# Kinetis MKW41Z IEEE® 802.15.4 Software

## Quick Start Guide

This document is a brief presentation of the Kinetis IEEE® 802.15.4 MAC/PHY Software for the MKW41Z wireless microcontrollers platforms, version 5.3.3. This software package is built using the Kinetis Software Development Kit (KSDK) version 2.2. This document covers installation of the software packages, hardware setup, build and usage of the provided demo applications.

**Contents**

# 1 Download

Navigate to https://mcuxpresso.nxp.com/en/welcome, and create a new configuration:

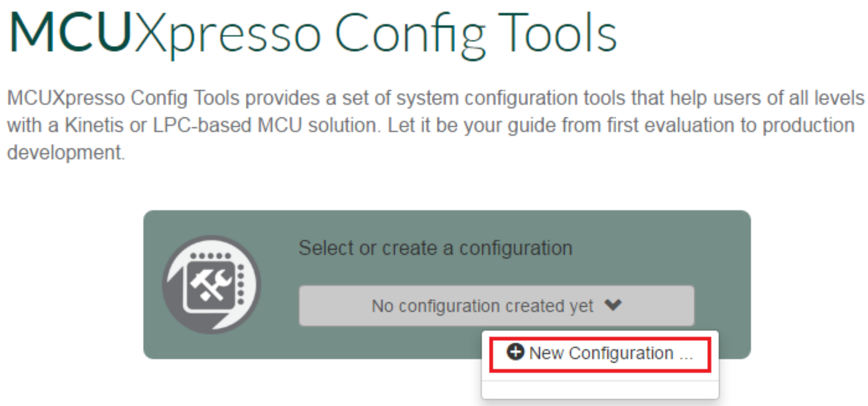


**Figure 1: The MCUXpresso welcome screen**

Type the desired platform or select from the list, then press “Specify Additional Configuration Settings”:

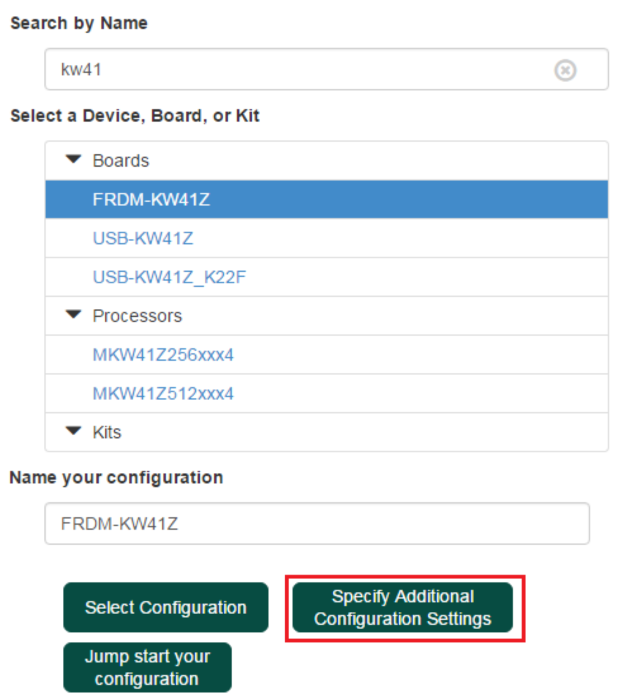


**Figure 2: Available configurations**

Select the desired toolchain(s) and middleware, then press the "Go to SDK Builder" button:

