**Introduction**

In this hands-on lab, we will use Kinetis Design Studio to build the MQX and KSDK libraries, and then build and test an MQX application example.

**Resources**

PC running Windows 7 with the following software:

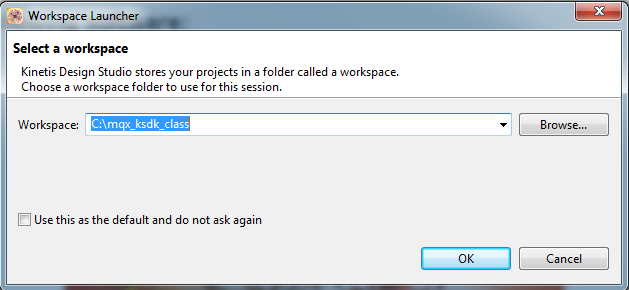
• Kinetis Design Studio (KDS) v2.0.0

• MQX for KSDK v1.1.0

Hardware:

• FRDM-K64F

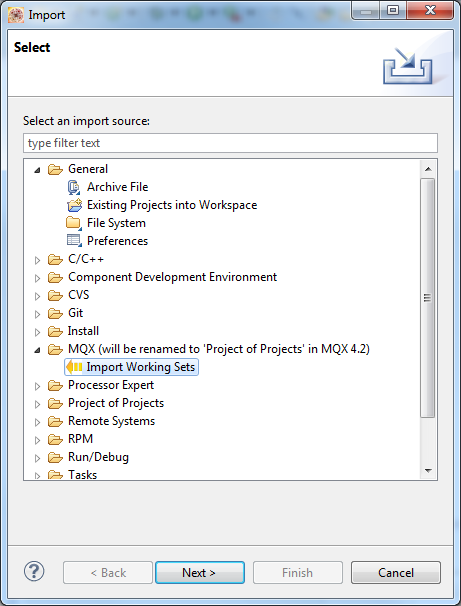
1. **Build MQX Working Set**
   1. Open KDS
   2. Set the workspace directory to **C:\mqx\_ksdk\_class** (or your choice of other new directory location) and click on OK.



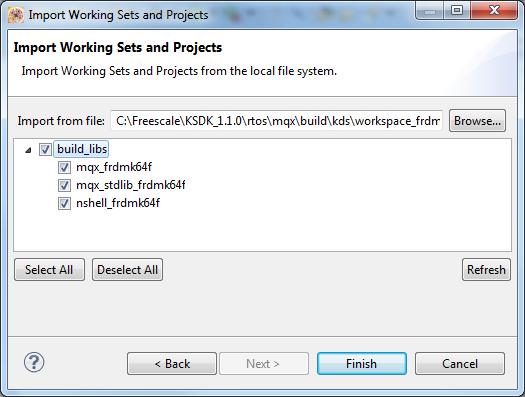
* 1. Click on the “Workbench” icon to go to the main Workbench screen. This only has to be done the first time a new workspace location is used.



* 1. Import the MQX libraries working set. A working set is a collection of Eclipse projects. Use the menu **File->Import**.
  2. Under the MQX group, select **Import Working Sets**. Then click ‘Next’.



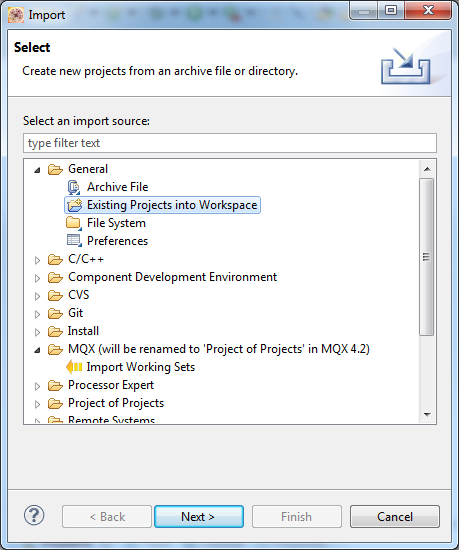
* 1. Browse to the MQX for KSDK directory **"C:\Freescale\KSDK\_1.1.0\rtos\mqx\build\kds\workspace\_frdmk64f\build\_libs.wsd"**This imports the MQX libraries used by MQX applications.



