Freescale Semiconductor, Inc.
User's Guide

Document Number: MKW01802154SW513QSG
Rev. 0,
11/2015

Freescale MKW01 IEEE® 802.15.4 Software

Quick Start Guide

This document is a brief presentation of the Freescale IEEE® 802.15.4 MAC/PHY Software for the MK01 wireless microcontroller platforms, version 5.1.3. This software package contains Kinetis Software Development Kit (KSDK) sources. This document covers installation of the software package, hardware setup, build and usage of the provided demo applications.

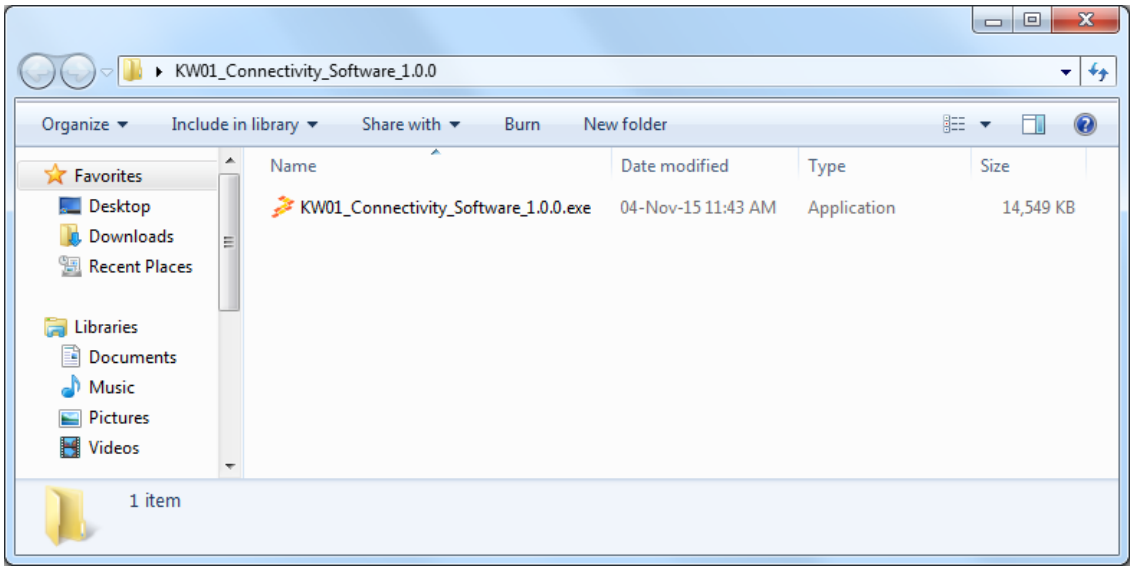
Contents

freescale™

1 Installation

This section covers the steps for a successful installation of the required software packages: Connectivity and Kinetis SDK.

The first step is to download the “KW01_Connectivity_Software_1.0.0.exe” installer.



The KW01 Connectivity Software Installer

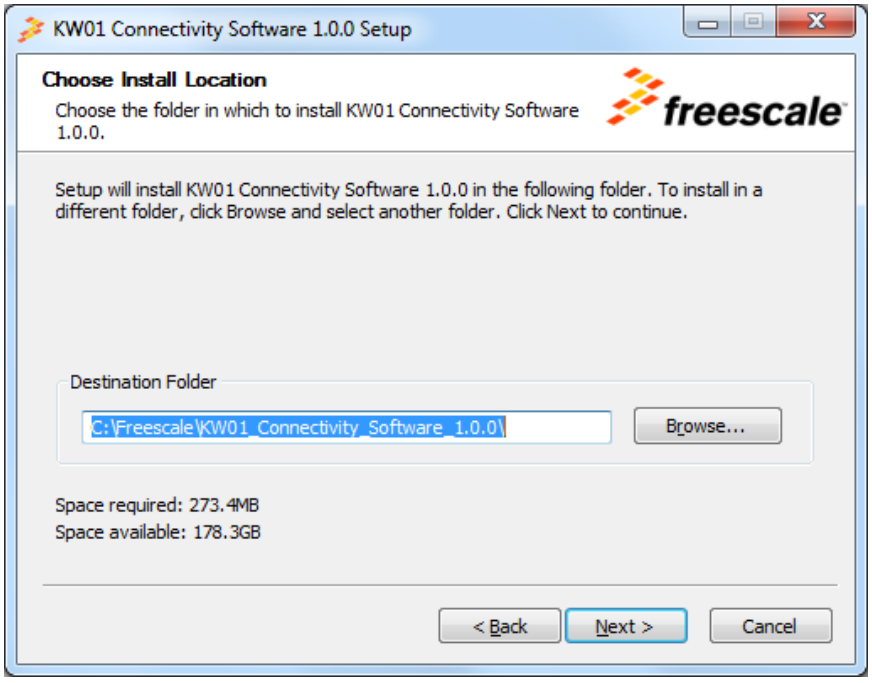
On the main screen, press the *Next* button.

KW01 Installer main screen

On the *License Agreement* screen press the *I agree* button to accept the license agreement.

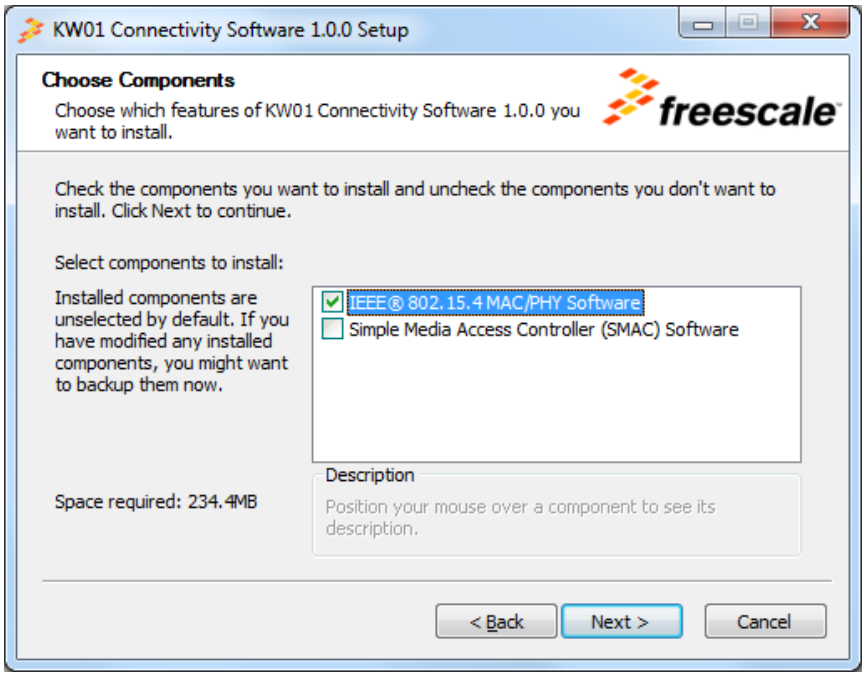
License agreement screen

On the next screen click *Browse* to select another destination folder for the KW01 802.15.4 MAC installation or click the *Next* button to continue.



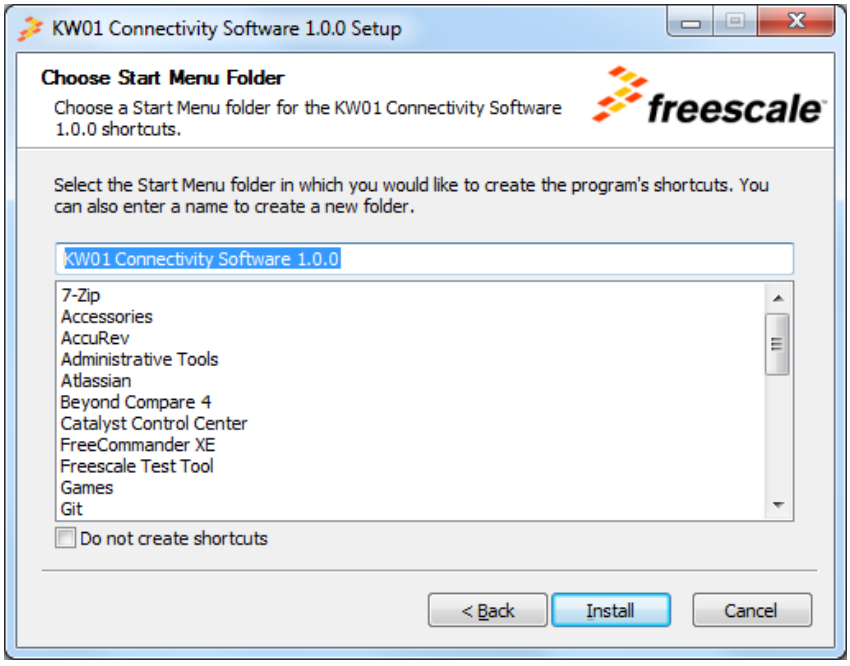
Destination folder selection screen

On the following screen select IEEE 802.15.4 MAC/PHY Software and press the *Next* button.