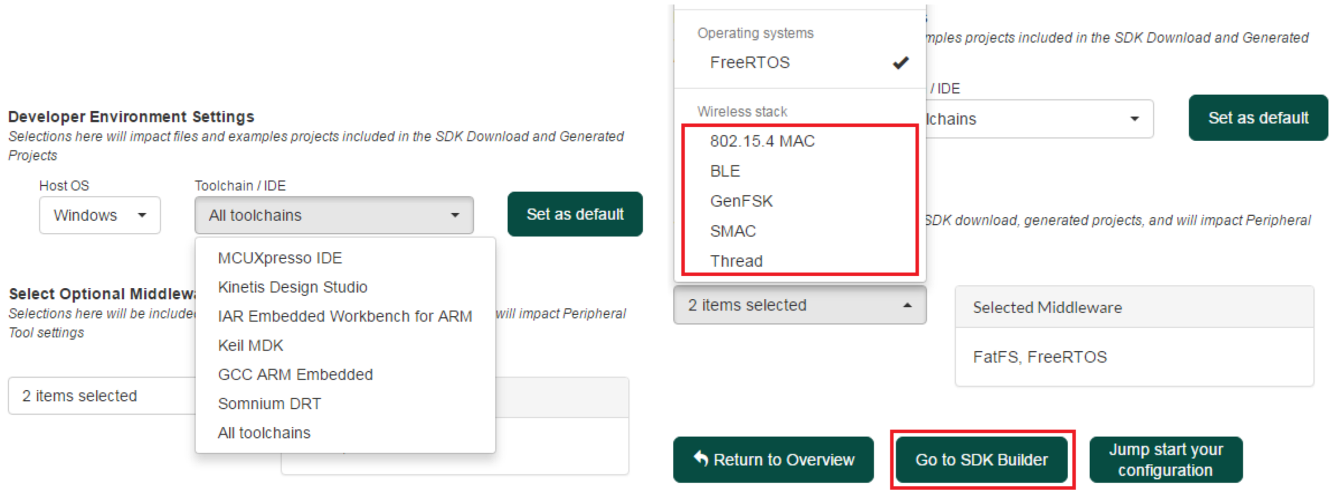


**Figure 3: Select toolchain(s) and middleware**

If the package is already built, the "Download" button will appear, else the "Request Build" button will appear:

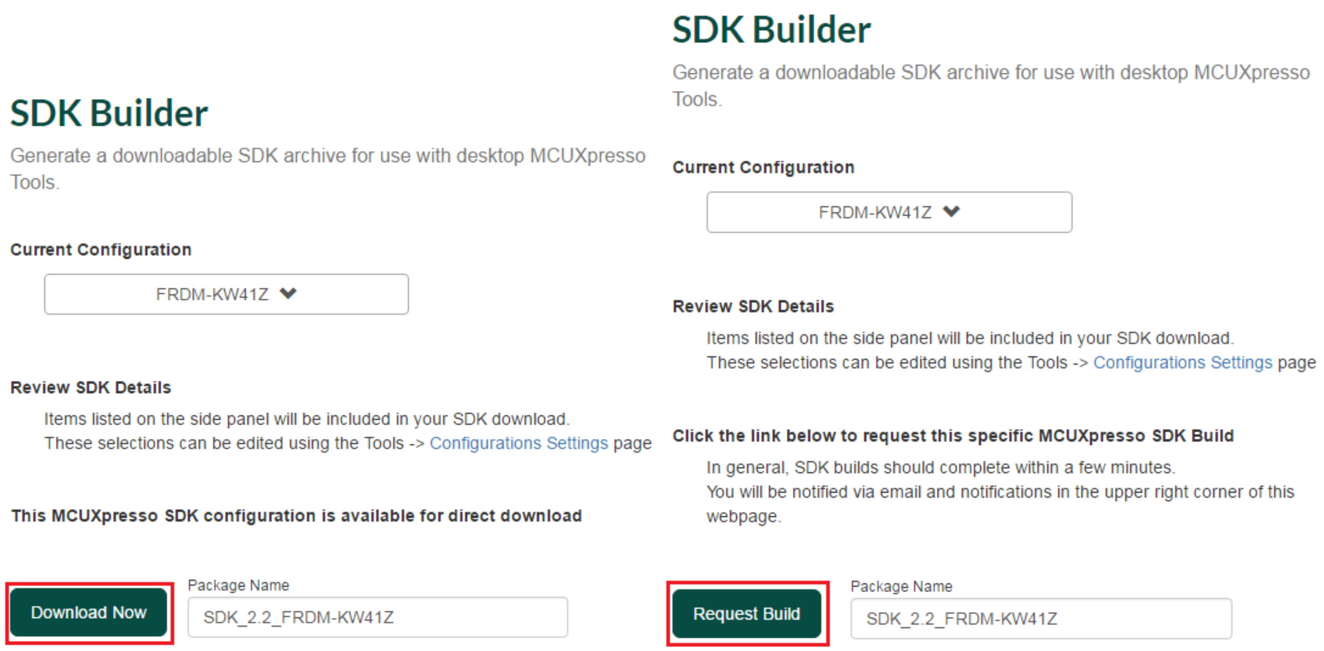


**Figure 4: The SDK Builder**

In one requests a new build, then a notification will be sent when process ends. To download the archive, access the SDK Archive manager:

