~"/.!\?;*|` in the name of your SDK

Board	FRDM-KL25Z
Device	MKL25Z4
Core Type / Max Freq	Cortex-M0P / 48MHz
Device Memory Size	128 KB Flash 16 KB RAM

SDK Version:	2.2.0 (released 2017-06-29)
Host OS:	Windows
Toolchain:	MCUXpresso IDE
Middleware:	<b>CMSIS DSP Library, FatFS, IISDK, USB stack, FreeRTOS</b>

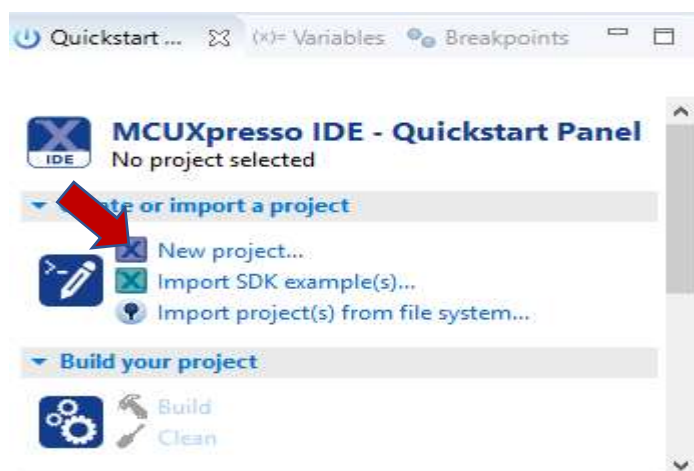
Base SDK:	<a href="#">MCUXpresso SDK API Reference Manual</a>
Middleware:	<a href="#">ISSDK API Reference Manual</a>
	<a href="#">SensorFusion API Reference Manual</a>



To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

Name	SDK Version	Manifest Version	Location
<input checked="" type="checkbox"/>  SDK_2.x_FRDM-KL25Z	2.2.0	3.0.0	 <Common>\SDK_2.2.0_FRDM-KL25Z.zip

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.

**Board and/or Device selection page**

**SDK MCUs**  
MCUs from installed SDKs.  
Please visit [mcuxpresso.nxp.com](http://mcuxpresso.nxp.com) to obtain additional SDKs.

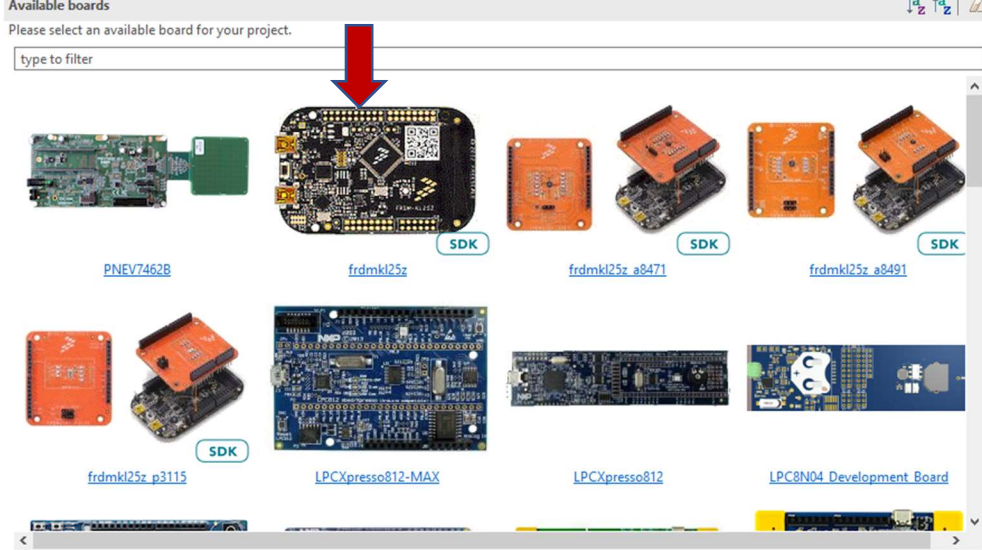
Target  
> KL2x

**Preinstalled MCUs**  
MCUs from preinstalled LPC and generic Cortex-M part support

Target  
> LPC1102  
> LPC112x  
> LPC11Axx  
> LPC11E6x  
> LPC11Exx  
> LPC11U6x  
> LPC11Uxx  
> LPC11xx  
> LPC11xxLV

