Quick Start Guide

– Using MCUXpresso SDK with PINs&CLOCKs Config Tools

By Jennie Zhang

MCUXpresso Software and Tools includes three parts:

MCUXpresso IDE - Eidt, compile, debug and optimize in an intuitive and powerful IDE. Free Eclipse and GCC-based IDE for C/C++ development on Kinetis and LPC MCUs. Learn more at : http://www.nxp.com/mcuxpresso/ide

MCUXpresso SDK - Runtime software including peripheral drivers, middleware, RTOS, demos and more. The software framework and reference for Kinetis & LPC MCU application development. Learn more at: http://www.nxp.com/mcuxpresso/sdk

MCUXpresso Config SDK – Online and desktop tool suite for system configuration and optimization. Integrated configuration and development tools for LPC and Kinetis MCUs. Learn more at: www.nxp.com/mcuxpresso/config

Taking good use of these three tools together will significantly speed up our program developing, and also make our work easy and fun.

In this article, I will take an example with LPCXpresso54608 board, illustrating how to use online PINs Config and Clocks Config tool with MCUXpresso SDK from start, step by step.

With this article, you will learn:

- How to use the Pins Config and Clocks Config tools to generate the initialization codes automatically.
- How to start a new SDK project with support of Pins Config and Clocks Config tools generated files.

Outline:

- Download and install MCUXpresso SDK
- Create a new SDK project
- Using online Pins Tool generate config files.
- Using online Clocks Tool generate config files.
- Using the generated config files for the SDK project
- Demonstration : print string to terminal via UART

1. Download and install MCUXpresso SDK

1.1 Download and install MCUXpresso

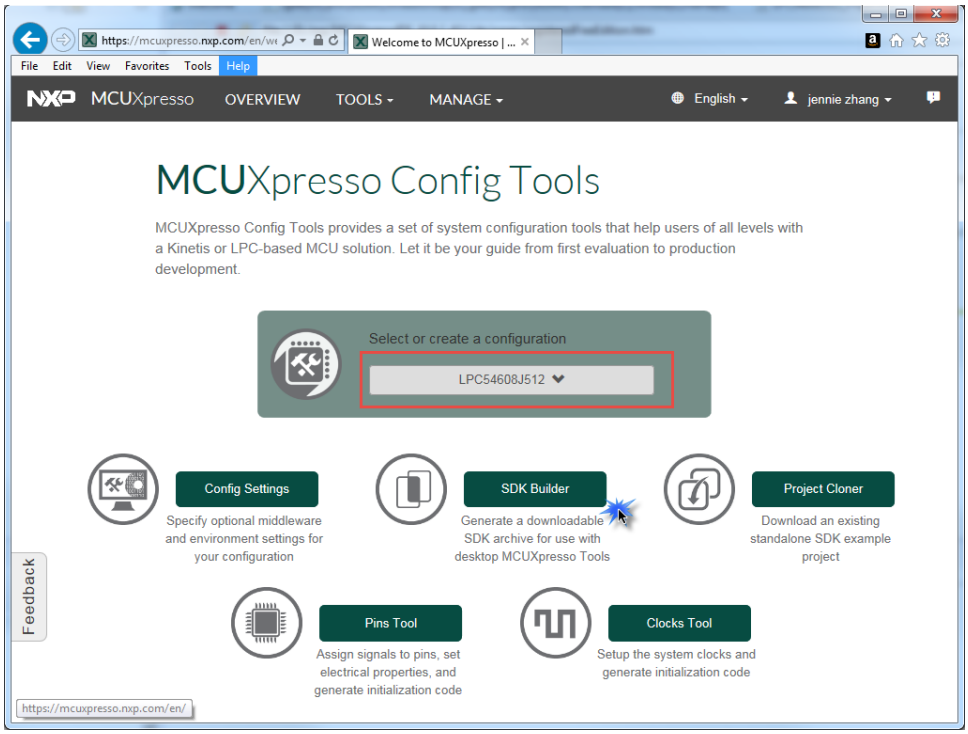
The latest MCUXpresso Integrated Development Environment (IDE) v10.0.2 in this url: http://www.nxp.com/mcuxpresso/ide

NOTE: MCUXpresso v10.0.2 is a standalone install version; it's not possible to upgrade any previous version to v10.0.2 with an update.