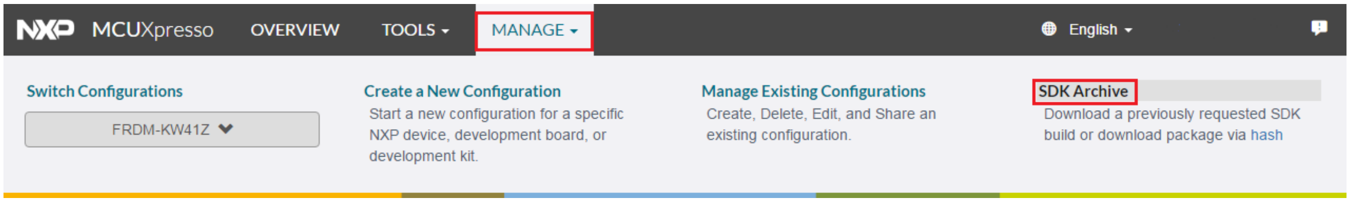


**Figure 5: Access the Archive manager**

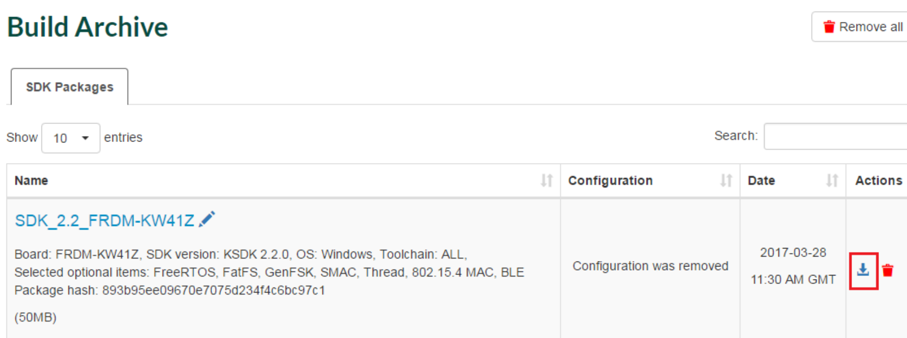


**Figure 6: Download the archive**

# 2 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 7.80.4 or higher or MCUXpresso Integrated Development Environment version 10.0 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.

This package includes various demo applications that can be used as a starting point.

The next section presents the steps required for building the *mac_fsci_black_box*. All applications can be found using the following placeholders for text:

- <connectivity_path> : represents the root path of the cloned application, or the root path for the IEEE 802.15.4 software package.
- <board> : represents the target board for the demo app, can be either “frdmkw41z” or “usbkw41z_kw41z”
- <RTOS>: represents the scheduler or RTOS used by the app, can be “bm” or “freertos”
- <demo_app> : represents the demo app name
- <IDE> : represents the integrated development environment used to build projects and can be “iar” or in the case of MCUXpresso IDE it can be ignored.

The demo applications general folder structure is the following:

<connectivity_path>\boards\<board>\wireless_examples\ieee_802_15_4\<demo_app>\<RTOS>\<IDE>\

**Kinetis IEEE 802.15.4 Software Demo Application Build Example**

Selected app: mac_fsci_black_box

Board: frdmkw41z

RTOS: FreeRTOS

Resulting location:

<connectivity_path>\boards\frdmkw41z\wireless_examples\ieee_802_15_4\mac_fsci_black_box\FreeRTOS\ <IDE>

### NOTE

If your FRDM-KW41Z board is configured for the buck or boost modes of the DCDC converter inside the KW41Z microcontroller, please note that the following defines need to be set: *gDCDC_Enabled_d* to 1 and *APP_DCDC_MODE* to *gDCDC_Mode_Buck_c* or *gDCDC_Mode_Boost_c* respectively, in the *app_preinclude.h* header file.

