Compile and upload using MBED interface a new bin file to Kinetis K64 board

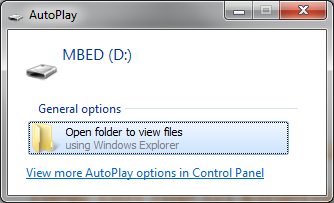
This document shows how to compile a new project in the kinetis SDK and upload it using MBED drag and drop interface.

Note , if you prefer to work with the KDS and download firmware from the debugger , this procedure is not applicable.

This working mode is for directly copy the bin file into the MBED disk and run it without the debugger.

**Prerequisites**: kinetis IDE 3.0 and **kinetis SDK 2.0**

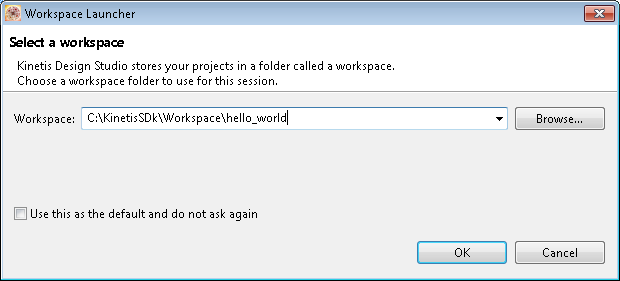
Kinetis K64 uses MBED interface. It will appear as a new disk drive at your system:



If you don’t see it and windows does not find the drivers for it, you can download it from here:

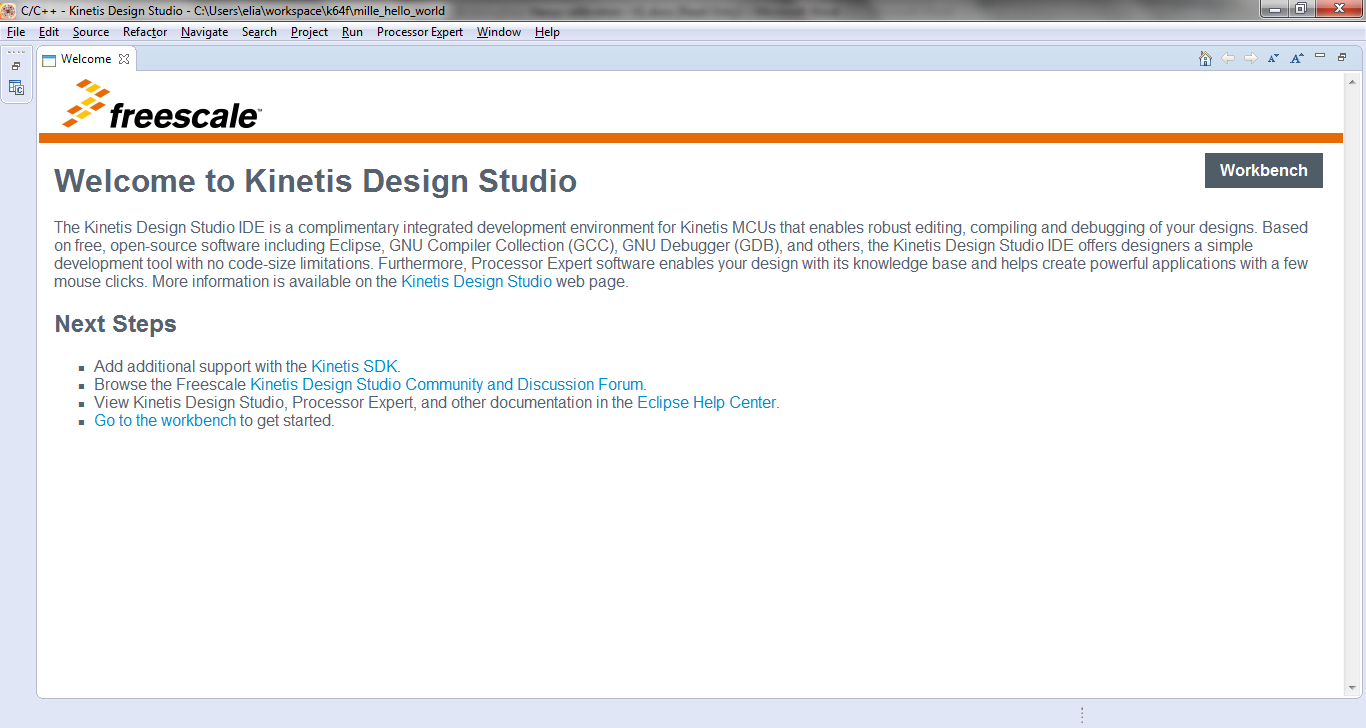
<https://developer.mbed.org/handbook/Windows-serial-configuration>