**Available boards**  
Please select an available board for your project.

type to filter



**Selected Device:**

Target Core:  
Description:

**SDKs for selected MCU**

Name	SDK Version	Manifest Ve...	Location
------	-------------	----------------	----------

At this point you are done with your environment setup, so lets now move to starting a project.

## Setting up your FRDM KL25Z

1. Download the FRDM-KL25Z Quick start package (if you don't have it already) from under 'Documents and Software' in the following link-> <https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z?&tid=vanFRDM-KL25Z>
2. Plug in the USB cable normally with the open sda port and observe if you have a green light on. In case you don't, unplug and plug it in the bootloader mode while pressing reset button. If you see 8 fast blinks followed by a 2 second gap, your board must be bricked. Follow the unbricking procedure below.
3. Plug in a USB cable (not included) from a USB host to the OpenSDA mini-B USB connector while pressing the tiny 'RST' or reset button between the OpenSDA and KL25Z USB connectors. This puts the board in bootloader mode. Release the button. The FRDMKL25Z will be powered by this USB connection and you will see green led blinking at 1hz, indicating that it's in idle mode. Check the blink pattern shown below.

### 3 LED Status Indicator





The OpenSDA LED indicator is used by the MSD Bootloader and the standard OpenSDA Applications to provide mode and status information. A description of the LED indicator states and patterns is provided in Table 2.

**Table 2. LED Status and Mode Information.**

OpenSDA Mode	State Description	LED Pattern
Bootloader	Prior to USB enumeration	Off
Bootloader	Idle – running normally with no error conditions	Blinking: 500ms on, 500ms off
Bootloader	Error	2 seconds off followed by 8 rapid on/off blinks
Application	Prior to USB enumeration	Off
Application	Running normally with no error conditions and no USB activity	On
Application	USB activity (for example, MSD or CDC)	Blinking
Application	Error	2 seconds off followed by 8 rapid on/off blinks

In case you see 8 blinks with 2 sec gap, detach, plug it in normally and see if you can see a steady green light. If not, you may have to follow the unbricking procedure.

4. On connecting with the computer, A removable drive should now be visible in the host file system with a volume label of BOOTLOADER.

BOOTLOADER (F:)					Search BOOTLO.
^	□ Name	^	Date modified	Type	Size
	 FSL_WEB		8/8/2012 9:58 PM	HTM File	1 KB
	 LASTSTAT		8/8/2012 9:58 PM	Text Document	1 KB
	 SDA_INFO		8/8/2012 9:58 PM	HTM File	2 KB
	 TOOLS		8/8/2012 9:58 PM	HTM File	1 KB

OpenSDA is an open-standard serial and debug adapter. It bridges serial and debug communications between a USB host and an embedded target processor. OpenSDA software includes a flash-resident USB mass-storage device (MSD) bootloader and a collection of OpenSDA Applications. <https://www.nxp.com/docs/en/user-guide/OPENSDAUG.pdf>

5. Download the Windows USB drivers from <http://www.pemicro.com/opensda/index.cfm>  
In case SDA\_INFO.HTML doesn't automatically opens the pemicro website. Run the exe file. This will help the computer recognize the FRDM device.

**Windows USB Drivers**

Download [PEDrivers\\_install.exe](#) for manual install.

Version 12.4, updated January 23rd, 2017.

6. While in OpenSDA Bootloader mode, double-click SDA\_INFO.HTML in the BOOTLOADER drive. A web browser will open the OpenSDA homepage containing the name and version of the installed Application. This information can also be read as text directly from SDA\_INFO.HTML.