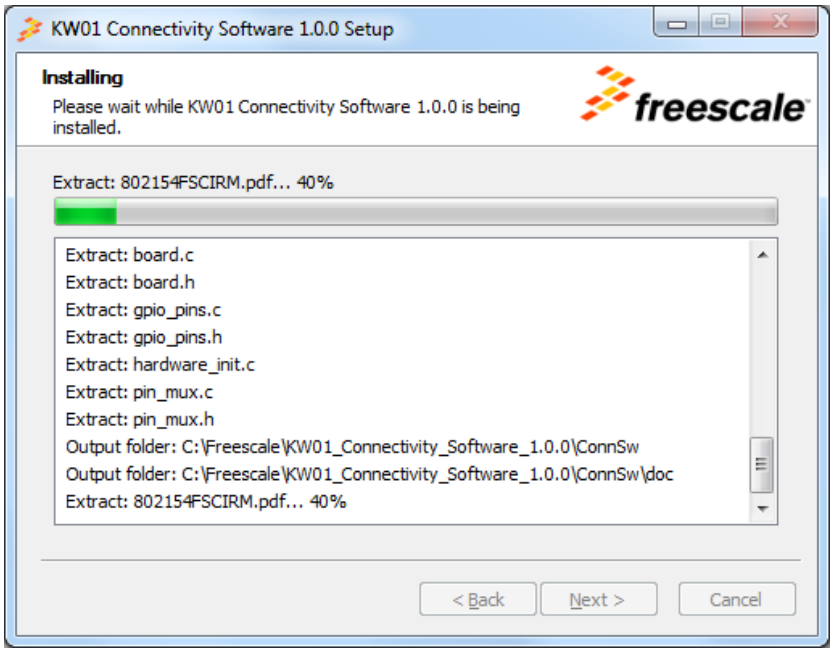


Component selection screen

Select a *Start Menu* folder and press the *Install* button.



KW01 installer Start Menu Folder selection screen



KW01 package installation process

Click *Finish* to close the installer.

KW01 802.15.4 MAC installation complete

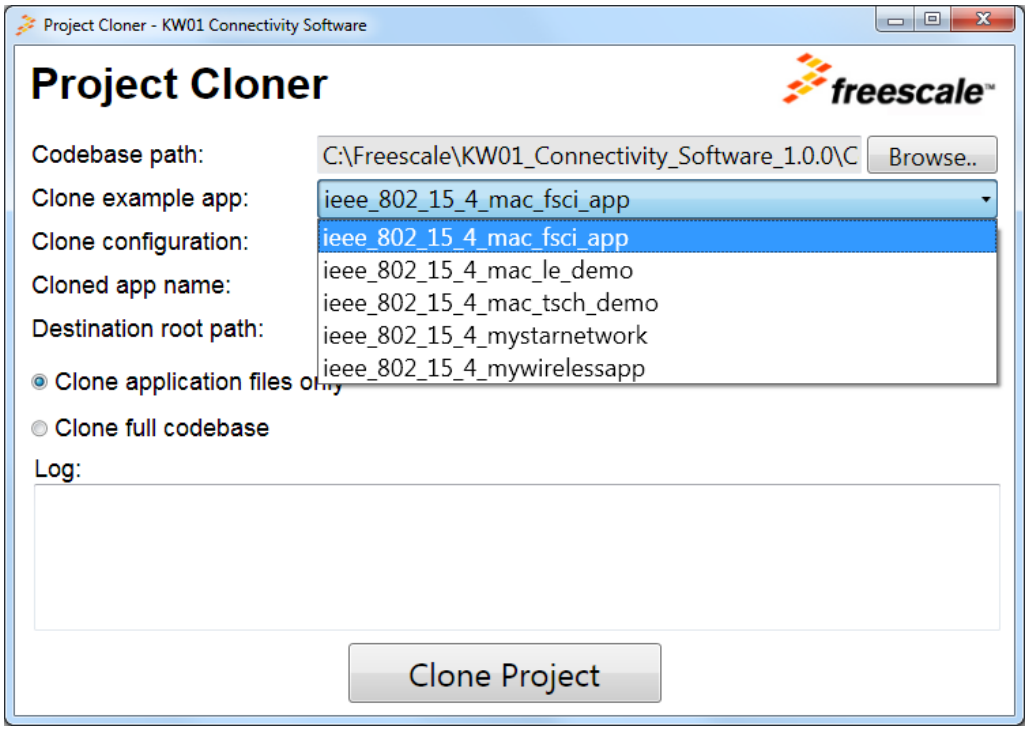
The installer automatically creates or updates the *KSDK13_FWK513_PATH* environment variable required by the KW01 MAC projects.

Once the above steps are performed, you can start using the MAC Demo Applications.

2 Cloning a project

Navigate to the KW01 Connectivity Software installation folder and run the Project Cloner application (**ConnSw\tools\project_cloner\project_cloner.exe**).

At the first run, the Project Cloner will display all the Connectivity Software example applications.



Connectivity Software example applications

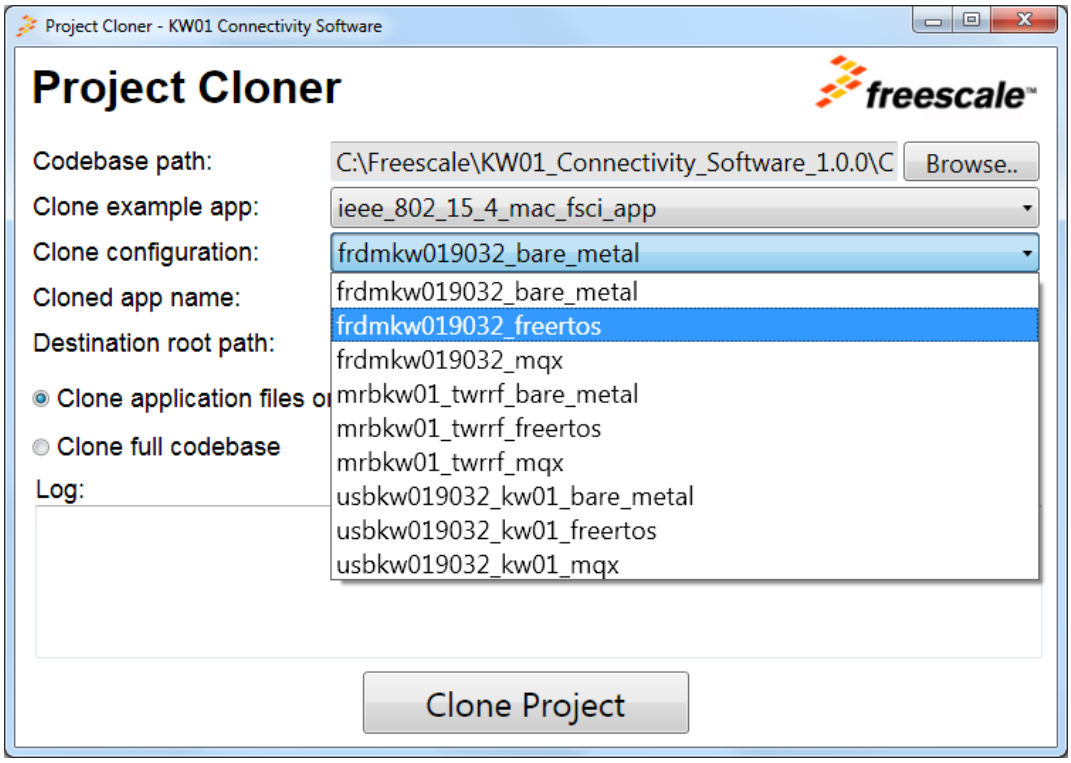
Next select the example application to be cloned (**Clone example app**), and the desired configuration (**Clone Configuration**).