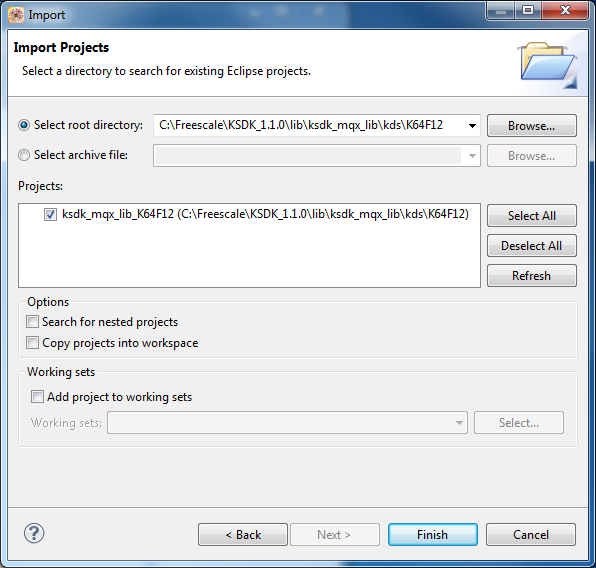
1. **Build KSDK MQX Library**
   1. Next import the KSDK MQX library. This library is not part of the MQX working set. This time we will import a single project instead of a working set. Use the KDS menu **File->Import**.

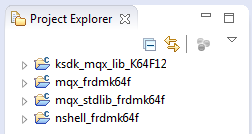
Note: If the Working Set plugin was not installed on your version of KDS, you could instead import each of the MQX libraries individually using the steps in this section.

* 1. In the General group, select **Existing Projects into Workspace**.

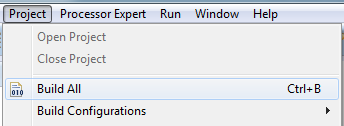


* 1. Browse to the root directory **C:\Freescale\KSDK\_1.1.0\lib\ksdk\_mqx\_lib\kds\K64F12**.
  2. Select the project **ksdk\_mqx\_lib\_K64F12**. Click ‘Finish’.



NOTE: Never check the box labeled “Copy projects into workspace” when importing KSDK or MQX projects. This will cause issues finding files when building the projects.  
  
The following projects are now imported in the workspace  


* 1. Build all the projects in the workspace. These can all be built together. Use the KDS menu **Project->Build All**.

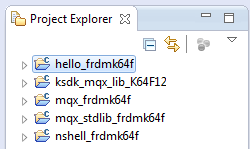


Note: You can go to the Console tab at the bottom to see the status of the build and any errors that may come up

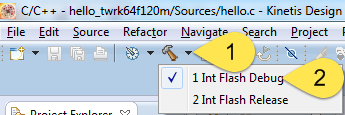
1. **Build and download MQX Application**

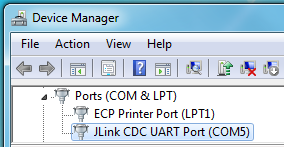
MQX includes a simple hello world application that prints to the terminal. We will use this application to learn the basics.

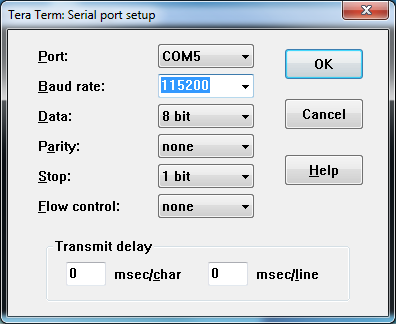
* 1. Import a project like before, using the menu **File->Import->General->Existing Projects into Workspace**. Browse to the project directory **C:\Freescale\KSDK\_1.1.0\rtos\mqx\mqx\examples\hello\build\kds\hello\_frdmk64f**.
  2. Build a single project this time. In the Project Explorer View, select the project **hello\_frdmk64f**.

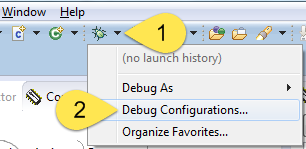


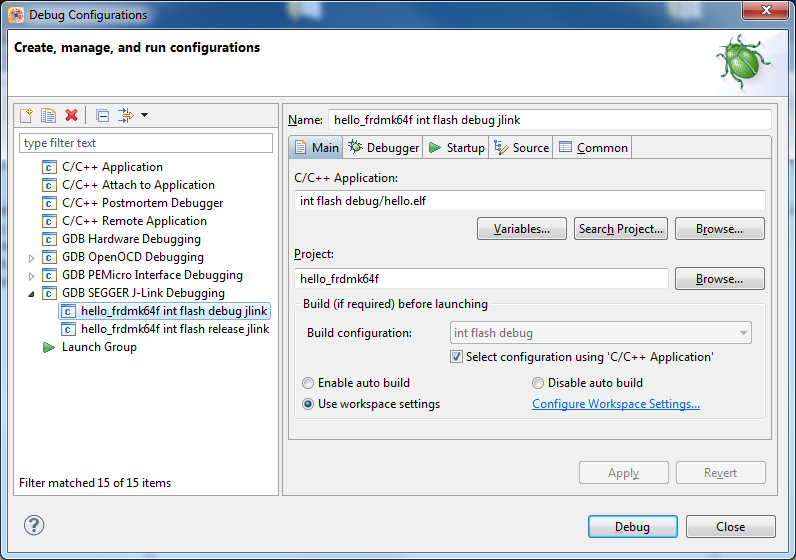
* 1. Build the application by clicking the **pull-down arrow** to the right of the Hammer icon on the upper toolbar. Then select the **Int Flash Debug** target. The Problems View should show no errors, and the Console View should show the application built.