**Your Hardware Information**

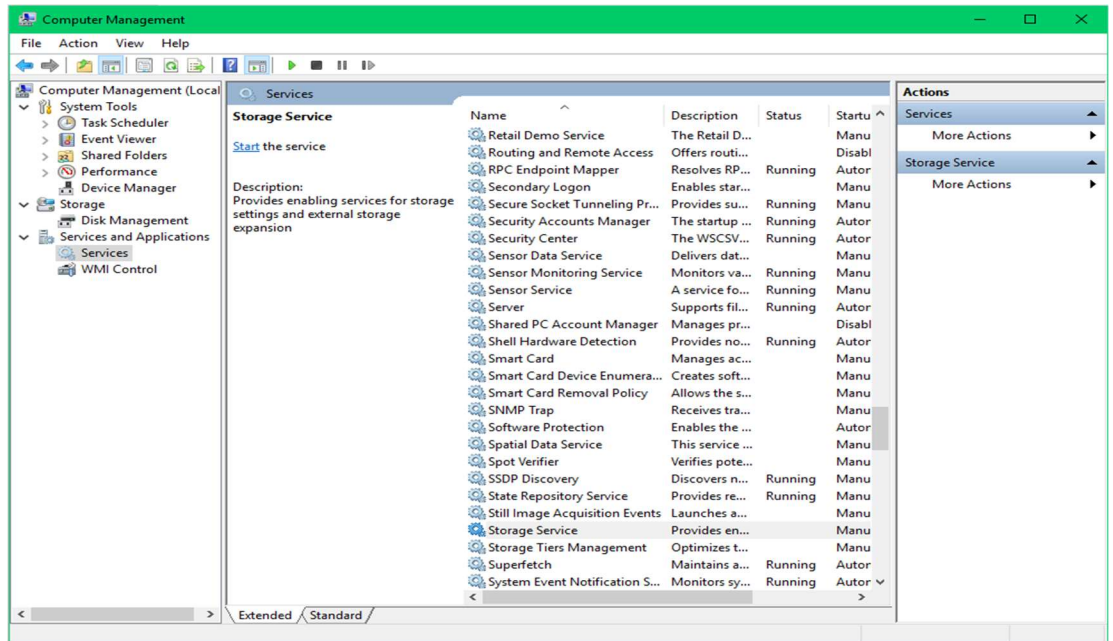
Board Name is: FRDM-KL46Z48M  
MicroBoot Kernel Version is: 1.05  
**Bootloader Version is: 1.09**  
Installed Application: PEMicro FRDM-KL46Z48M Mass Storage App  
Application Version is: 1.09  
DUID is: F9533938-F2CA81C0-377E1805-C45DE678  
EUID is: 01D1A239-7CB0874E-187F3A22-A66668D6  
TUID is: 74823938-473281B0-376CE80D-C5C8E678  
TOA is: 86B6E505-45E2DC47-FEB041DB-B2FD8F37  
TOA2 is: 86B6E505-2AAB964B-6F23F598-F4F69DCC  
SUID is: 86B6E505-8D48971F-37239804-8003EC65

► [Register your board](#)