After this, the **Cloned app name** text box will contain a default name for the selected application. This name can be modified to any value.

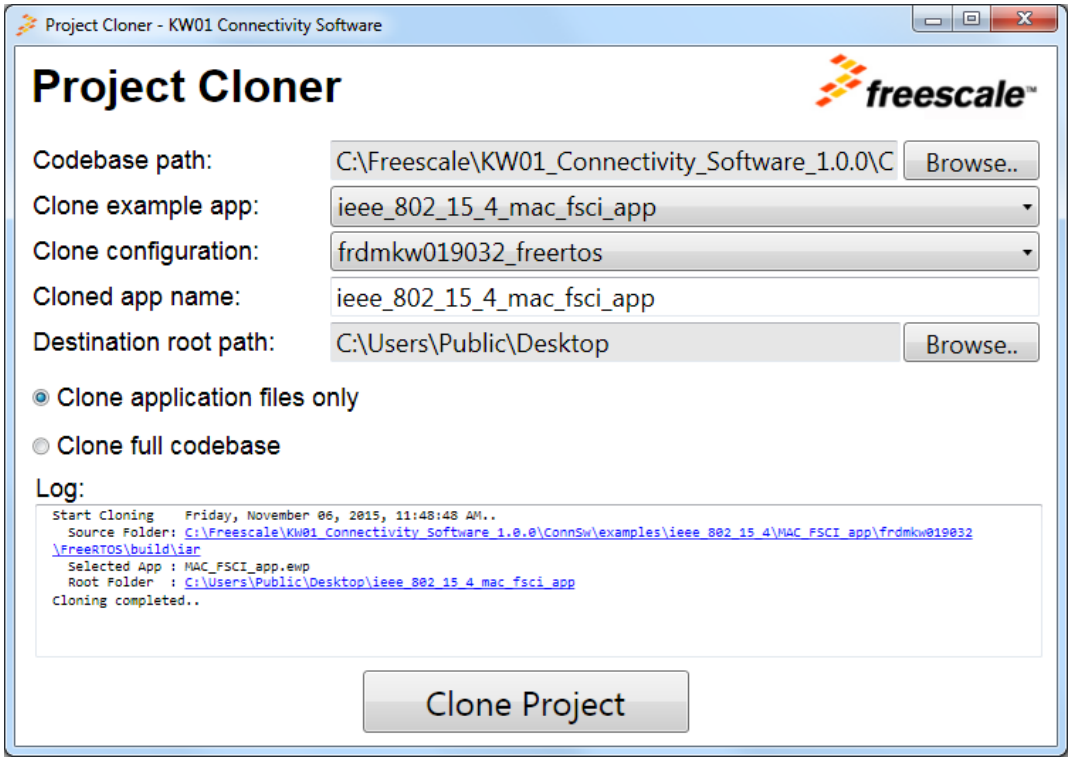
The default **Destination root path** for the cloned application is the Documents folder of the current user. To change this path click the **Browse** button to select a new location.

By default the Project Cloner will clone only application files (board specific files and example app files). To clone all the files, select the **Clone full codebase** radio button.

Now press the **Clone Project** button to start the cloning process. The log window will display “Cloning completed” when the process ends.

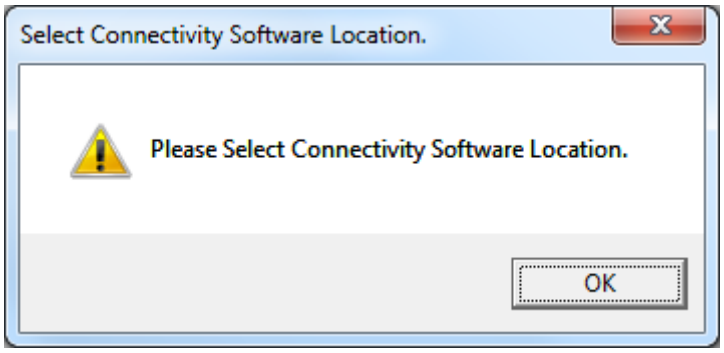


Available configurations for the selected example application



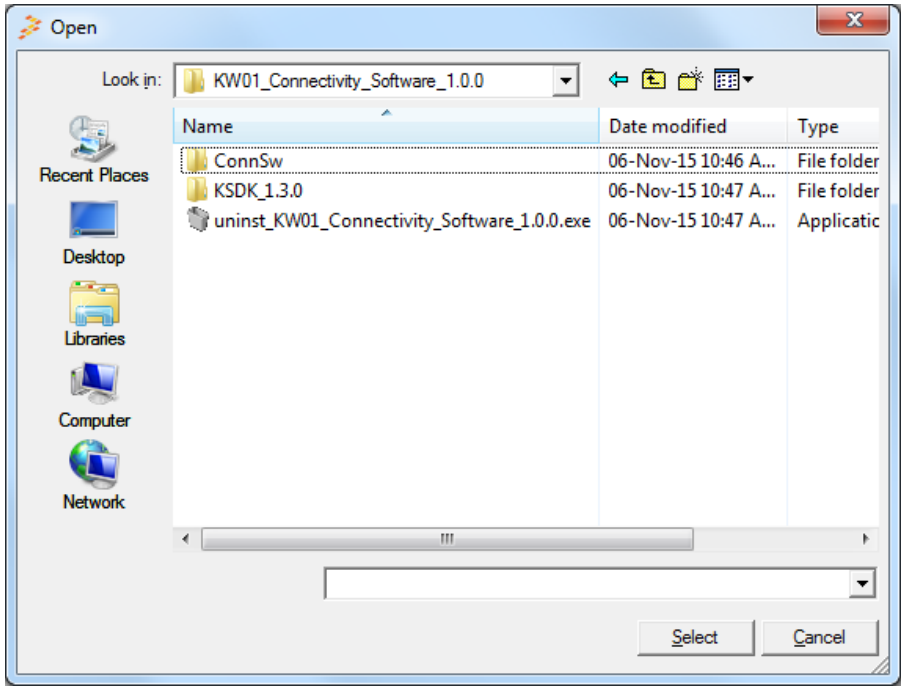
Application cloning done

If the project_cloner is executed from a different location than the installation path, the following notification will pop up.



Connectivity Software Location window when project_cloner is moved

Then you must specify the location of the KW01 Connectivity Software installation folder. This location can be modified at any time.



Connectivity Software location selection

3 Building the Binaries

This section details the required steps for obtaining the binary files for usage with the boards.

NOTE

In order to be able to build any of these packages you need a copy of the IAR Embedded Workbench for ARM® version 7.40.2 or higher. This connectivity software package does not include support for any other toolchains.

The packages must be built with the debug configuration in order to enable debugging information.

3.1 Building the KSDK Libraries

This release supports all development platforms based on the KW01 wireless microcontroller. The KSDK platform libraries are RTOS dependent, so appropriate libraries must be built for the RTOS chosen.

For any connectivity application, the following Kinetis SDK libraries must be built with the IAR Embedded Workbench for ARM® in order to enable the complete board support and RTOS kernel support:

- Platform drivers library
- Platform MQX library, for MQX connectivity demo applications

The location of the KSDK platform projects is described using the following placeholders for text:

- <KSDK13_FWK513_PATH> : represents the root path of the KSDK installation folder
- <device> : represents the board MCU: KW01Z4
- <board> : represents the board: frdmkw010932
- <RTOS>: represents the Real Time Operating System

Using the placeholders, these are the required Kinetis SDK v1.3.0 projects locations:

- <KSDK13_FWK513_PATH>\lib\ksdk_<RTOS>_lib\iar\<device>\ksdk_<RTOS>_lib.eww
- <KSDK13_FWK513_PATH>\lib\ksdk_platform_lib\iar\<device>\ksdk_platform_lib.eww

NOTE

The IAR projects for KSDK libraries are included in the IAR workspaces corresponding to the IEEE 802.15.4 demo applications and it is recommended to access them this way.

3.2 Building and Flashing the Freescale IEEE 802.15.4 Software Demo Applications

The package contains various demo applications that can be used to get a first feel for the software.