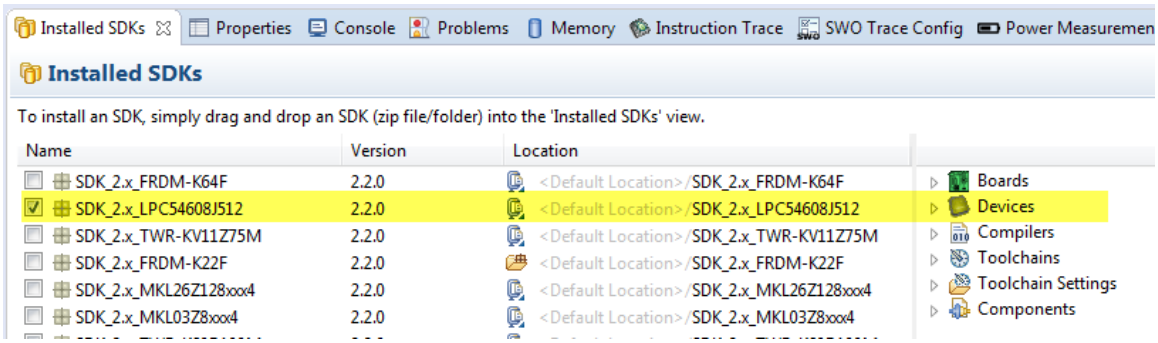
The MCUXpresso IDE is available in full-featured free (code size unlimited) and affordable professional editions. We can use

1.2 Download and install MCUXpresso SDK 2.2 for LPC54608

Go to https://mcuxpresso.nxp.com/en/welcome download SDK builder for LPC54608J512



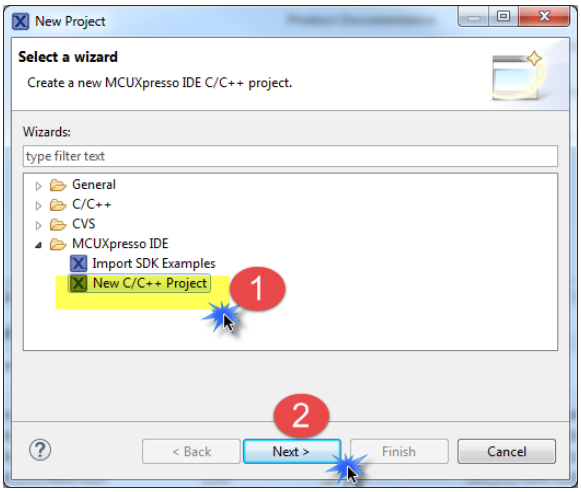
In MCUXpresso IDE, to install an SDK, simply drag and drop an SDK (zip file/folder) into the "installed SDK's view"



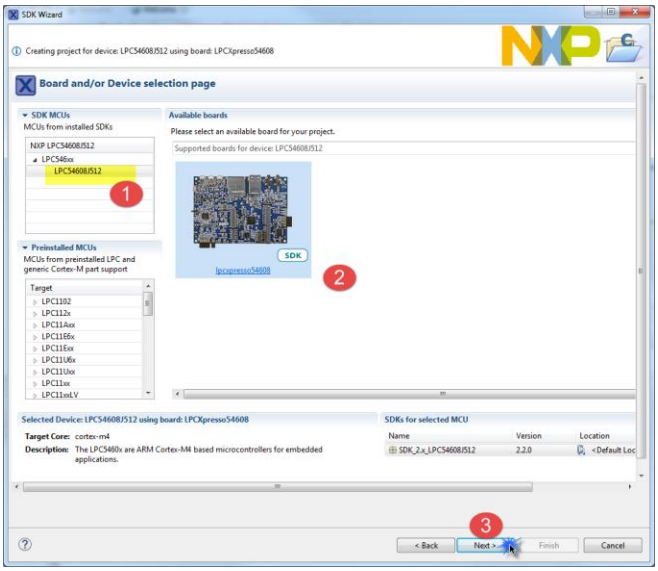
Restart IDE, Make sure SDK_2.x_LPC54608J512 checked.