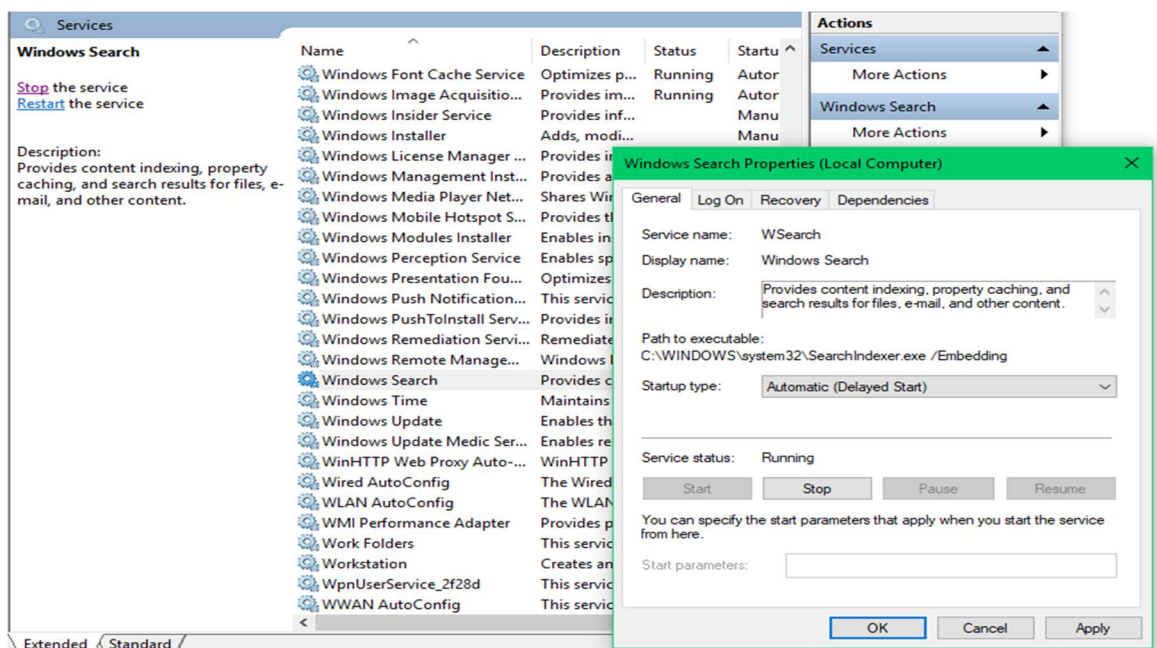
Notice the bootloader version. If your machine is windows 7 or less, you can skip it this step. The firmware was developed long time back and is not compatible with Windows 8 or above. Hence we need to update the bootloader of the board. From version 1.09 to 1.11.

- a. Unplug the board from host PC.
- b. stop/disable several Windows 10 services which interfere with the bootloader.
  - I. Go to Computer Management -> Services and Application > Services -> Scroll down to 'Storage Service' and click 'stop the service'.

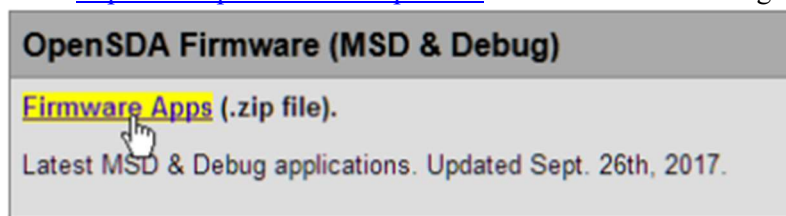




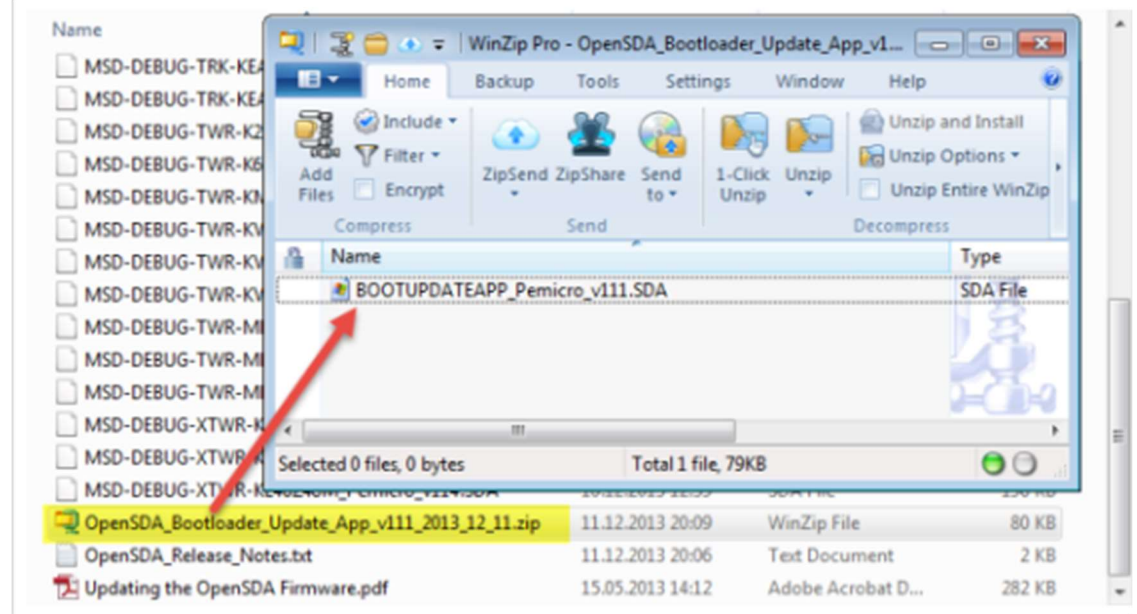
- II. Scroll down further to 'Windows Search', right click -> properties -> under startup choose disabled -> okay and then 'click stop the service'