Open KDS studio and select new working space:

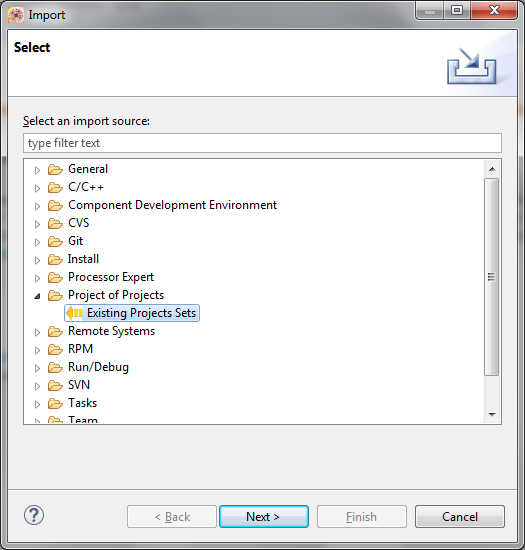


Press ok to continue:

From this screen:



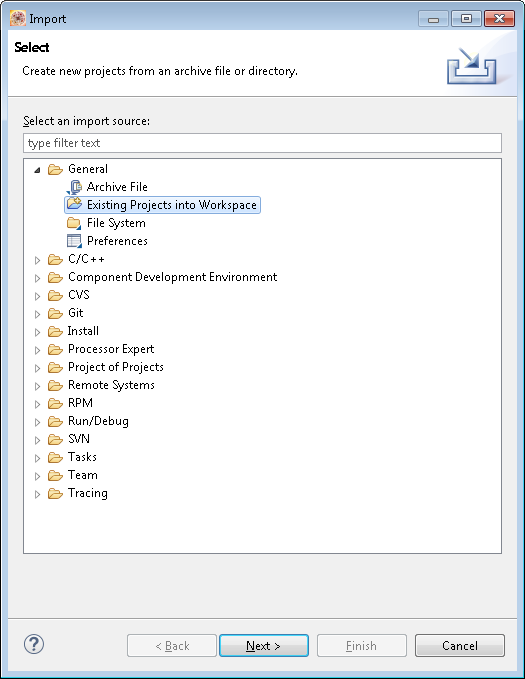
Select file->import



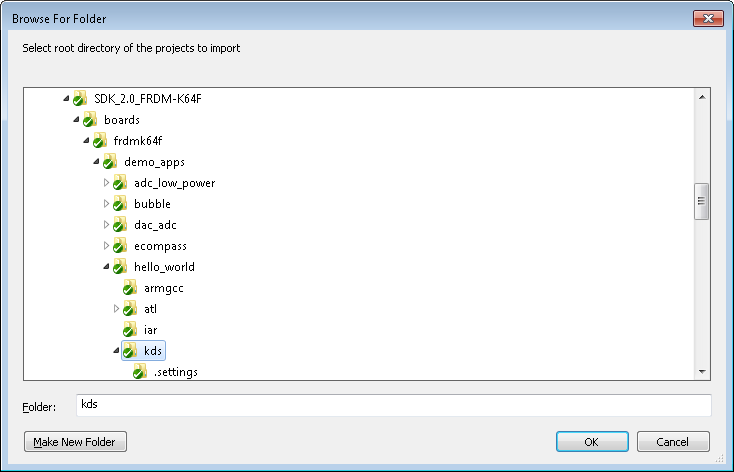
Select Project of Projects -> Existing Projects Sets and press next.