## 2.1 Building and Flashing the Kinetis IEEE 802.15.4 Software Demo Applications using MCUXpresso IDE

### Step 1:

Drag and drop the archive into the MCUXpresso Installed SDKs window.

### Step 2:

Import the SDK examples into Workspace.

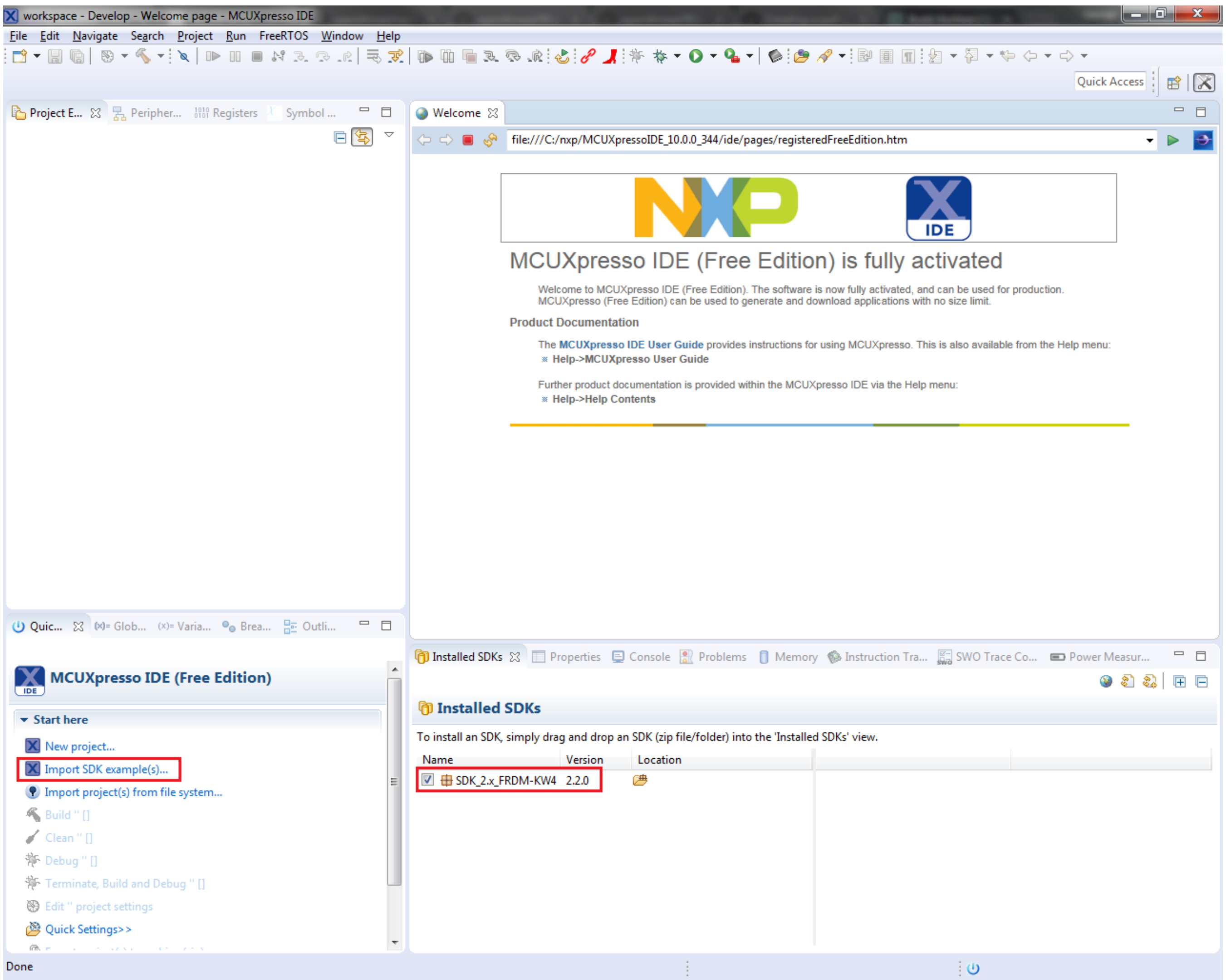


**Figure 7: Installed SDKs**

## Step 3:

Select the board, then the desired example(s):