- III. Go to <http://www.pemicro.com/opensda/> download the following

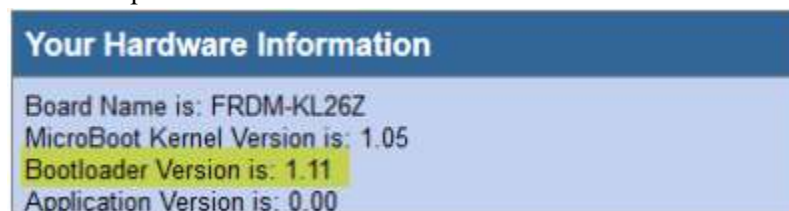


Extract all files. Extract the 'OpenSDA\_Bootloader\_Update\*.zip' inside that zip file as well.



You will need the BOOTUPDATEAPP\_Pemicro\_v111.SDA.

- IV. Press first the Reset button (and keep it pressed) while power the board with the Open SDA USB connector. And the board shows up as Bootloader drive as mentioned above.
- V. Copy the update file mentioned above to the bootloader drive and then unpower the board.
- VI. Power the board again without pressing the reset button, this should open up the bootloader driver. Click on SDA\_INFO.HTML file and you can see that the bootloader has been updated



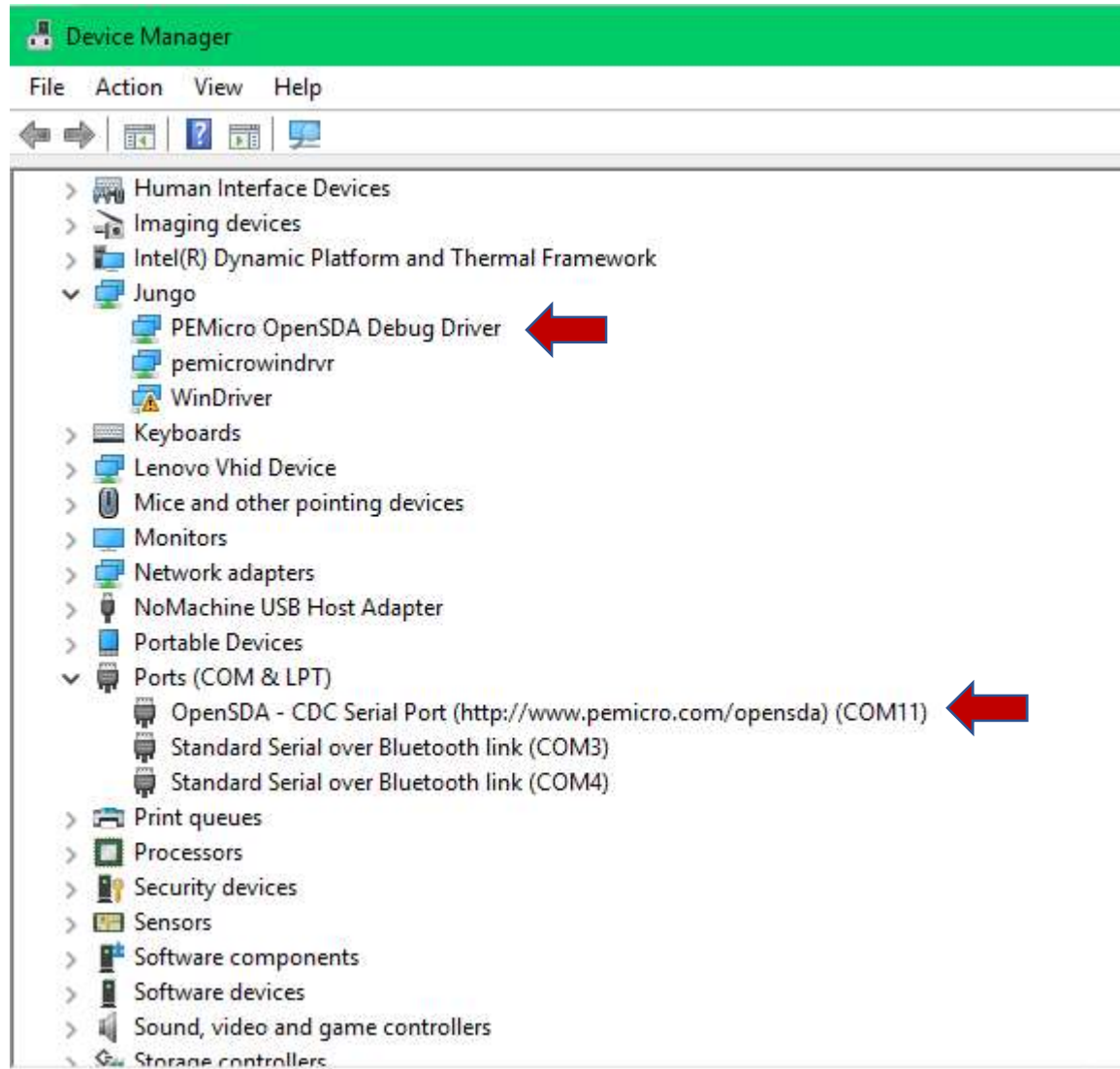