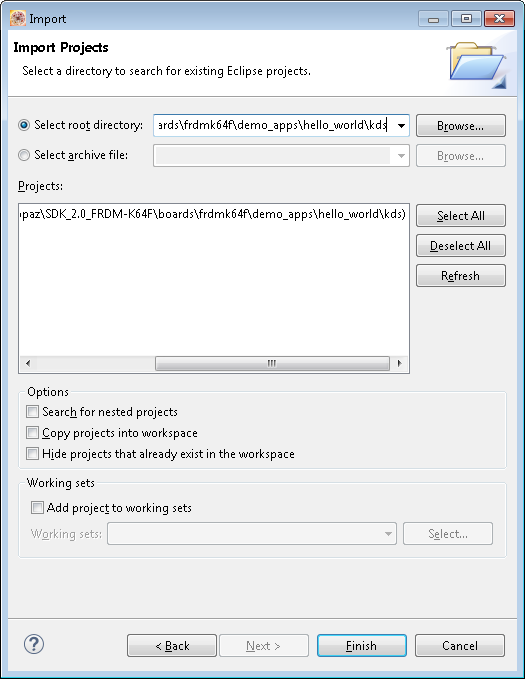
In the new SDk 2.0 , we don’t need to select Projects of projects any more:

We can select existing project into workspace



Select he KDS directory, otherwise it will let you select all the evironments ( IAR )





Select the KDS directory of the kinetis SDK hello world sample

In my computer it is located here:

C:\KinetisSDK\_2.0\_FRDM-K64F\boards\frdmk64f\demo\_apps\hello\_world\kds

And open the hello\_world\_frdmk64f.wsd file

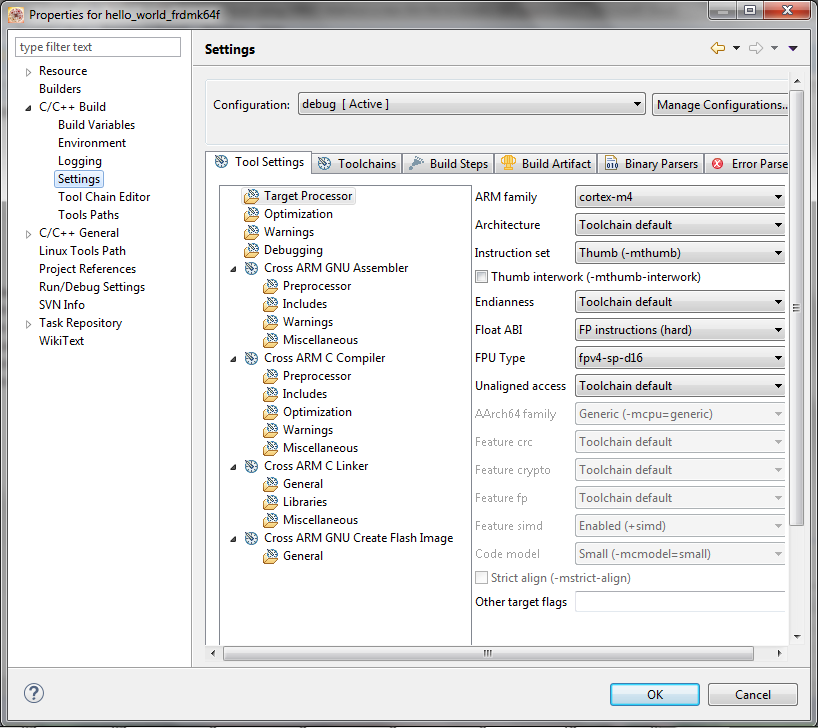
And click finished

The MBED interface uses BIN file.