2. Create a new LPC54608J512 SDK project

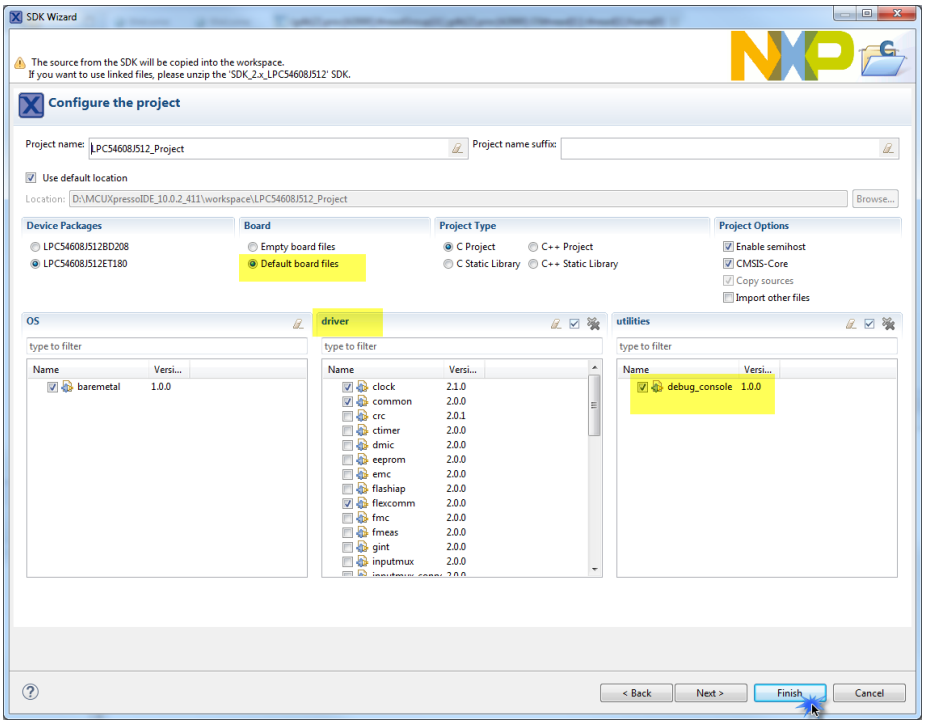
2.1 Go to IDE menu “File”, “New”, “Project…”, “MCUXpresso IDE”, “New C/C++ Project”



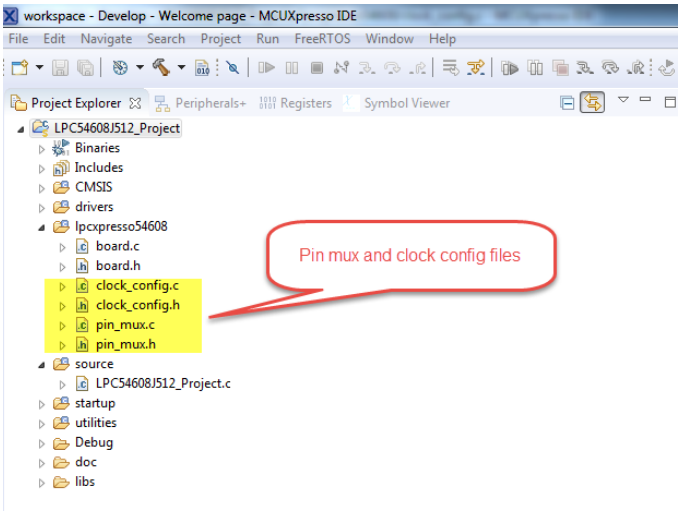
2.2 Choose the device



2.3 Make sure check the “Default board files”, make sure the clock related drivers are seclected.



2.4 In the new created project, under “Project Explore”, we see below four pin_mux and clock_config files. We will replace them with the files generated by PINs&CLOCKs Config tools. See the following sections.