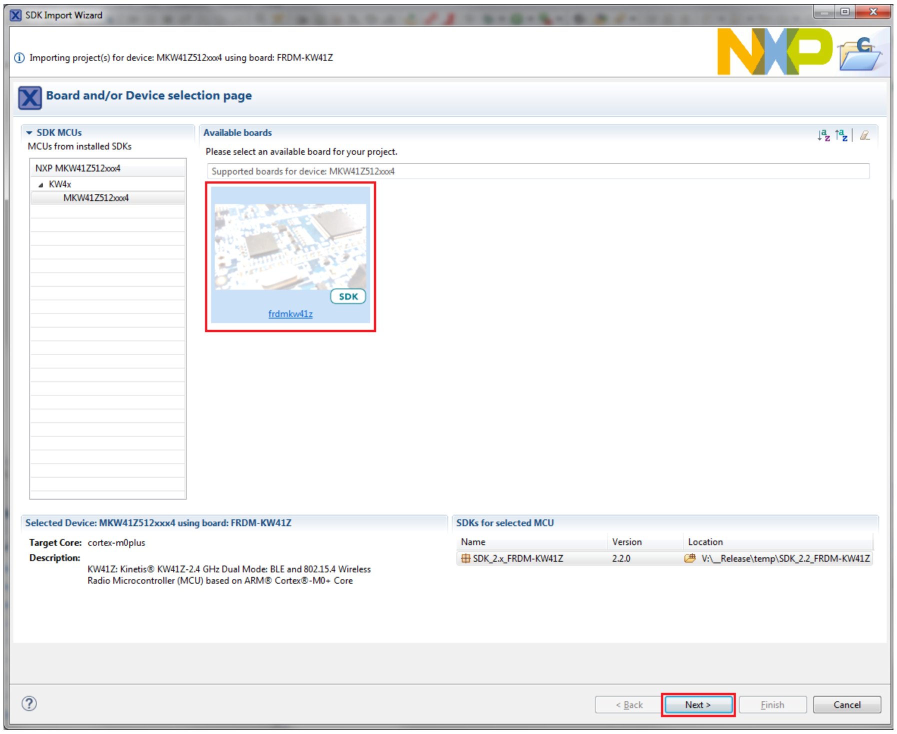


**Figure 8: Select the board**

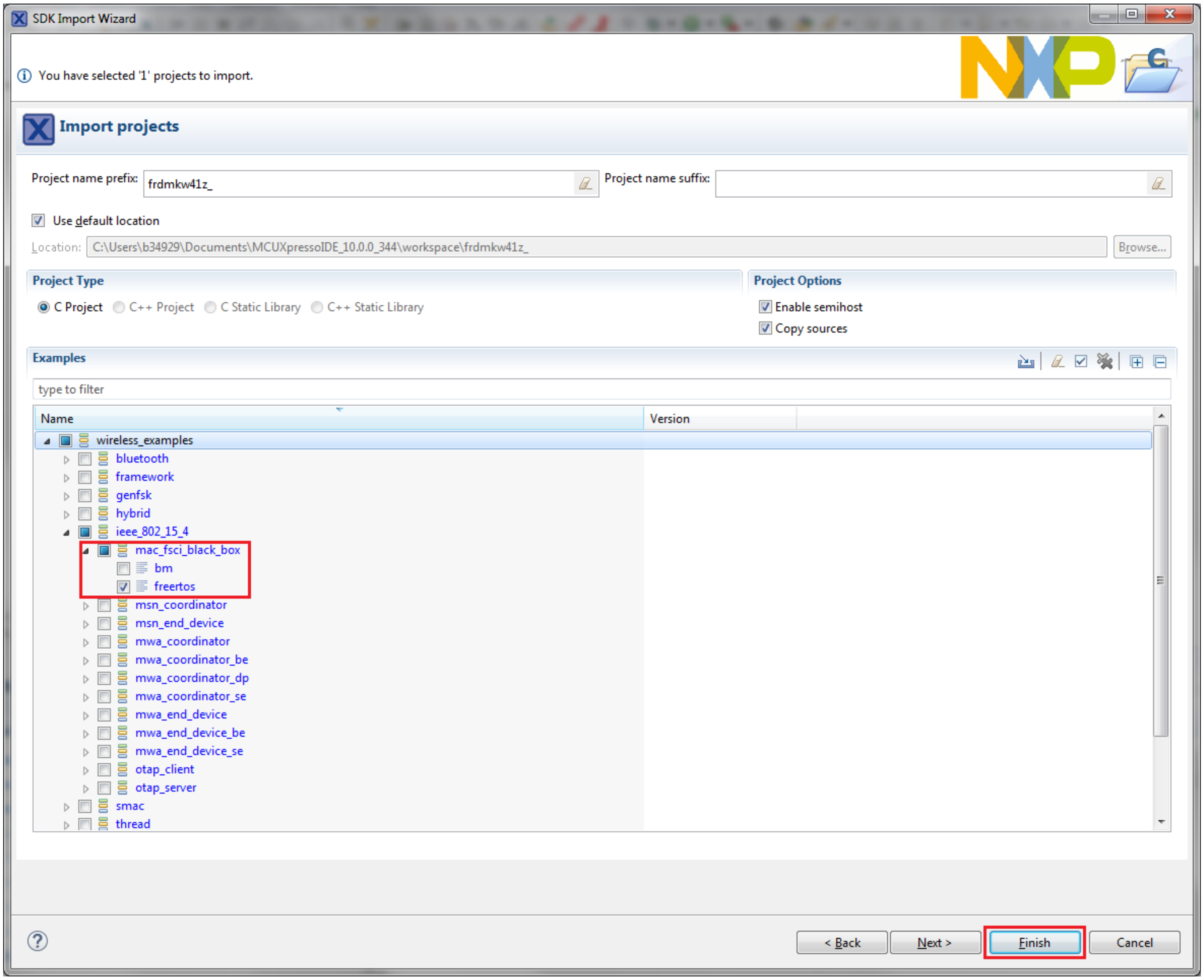


**Figure 9: Select the example(s)**

## Step 4:

Build the mac_fsci_black_box project.