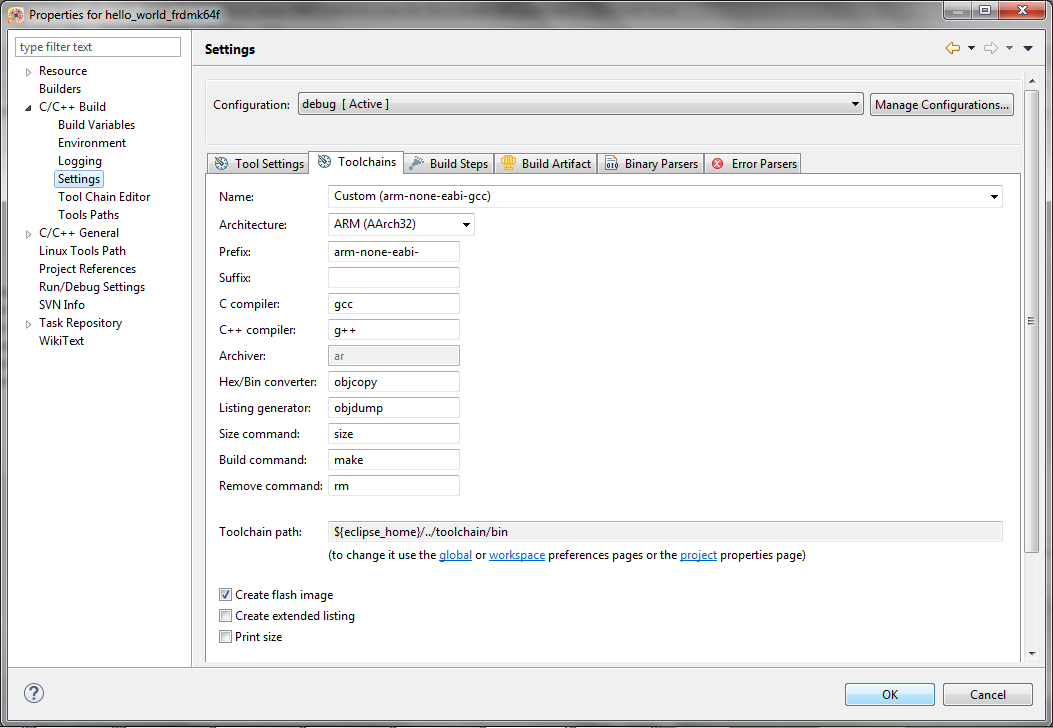
The current output generation format is elf file and we need to change the settings before we compile our project.

Select from menu:

Project ->properties-> C/C++ build -> settings:



Go to tool chain and select **Create Flash Image**

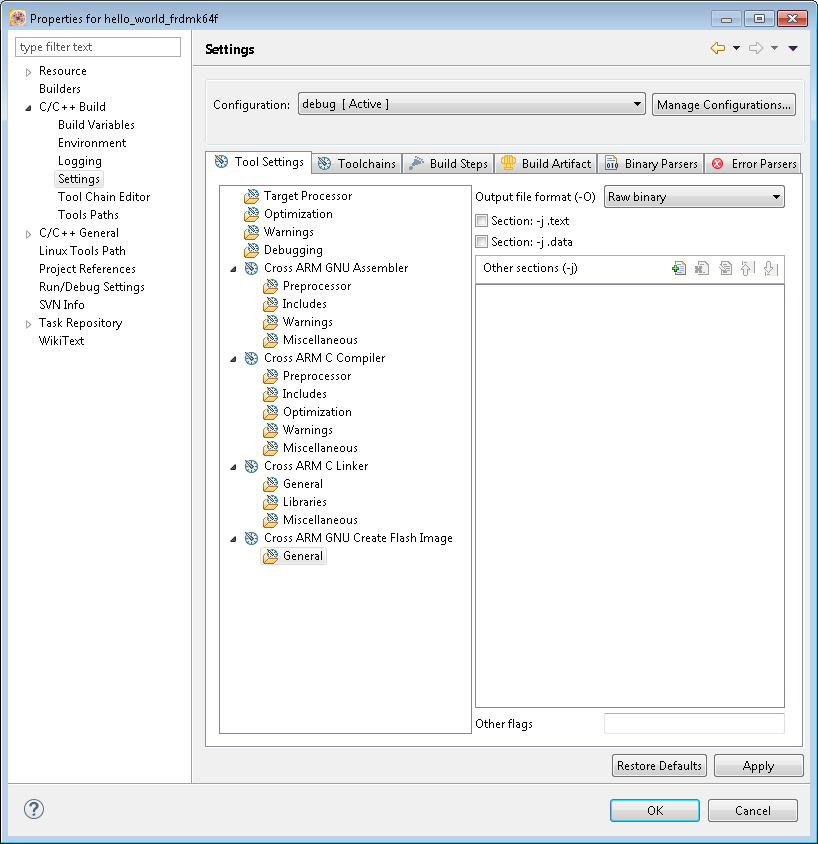


Press OK

And again, go to Project ->properties-> C/C++ build -> settings:

Now to Tools settings->general ( at the end)

Select output file format to row binary



And select OK

Compile your project:

Project->build all

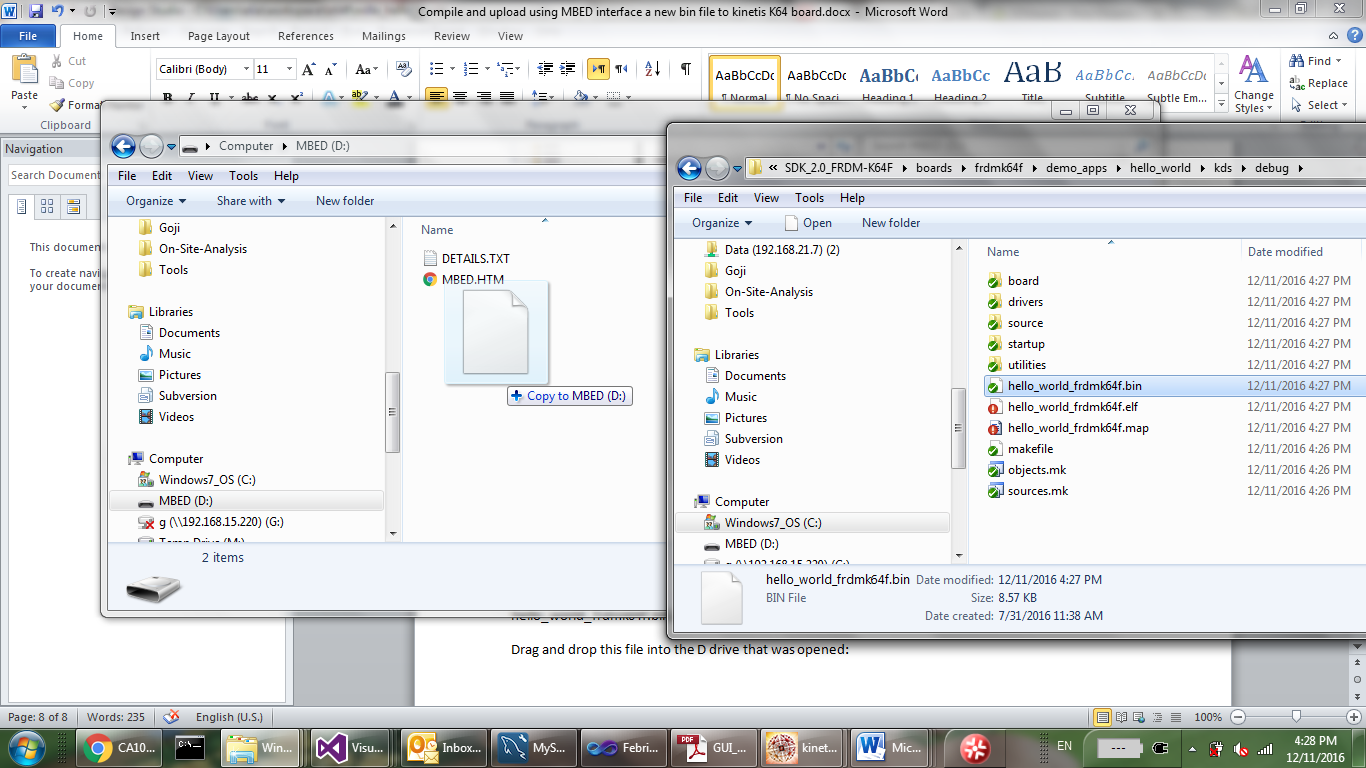
The output will be generated at:

C:\SDK\_2.0\_FRDM-K64F\boards\frdmk64f\demo\_apps\hello\_world\kds\debug

File name:

hello\_world\_frdmk64f.bin

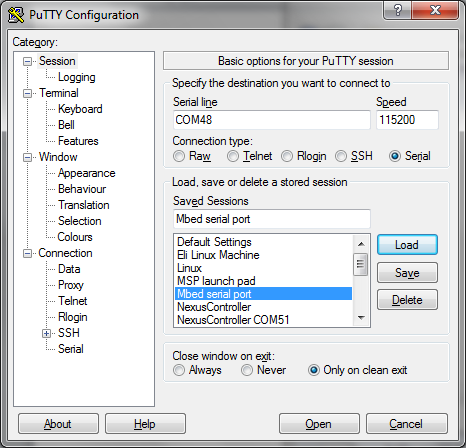
Drag and drop this file into the D drive that was opened:



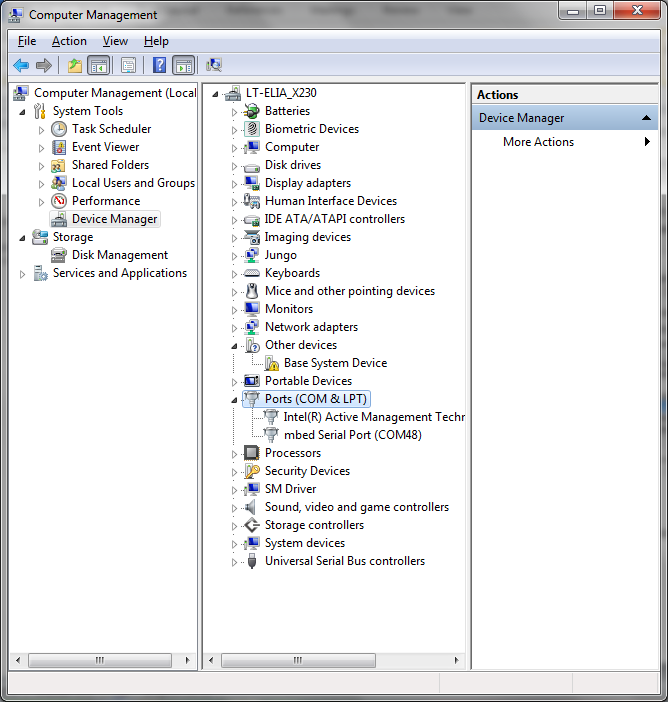
When the build finished, open putty terminal (download it from the internet)

You can use different serial terminal although this manual refer to putty.

Configure putty to the correct port and speed to 115200



To understand which is the comport of the MBED serial , open device manager:



You will see the mbed serial port and its port name in brackets.