3. Using online Pins Tool generate config files.

Pins Tool - assigns internal signals to external pins, sets electrical properties, I/O conflict resolution options and generates ANSI-C source code that drops into the MCUXpresso SDK environment.

Pin Tool includes both desktop and online version. In this article, we use online version to demonstrate its usage.

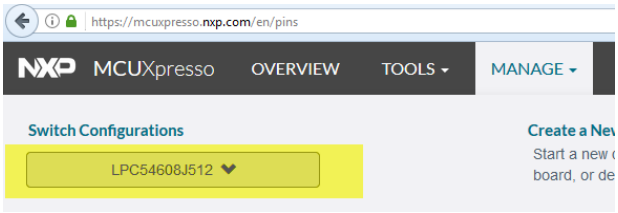
Config Purpose:

- We will use Pins tool to configure FLEXCOMM0 pin B13 for UART RDX and pin A2 as UART TXD.

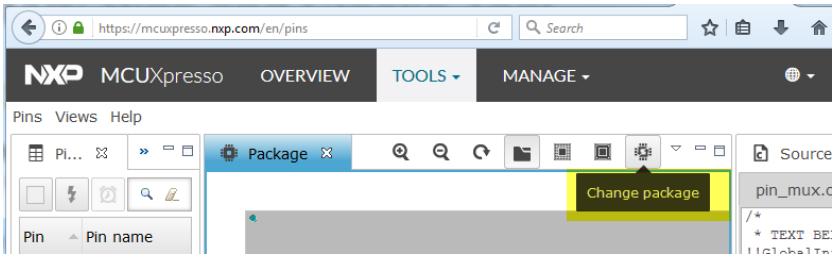
3.1 Login NXP website and access pin tool web version:

https://mcuxpresso.nxp.com/en/pins

Under “MANAGE”, “Switch Configurations”, make sure “LPC54608J512” selected.



Make sure choose the right package, use “Change package” for the right one.



3.2 Config Pins

Configure FLEXCOMM0 pin B13 and pin A2 as SCI RXD and TXD.

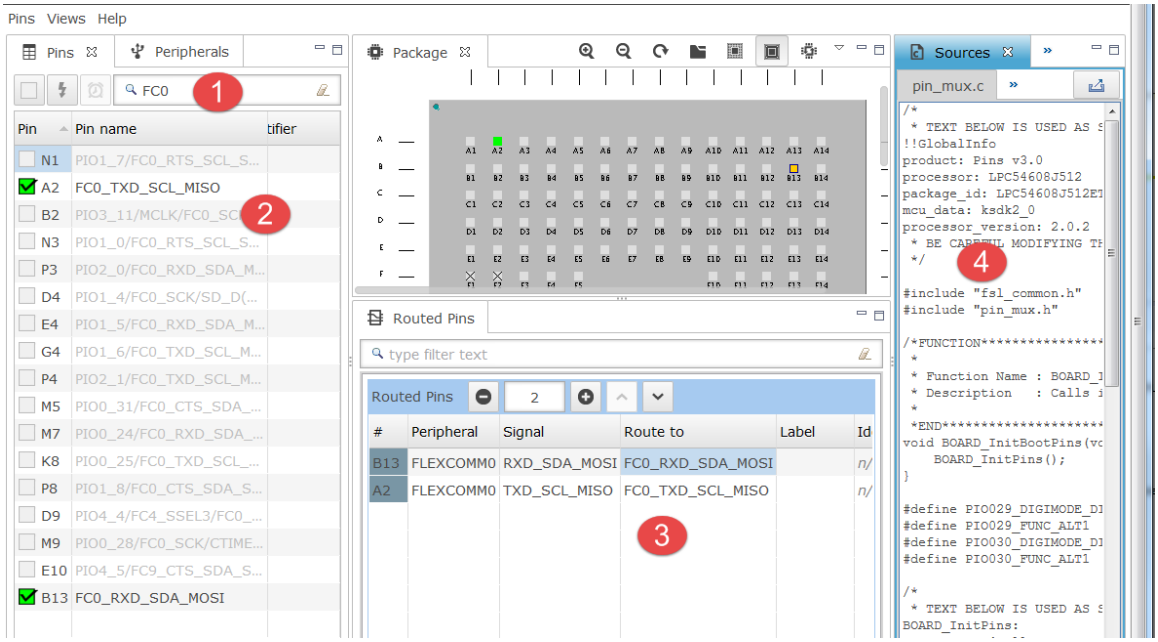
(1) Search the pin number or pin name or module name in search field.