In this section you will be guided through building the ieee_802_15_4_mac_fsci_app. After you select the configuration you want to clone, locate the destination folder and follow the steps bellow.

Freescale IEEE 802.15.4 Software Demo Application Build Example

Selected app: ieee_802_15_4_mac_fsci_app

Board: frdmkw010932

RTOS: FreeRTOS

Destination root path: C:\Users\Public\Desktop

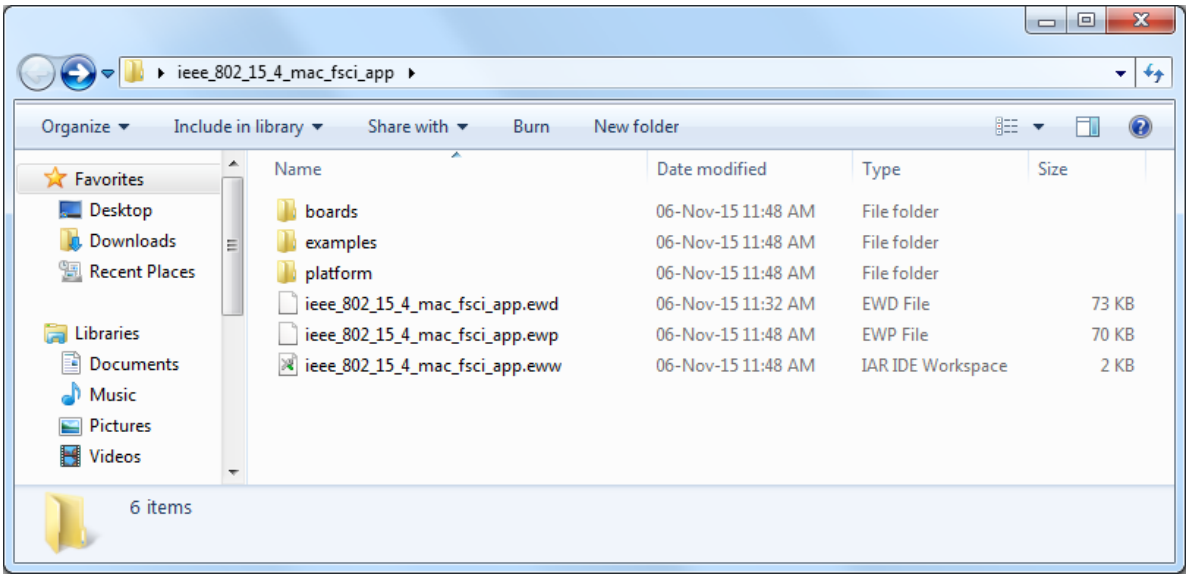
Resulting location: C:\Users\Public\Desktop\ieee_802_15_4_mac_fsci_app\

Step 1:

Navigate to the resulting location.

Step 2:

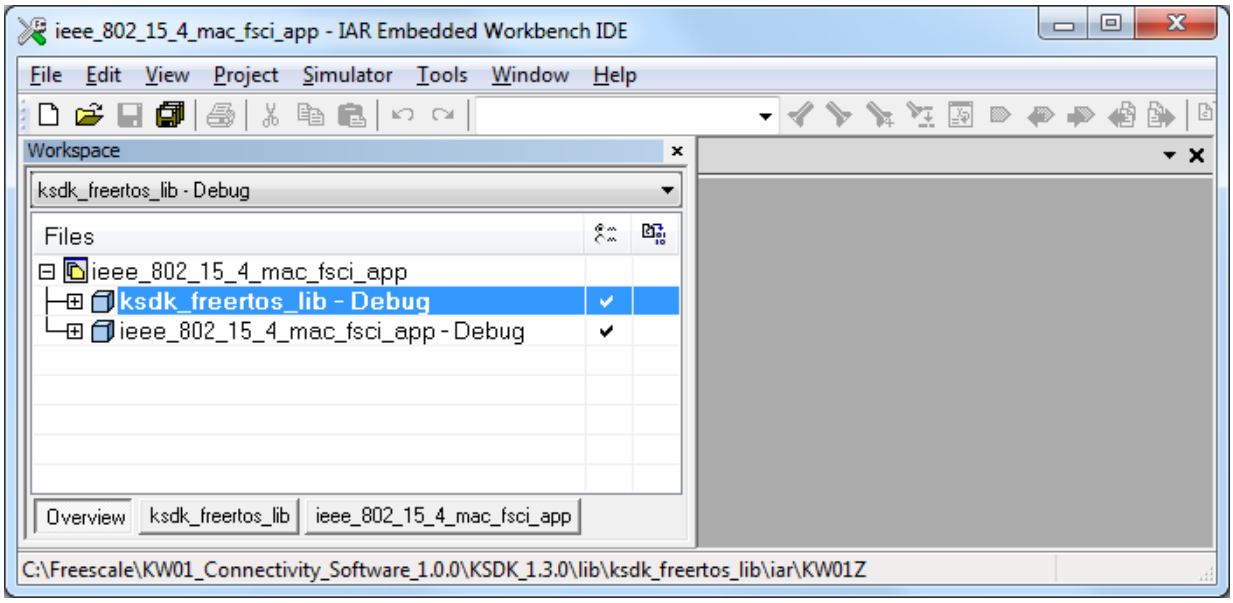
Open the highlighted IAR workspace file (*.eww file format):



ieee_802_15_4_mac_fsci_app demo project location

Step 3:

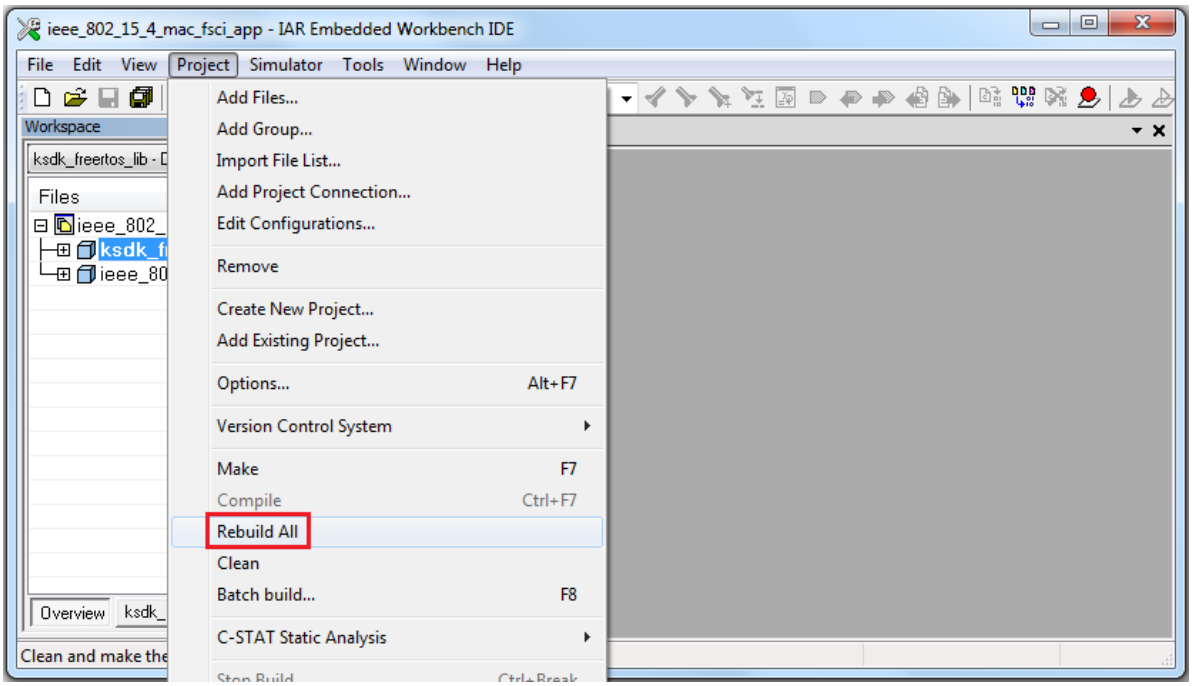
Select the KSDK FreeRTOS platform library project.