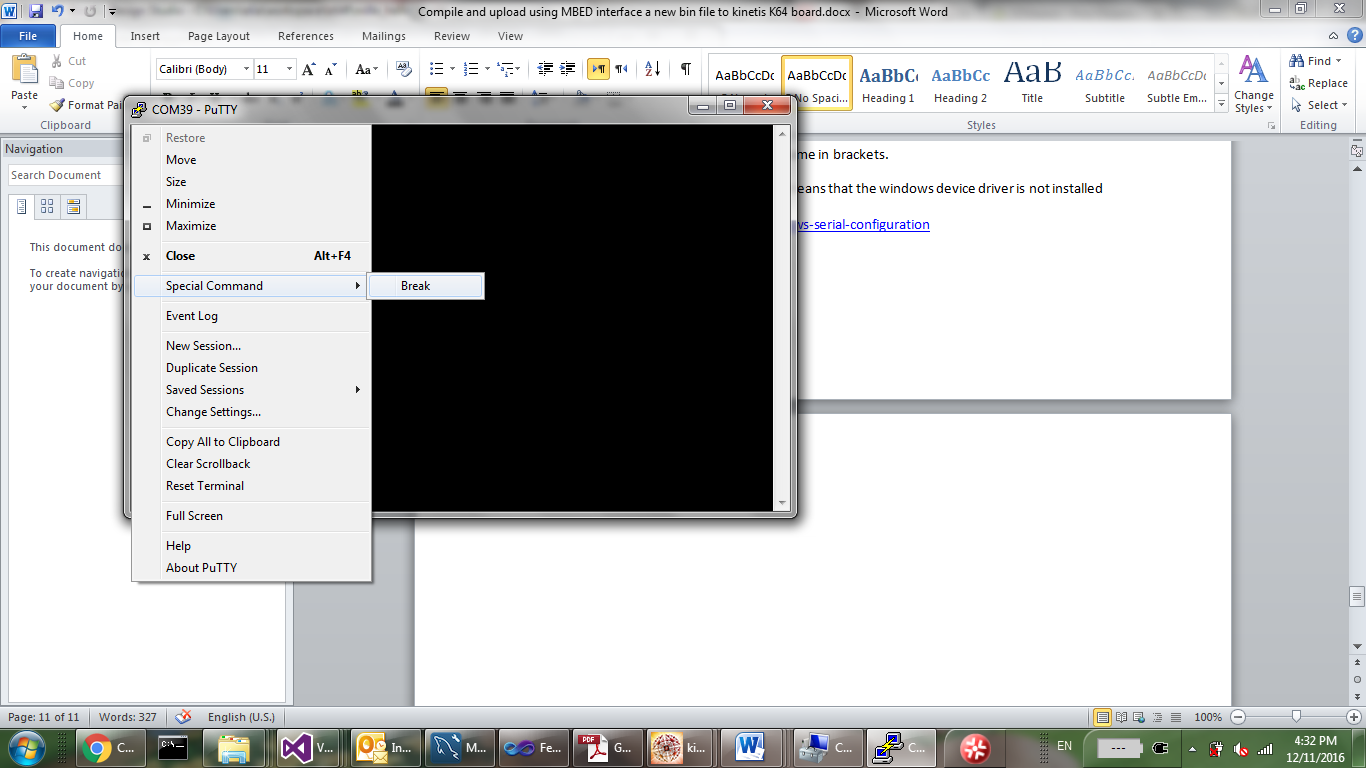
Note , If you don’t see the MBED serial port it means that the windows device driver is not installed

Again , you can download it from here:

<https://developer.mbed.org/handbook/Windows-serial-configuration>

You can now choose how to reset your software:

1. By pressing the reset button on the board.
2. Using putty break command:



Special commands->break

I change a bit my hello world to print the following:

**int** **main**(**void**)

{

**char** ch;

/\* Init board hardware. \*/

BOARD\_InitPins();

BOARD\_BootClockRUN();

BOARD\_InitDebugConsole();

PRINTF("HELLO WORLD\n\r");

**while** (1)

{

ch = GETCHAR();

PUTCHAR(ch);

}

}

And it does print as we can see



Ways to improve your working environment.

Due the fact the you will probably compile and need to copy a new firmware every 5 minutes you can create a small windows app (could be in c# or VB)

That will do the copy for you



This small up located at the right corner, just do simple file copy from the location of the bin file into drive d:\

If you need this application, please contact us.

The source and target are configurable via XML.

Please read next:

The boot loader that does not erase the flash to replace your k64 boot loader.

To create a c# app with button , just create a new windows form app with visual studio

Add a button and in the button click add:

task = new Thread(() =>

{

try

{

File.Copy(m\_sourceFile, @"d:\frdmk64f.bin");

}

catch (Exception err)

{

fail[1] = true;

}

});

Task.Start();

Note, that the name of the target bin is not important

sourceFile is where the binary produce by KDS