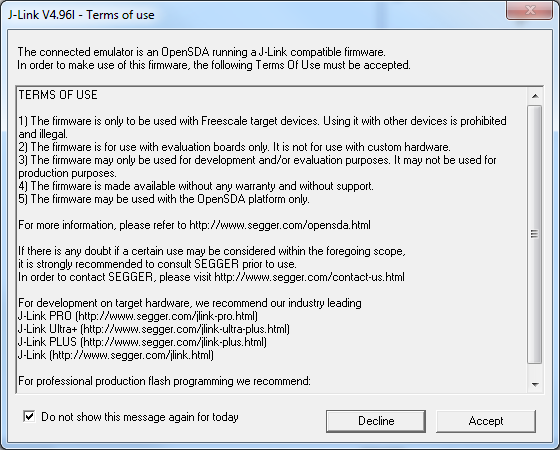
* 1. Now plug the USB cable to the FRDM-K64F board. Connect to the mini-USB connector labeled “SDA USB”. The board should have the JLink OpenSDA app loaded. Windows may need some time to load the driver.
  2. Find the COM port for the OpenSDA serial connection. Windows Device Manager will show the COM number under the Ports (COM & LPT) group. Here, the COM port is 5.  
     
  3. Connect a terminal program to this COM port



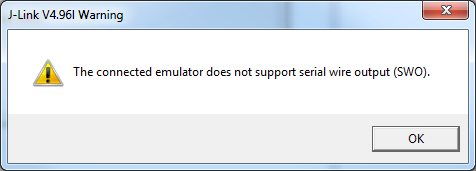
* 1. Back in KDS, open the Debug Configurations by clicking the **pull-down arrow** next to the green bug on the upper toolbar. Then select **Debug Configurations**.  
     
  2. In the left panel, select the configuration **hello\_frdmk64f int flash debug jlink**, then hit the **Debug** button.



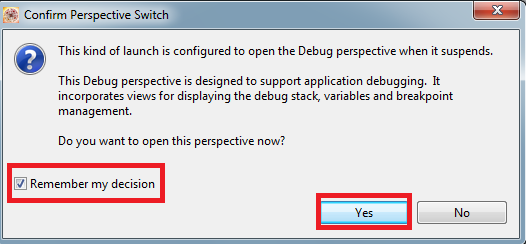
* 1. You may get the following message about using the JLink OpenSDA app. Click on the checkbox to not show the message again today, and then click Accept



* 1. Then you will get the following message. Just click OK

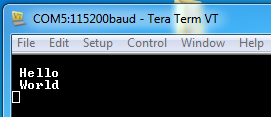


* 1. Then you may also get the following message. Click on “Remember my decision” and then click on “Yes”. This will take you to the Debug view.



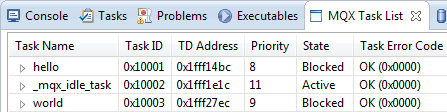
* 1. After the programming has completed and the debugger halts in main(), click the **Resume** button or press the **F8** key.



* 1. The MQX application prints to the terminal  
     

Note: If you do not see any output on the terminal, try closing Tera Term, then unplug the board. Re-plug in the board, re-open Tera Term, and then go through the debug process again. This is necessary because if the board is plugged in while Tera Term is already open, it will not properly connect to the virtual serial port.

* 1. Use MQX Task Aware Debug (TAD) to see the current state of MQX kernel and tasks. **Suspend** the debugger by pressing the Pause button.  
     
  2. Use the KDS menu to select **MQX->Task List** and then look down at the tabs at the bottom for the MQX Task List tab. You may have to click on it twice.



* 1. Explore the other TAD options in the menu MQX.
  2. When done, Terminate the debugger  
     
  3. Switch back to the **C/C++ Perspective**.  
     