KSDK FreeRTOS platform library IAR project

Step 4:

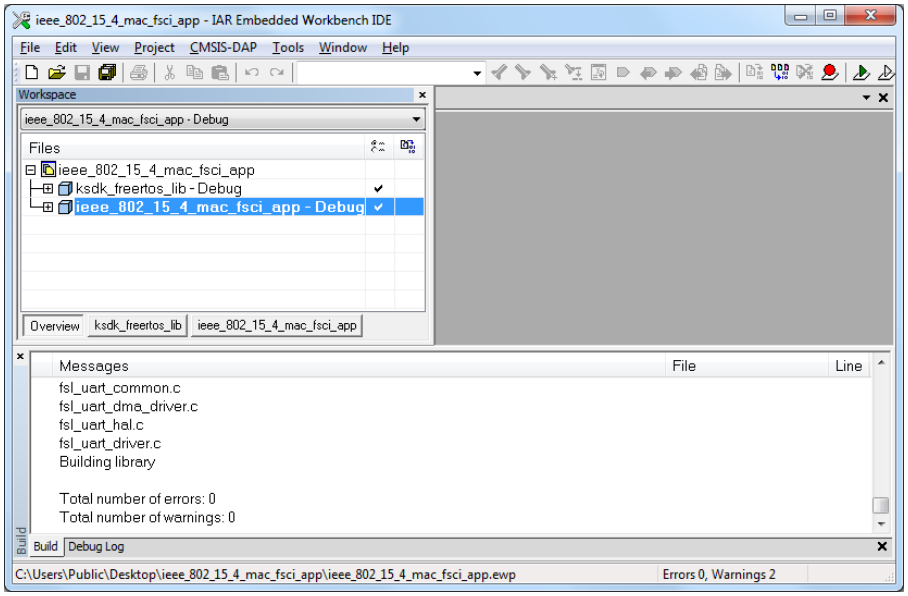
Build the KSDK FreeRTOS platform library project.



KSDK FreeRTOS platform library build

Step 5:

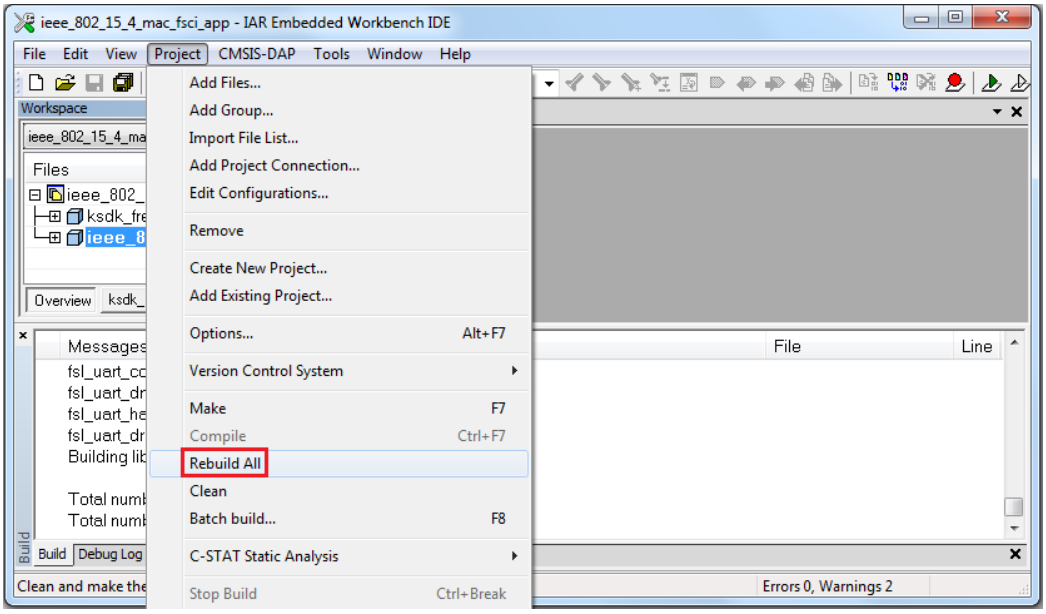
Select the ieee802_15_4_mac_fsci_app project.



ieee802_15_4_mac_fsci_app FreeRTOS IAR project

Step 6:

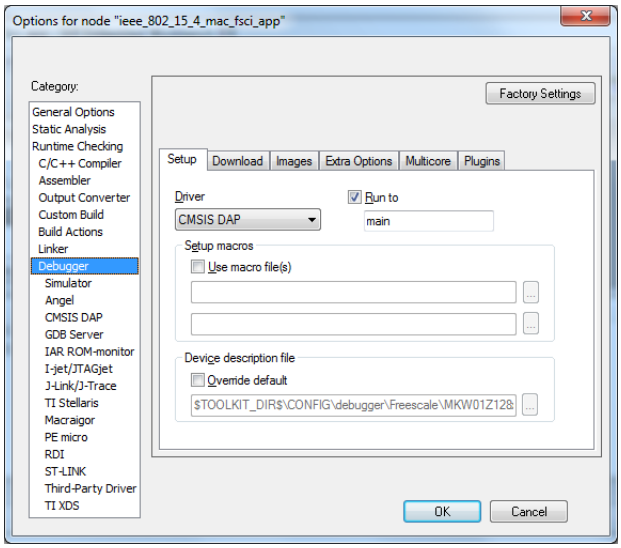
Build the ieee802_15_4_mac_fsci_app project.



ieee802_15_4_mac_fsci_app FreeRTOS build

Step 7:

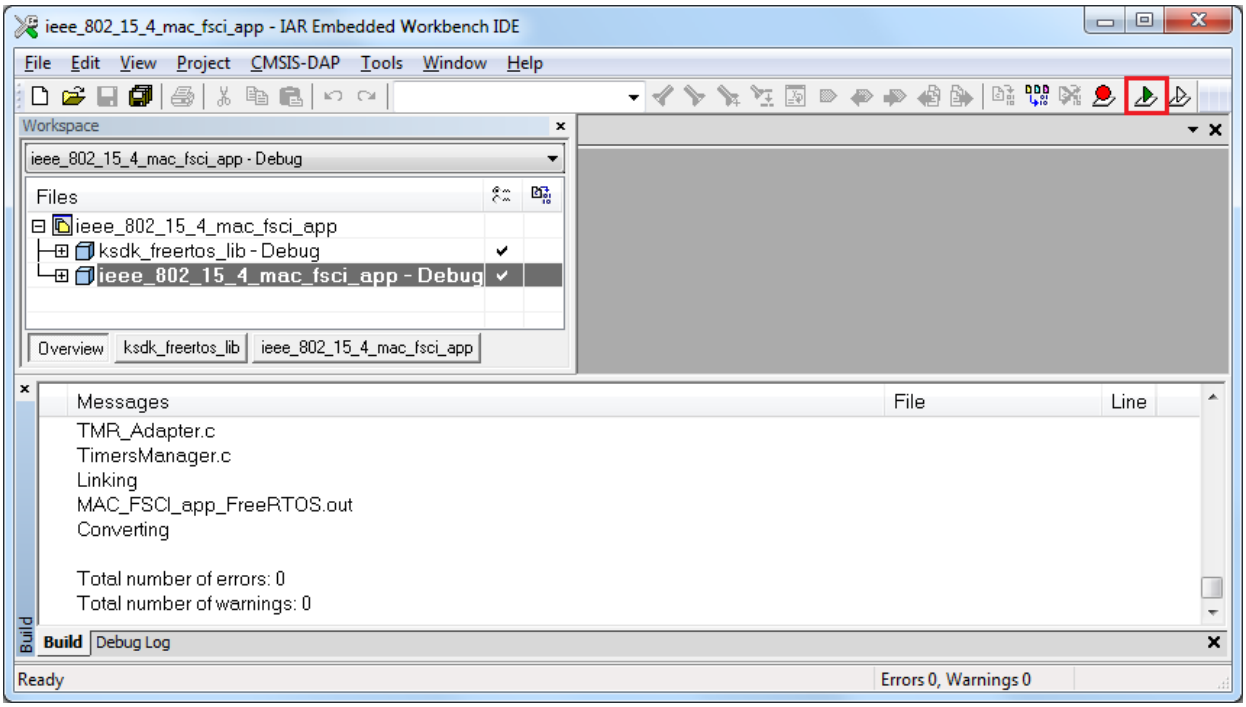
Check the appropriate debugger settings in the project options window:



Debugger Settings

Step 8:

Click the "Download and Debug" button to flash the executable onto the board.



ieee802_15_4_mac_fsci_app Download and Debug

NOTE

The projects are configured with “CMSIS-DAP” firmware as the default debug configuration. Please make sure that your board’s OpenSDA chip contains a CMSIS-DAP firmware or that the debugger selection corresponds to the physical interface used to interface to the boards. See the section below for more information.