7. Now go to your earlier downloaded OpenSDA firmware, MSD & DEBUG (look above) Select the 'MSD-DEBUG-FRDM-KL25Z\_Pemicro\_v118.SDA' file and copy it to your bootloader drive. After that, un power your board.
8. Power the board again, in the normal mode (without pressing the reset button). The board will show up with it's name as the device. And the green LED will always be on. Lastly click on SDA\_INFO.HTML again and the final hardware information should look something like this:



## Your Hardware Information

Board Name is: FRDM-KL25Z  
MicroBoot Kernel Version is: 1.05  
Bootloader Version is: 1.11  
Installed Application: PEMicro FRDM-KL25Z Mass  
Storage/Debug App  
Application Version is: 1.18  
GUID is: 04032038-B4D8-4D8-873E-0000560005670

Go to device manager and see if you have them:

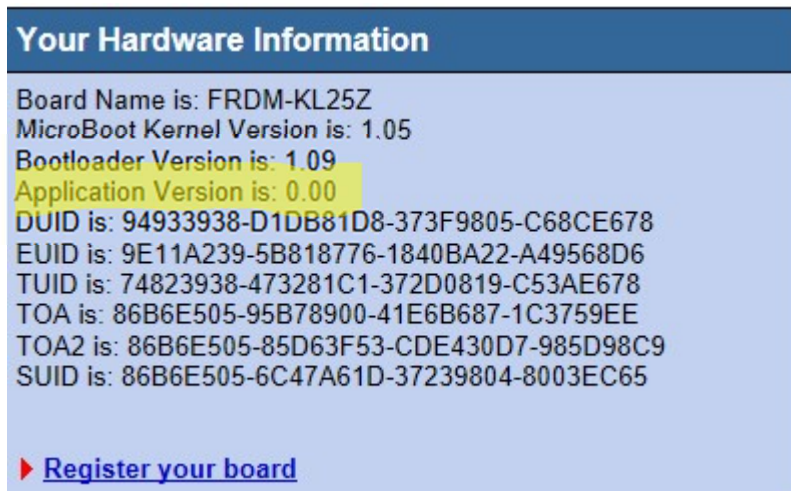


At this point you are all set and ready to start with your project!!