(2) Select the pin you want to config.

(3) Route the pin.

(4) Then we will see the pin files generated in Sources window automatically.

see below screenshot.



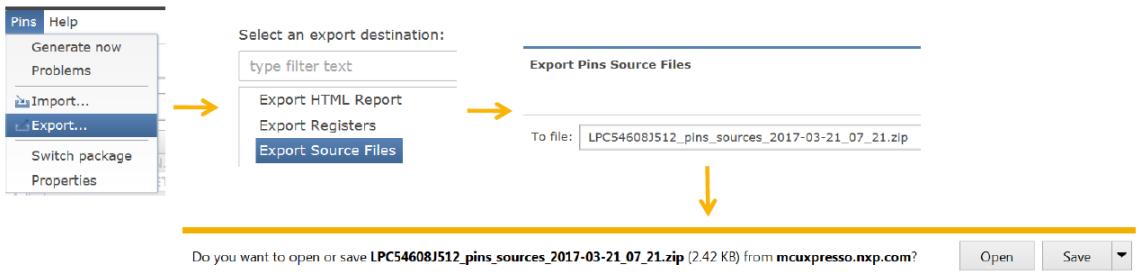
3.3 Export the source files.

(1) Click the “Export…” item from the menu named “Pins”

(2) Select “Export Source Files” and click “Next” in the pop window.

(3) Click “Finish” with the default or changed name of exporting the .zip file.

(4) Save the .zip file to the local PC and extract the source files which can be used in your project.



The .zip file includes pin_mux.c and pin_mux.h. We will use it in our project in section 5.

State Indicator: Pin Tool

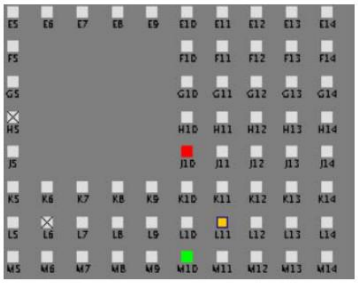
• Color Coding to indicate the state of the pins in package view:

Green indicates pin/peripheral is routed

Yellow indicates pin/peripheral is currently selected

Red indicates an error

Light grey indicates pin/peripheral is available but is not currently routed



Routed Pins - 4 + ^ v

#	Peripheral	Signal	Route to	Label	Identifier	Direction	Mode
L11	SUPPLY	VDD	VDD6		n/a	Not Spec...	n/a
J10	DMA0	TRIG, 1	PIO1_26		n/a	Input	PullUp
M10	FLEXCOMM2	SCK	FC2_SCK		n/a	Not Spec...	PullUp
J10	FLEXCOMM2	CTS_SD...	FC2_CTS...		n/a	Not Spec...	PullUp