3.3 Flashing a Binary Image File Without Using an IDE

The KW01 connectivity software package contains in the ConnSw\tools\binaries folder a series of pre-compiled binary applications that can be flashed onto a development board.

In order to flash the corresponding binaries to the FRDM-KW01 board, the best approach is to use the OpenSDA on-board interface CMSIS-DAP Mass Storage Device functionality, by simply dragging and dropping the binary image in the mass storage drive exposed by this OpenSDA firmware. For more information, see the CMSIS-DAP firmware github project: https://github.com/mbedmicro/CMSIS-DAP.

In order to flash the firmware on the USB-KW01, for either the KW01 or the K22F silicon on board, a J-Link probe is needed along with the latest J-Link software from www.segger.com.

Run the *jlink.exe* executable provided in the J-Link software installation and type the commands below for flashing the image on the microcontroller. Make sure that the binary file is in the same folder with the *jlink.exe* executable, or specify the absolute path to the file.

```
unlock kinetis
device mkw01z128xxx4
loadbin ieee_802_15_4_mac_fsci_app.bin 0
```

4 Hardware Setup

The hardware setup in this example uses a FRDM-KW01 development platform shown in the figure below:

FRDM-KW01

The FRDM-KW01 should have its OpenSDA (OpenLink) USB port connected via micro-USB cable to a Windows PC. The OpenSDA chip on the freedom platform should have appropriate firmware flashed, with debugging and virtual serial COM port capabilities. For more information on OpenSDA please refer to the following webpage: www.freescale.com/opensda.

Variants of embedded firmware for the OpenSDA chip can be downloaded from:

https://github.com/mbedmicro/CMSIS-DAP
https://www.segger.com/opensda.html
http://www.pemicro.com/opensda/

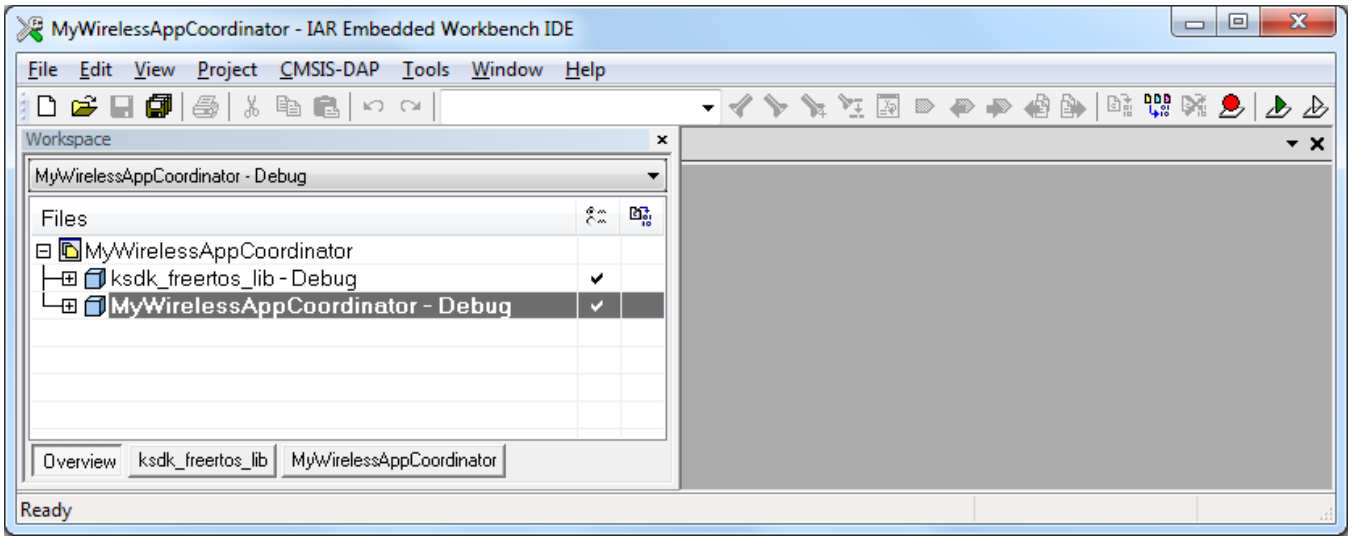
CMSIS-DAP is the default interface selected in the IAR Embedded Workbench for ARM® projects included in this release.

5 Example: Running the MyWirelessApp Demo Application

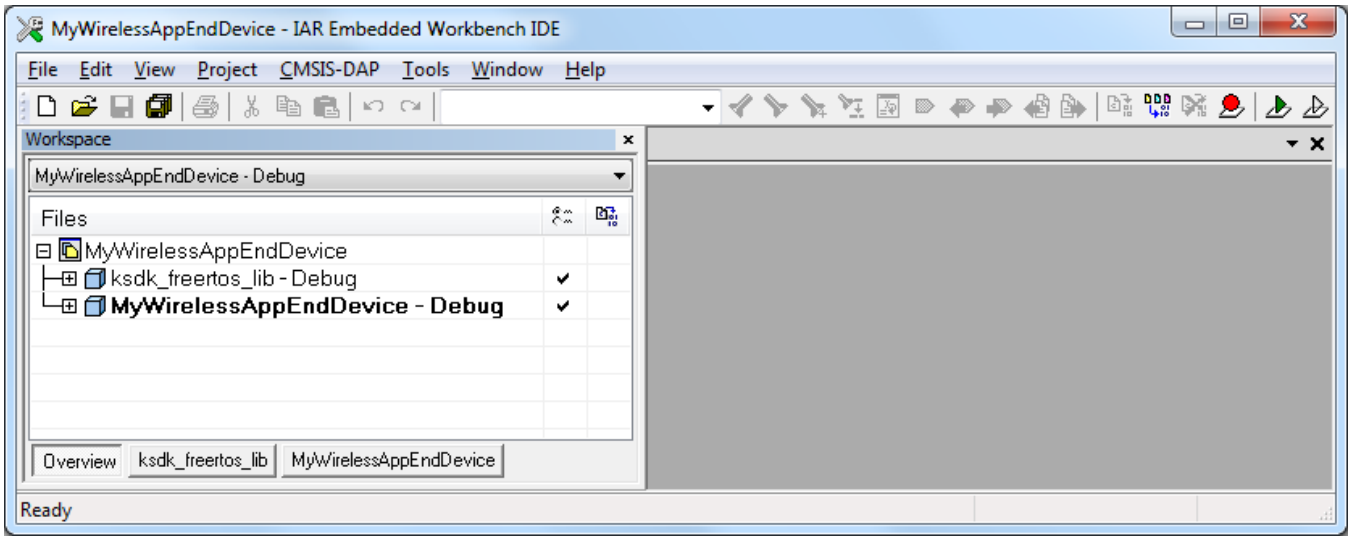
The MAC "MyWirelessApp" demo application requires a serial terminal program to connect to the boards. For this example, Tera Term was chosen.