# How to Unbrick your board?

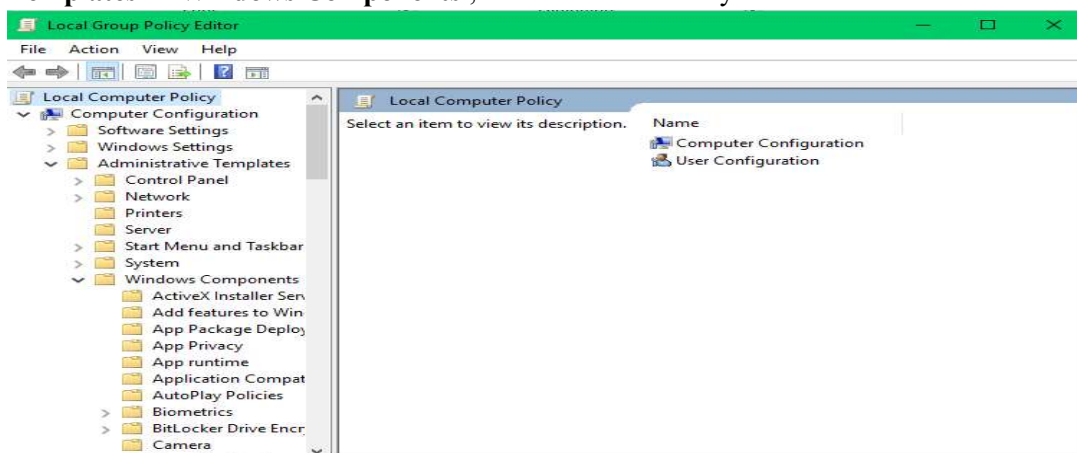
FRDM KL25Z will work without a problem on windows7, but for windows 8 and above, it doesn't work. Its firmware is outdated. If on normal plug in your green LED is off and in booth loader more, you LED pattern shows the error code, that means the OpenSDA debug application has been erased and the board does not boot into debug mode anymore. It's extremely easy to brick your board if you are using windows 10. Bootloader gets stuck by the information send by the windows 10 machine and erases the debug application.

1. Plug it in, in the bootloader mode holding the reset button. You will be able to see the boot loader drive. Click on 'SDA\_INFO' HTML file. And you should see something like this:

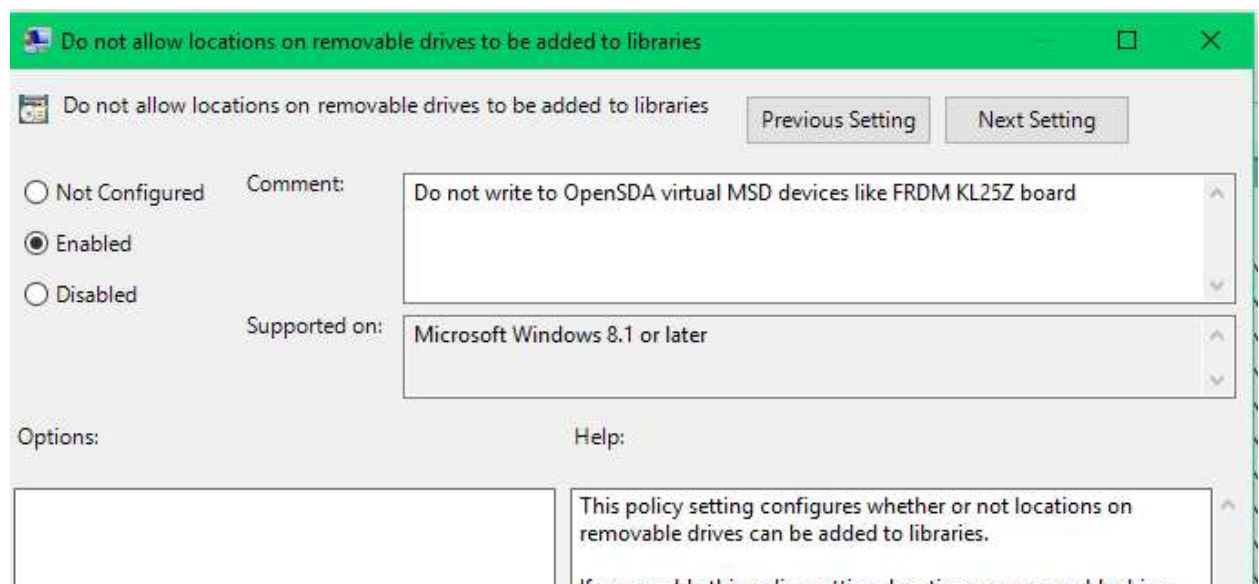
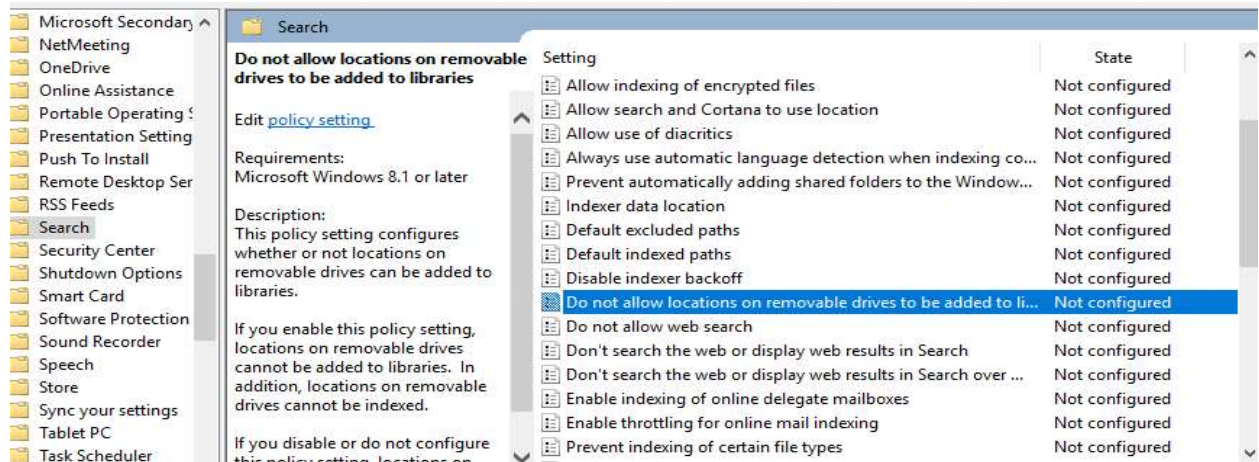


Observe that the application version is 0.00 that means there is no debug application any more on the board and hence you can't debug anymore. This should confirm that your board is definitely bricked.

2. We need to prevent windows 10 to write anything to the virtual msd device. Press Windows+R to open 'Run' and type 'gpedit.msc' to launch the Microsoft configuration console. In case you do not have gpedit.msc enabled. Follow this tutorial -> <https://www.youtube.com/watch?v=J28r5u5Wqy4>
3. This opens a Local Group Policy Editor. Go to **Computer Configuration -> Administrative Templates -> Windows Components**, and scroll all the way down to 'search' and click on it.



Click on 'Do not allow locations on removable devices to be added to libraries' option and double click on it.



Write an informative comment in the comment section and change from 'Not Configure' to 'Enabled'. Press Apply and then Ok. Now under the search settings, the options should show Enabled.

4. Restart your PC and plug your device in bootloader mode. The green LED now blinks at 1Hz. Phew!! Still a little more to go!

Follow the normal steps as mentioned under the setup of KL25Z.