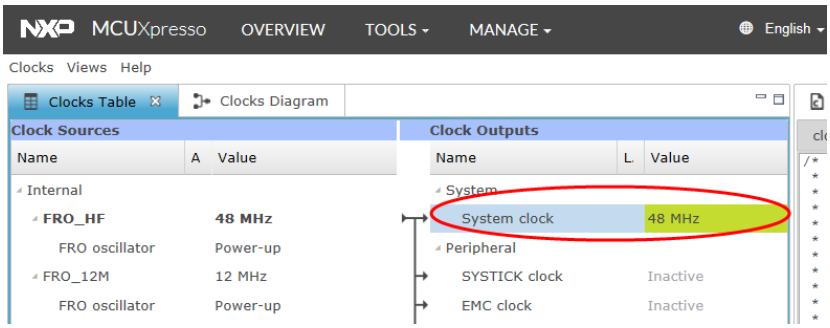
4. Using online Clock Tool generate config files.
   - Configuration of clock settings of peripherals – clock source, clock output, pll/divider values etc.

Config Purpose:

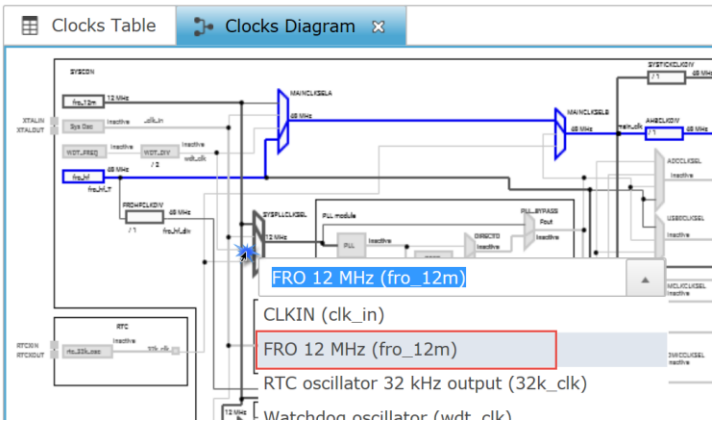
- Set system clock frequency = 48 MHz with FRO_12M as clock source.

4.1 Login NXP website and access Clock tool web version:
https://mcuxpresso.nxp.com/en/clock
Under “MANAGE”, “Switch Configurations”, make sure “LPC54608J512” selected.

4.2 Under “Clocks Table” Tab, Set system clock as 48MHz



Under “Clocks Diagram” Tab, Select PLL system clock source as FRO_12M

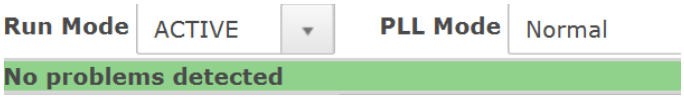


4.3 Export the source files

Then, we can see the source files generated. Refer 3.3 Export the source files to export the source file in zip.

The .zip file includes clock_config.c and clock_config.h. We will also use it in our project in section 5.

Note: make sure “No problems detected” bar appears. It indicates all your clock configuration is correct.



5. Using the generated config files for the SDK project

Below is the steps of how to use config files with a new created SDK project:

(1) With Pins and Clocks Config Tools, they generated 4 files totally:

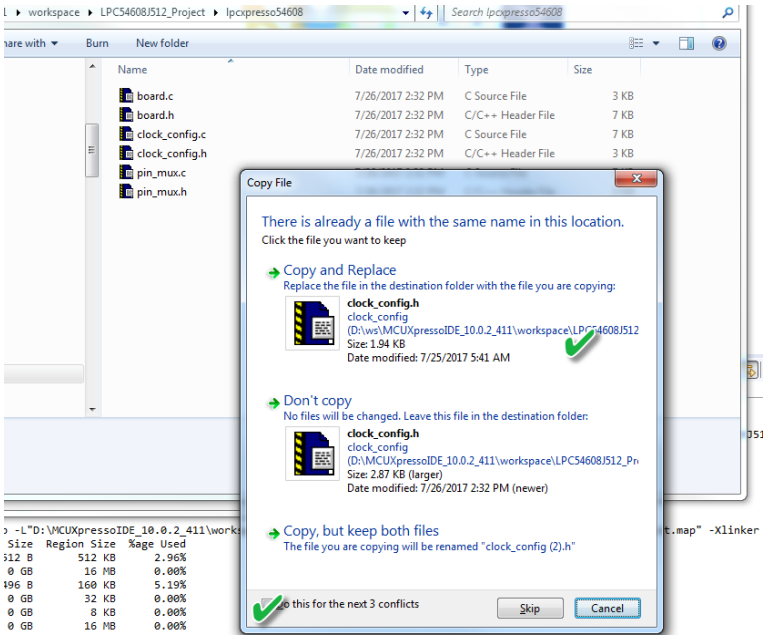
pin_mux.c

pin_mux.h

clock_config.c

clock_config.h

Next, we copy these 4 files and replaced the same files under the project we generated in section 2.4



(2) Reopen the project with MCUXpresso IDE, in main file, include below code to the source code.

#include "fsl_debug_console.h"

Add print code in main

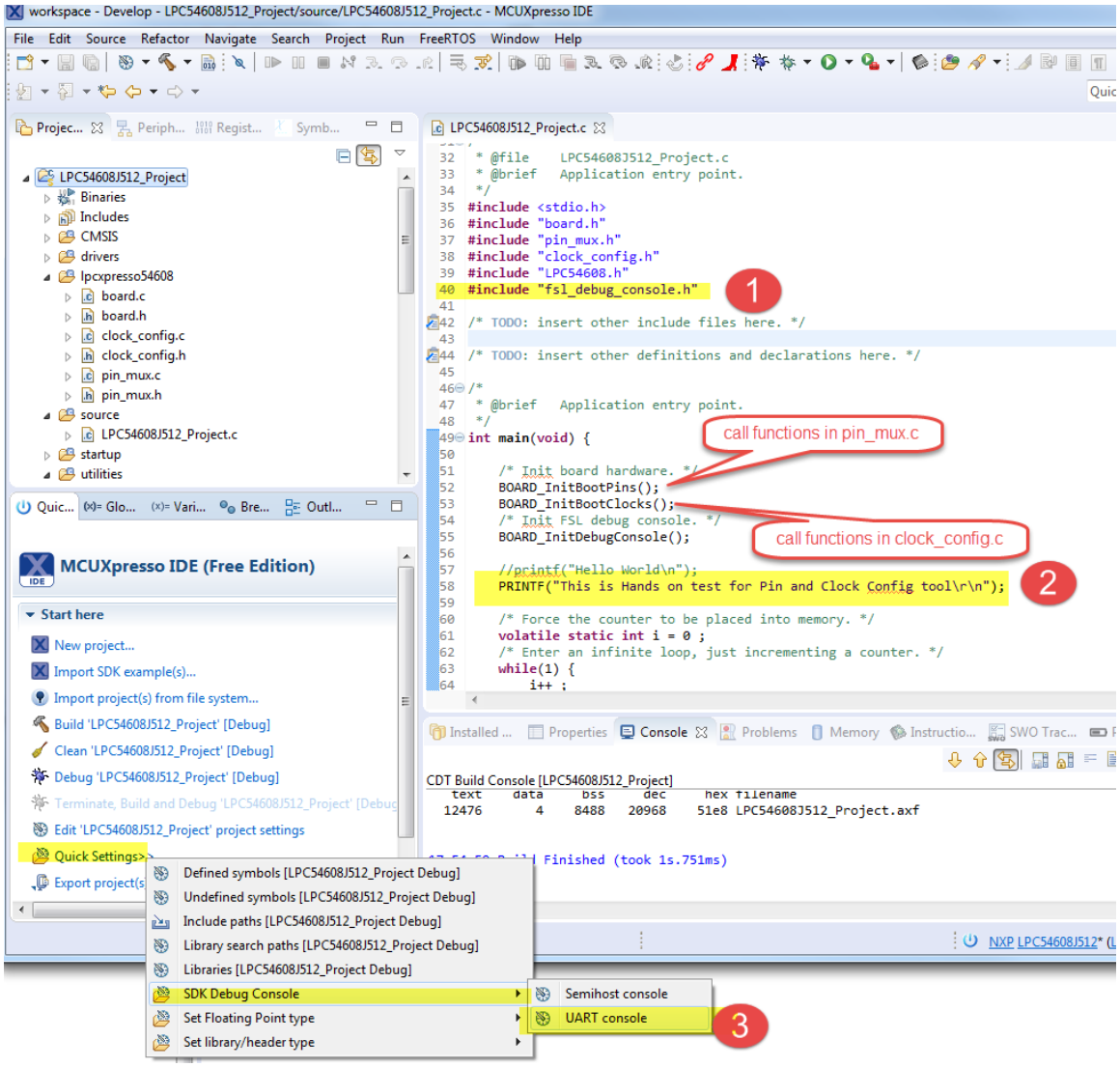
PRINTF("This is Hands on test for Pin and Clock Config tool\r\n");