Step 1:

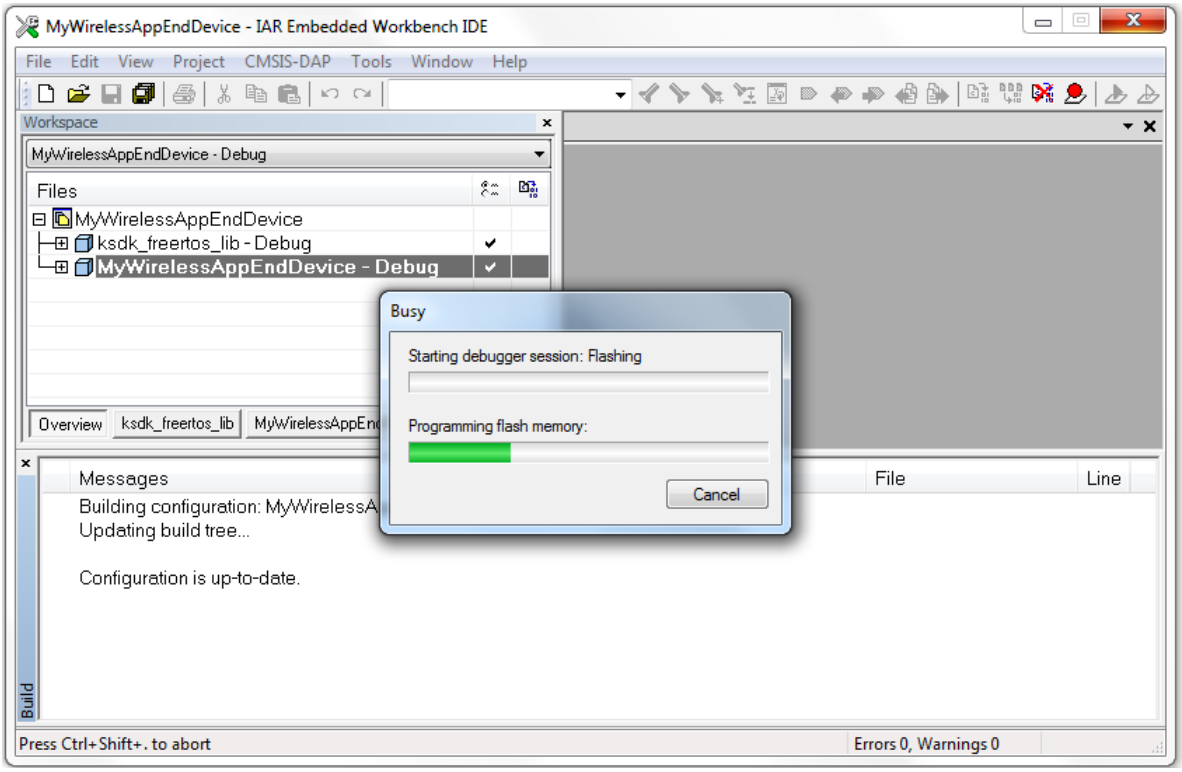
Load the applications on the boards using IAR Embedded Workbench for ARM®. This demo has two configurations: a "coordinator" and an "end device".



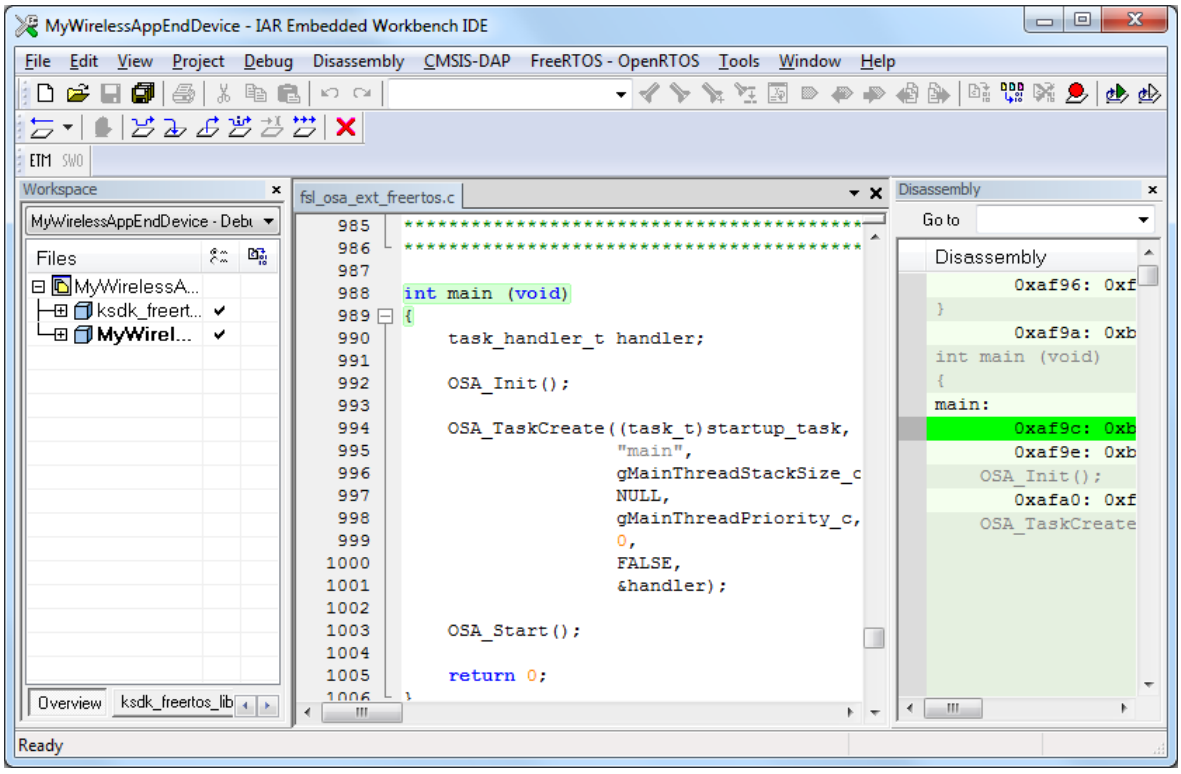
MyWirelessApp coordinator project



MyWirelessApp end device project



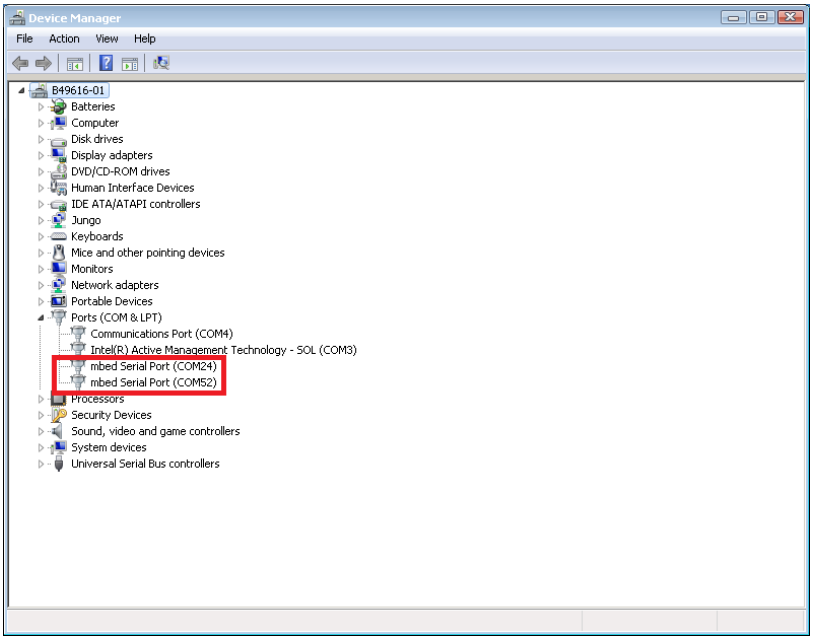
MyWirelessApp end device loading stage example



MyWirelessApp end device application loaded

Step 2:

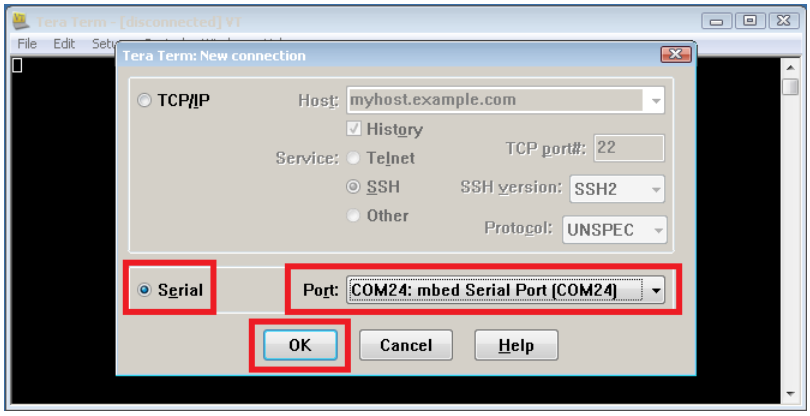
After loading both applications check “Device Manager” to get the serial ports numbers. These should appear with the prefix “mbed Serial Port”.