In the source code,

BOARD_InitBootPins() calls initial code in pin_mux.c/h;

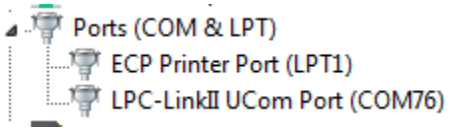
BOARD_InitBootClocks() calls initial code in clock_config.c/h;

(3) In Quick Setting, set debug console as UART. (The default setting is to print to console window)

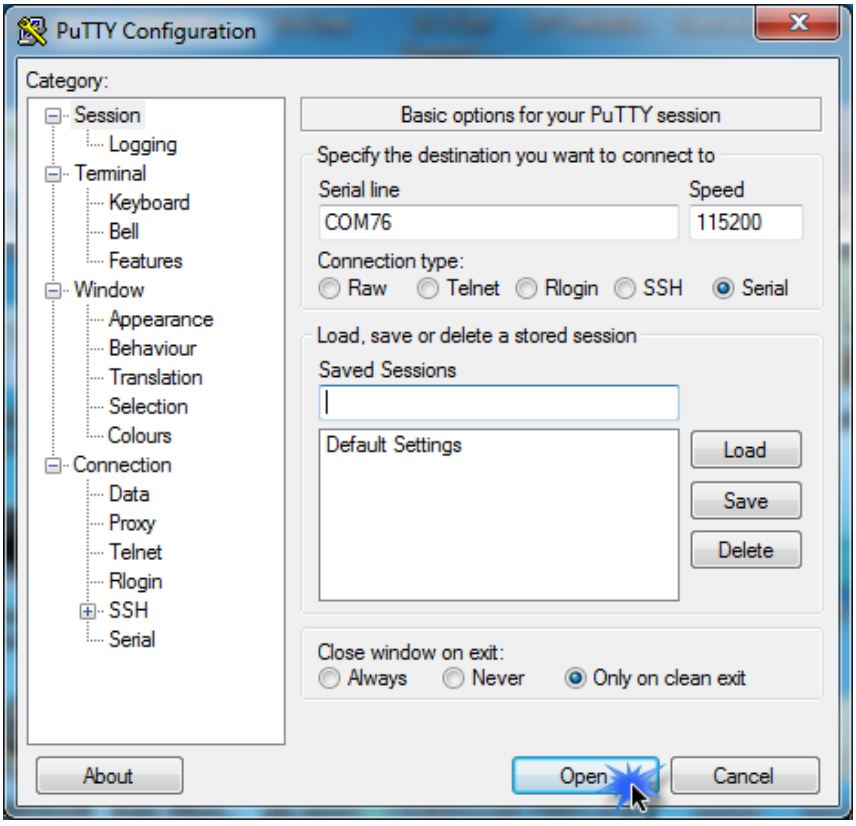


6. Demonstration : print string to terminal via UART

   Check the COM port number which the LPC-LinkII is simulated in “Device Manager” on your computer:



Set PC UART terminal baud rate as 115200



Download and run program, we can see output in Putty terminal.