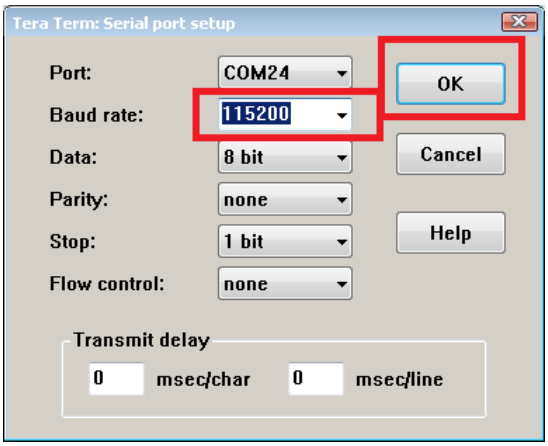
Device Manager serial port look up

Step 3:

Using the port numbers specified in Device Manager, open two Tera Term instances and connect to the devices using the 115200 baud rate. To change the baud rate of the terminal go to “Setup-> Serial Port” menu.



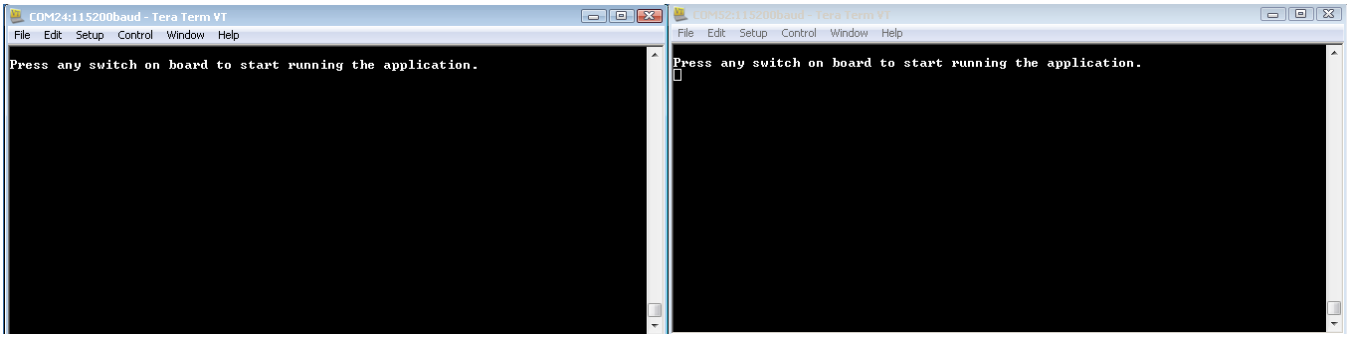
Select mbed serial connection COM port



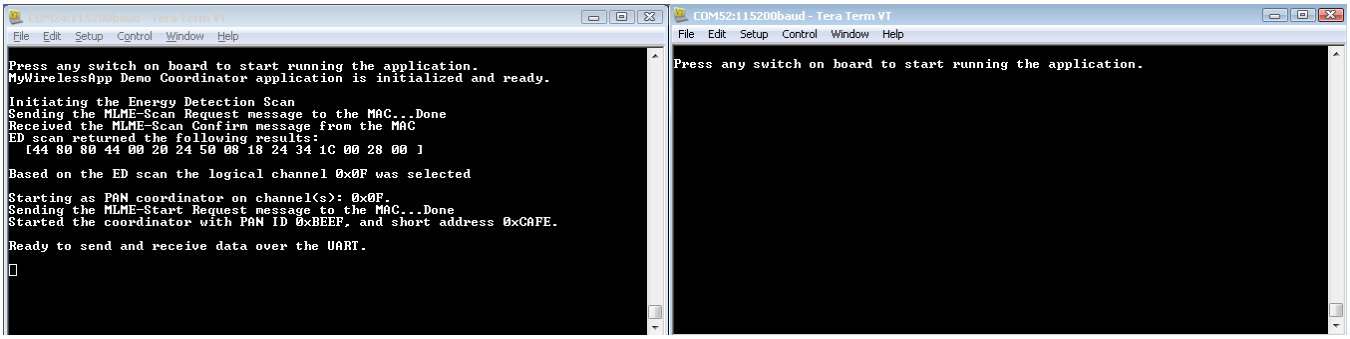
Setting correct baud rate

Step 4:

Start the applications by pressing any available key on the FRDM boards: first the coordinator and then the end device.

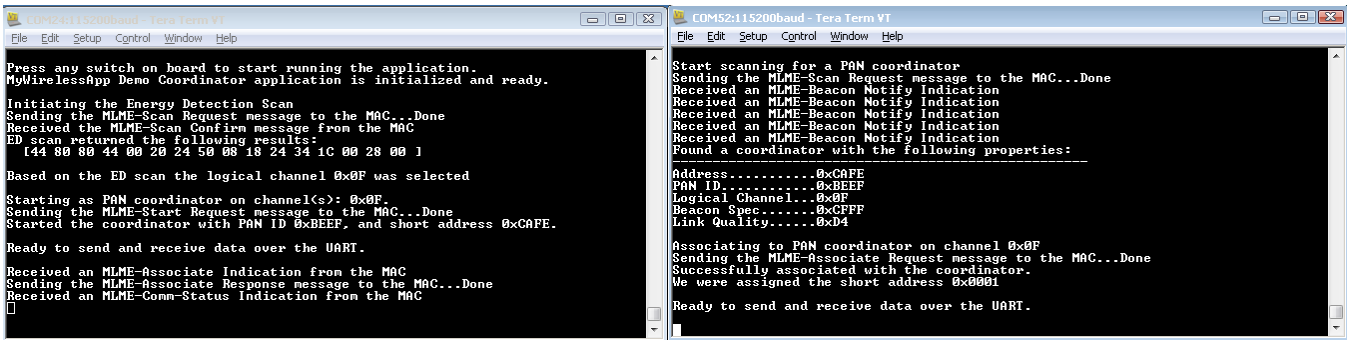


Both applications after reset



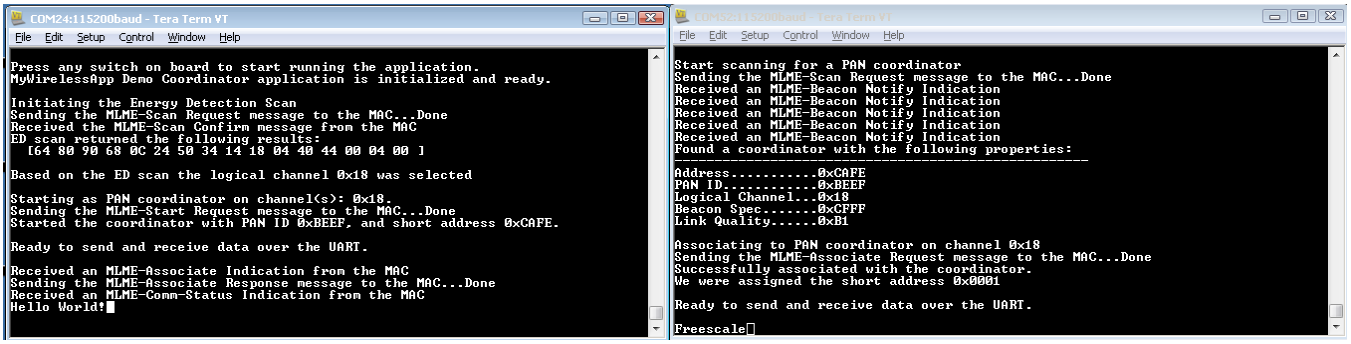
MyWirelessApp coordinator started

The coordinator performs an energy detect scan to determine the least occupied channel and then selects it to start the network. The end device performs an active scan and after receiving a beacon from a coordinator, it issues an association request.



MyWirelessApp end device associated to coordinator

After the association procedure completes, in the two terminal windows messages can be written to be exchanged by the two wireless nodes.



MyWirelessApp message exchange

The previous section demonstrates the basic steps to run a demo application. For detailed information about the demo applications, please refer the Demo Applications User's Guide included in the installer (*802154MPDAUG.pdf*).

Home Page:
freescale.com

Web Support:
freescale.com/support4

Document number: MKW40Z802154SW523QSG
Rev. 0
10/2015