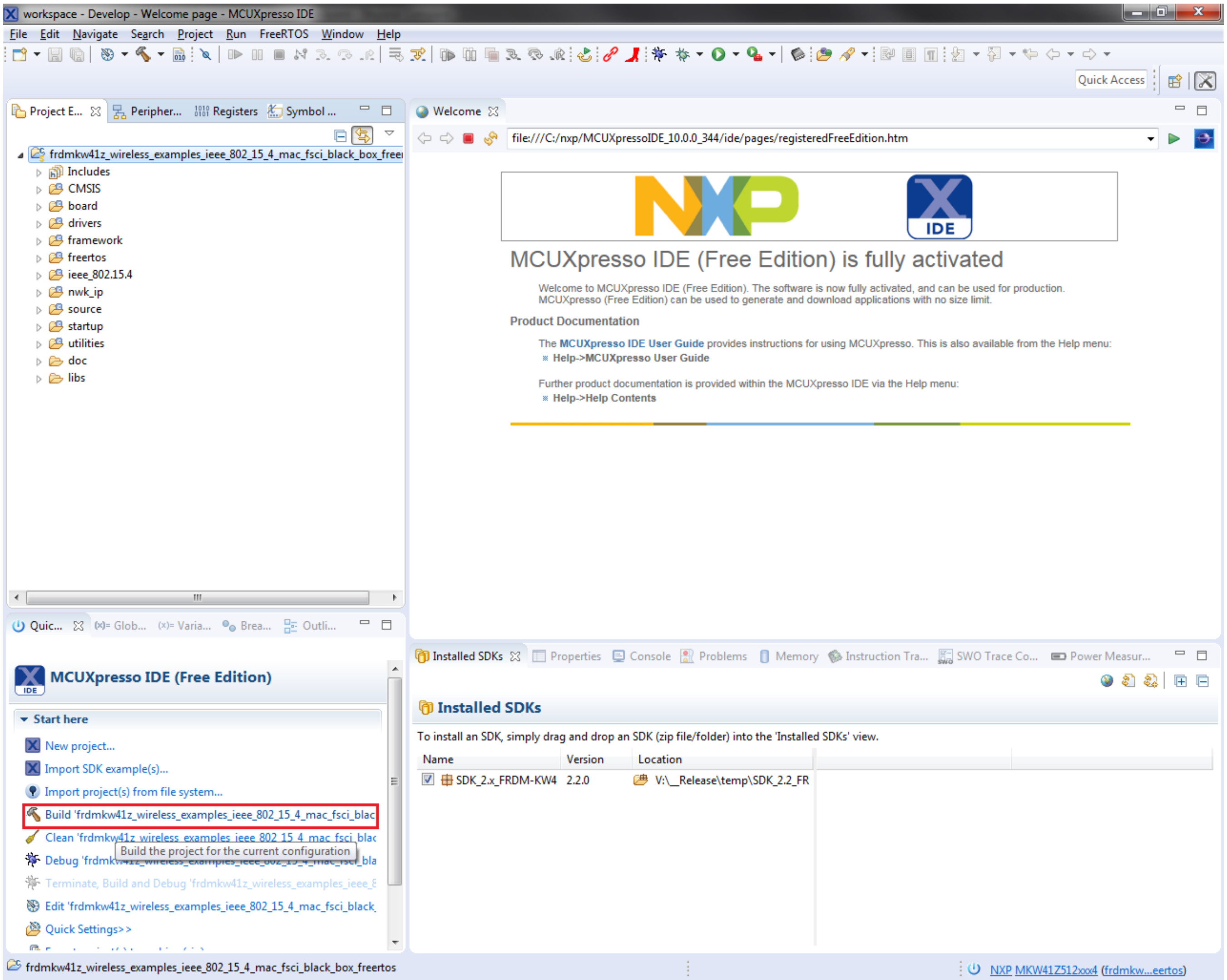


**Figure 10: "mac_fsci_black_box" FreeRTOS build**

Step 5:

Click the “Debug” button to flash the executable onto the board.

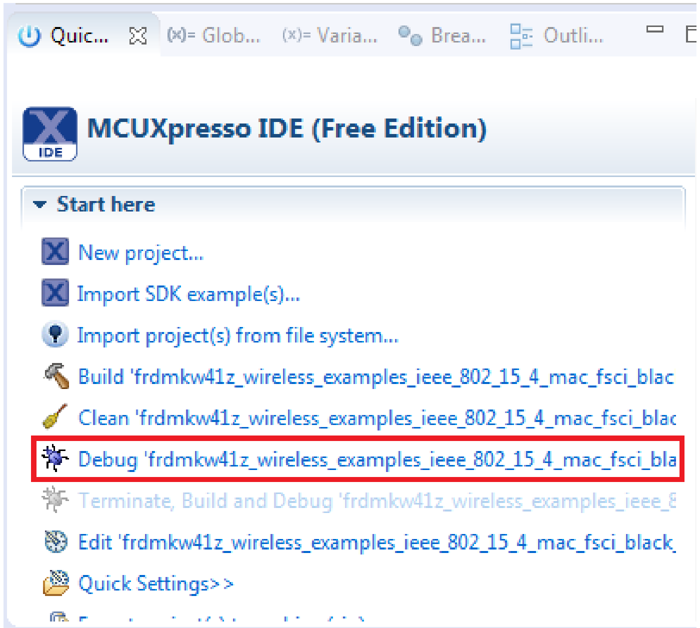


**Figure 11: “mac_fsci_black_box” Debug**

## 2.2 Building and Flashing the Kinetis IEEE 802.15.4 Software Demo Applications using IAR

Step 1:

Navigate to the resulting location in either the connectivity software installation directory or the cloned application root directory.

Step 2:

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

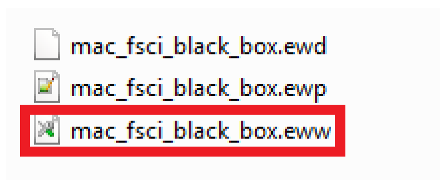


**Figure 12: "mac_fsci_black_box" demo project location**

Step 3:

Select the mac_fsci_black_box project.

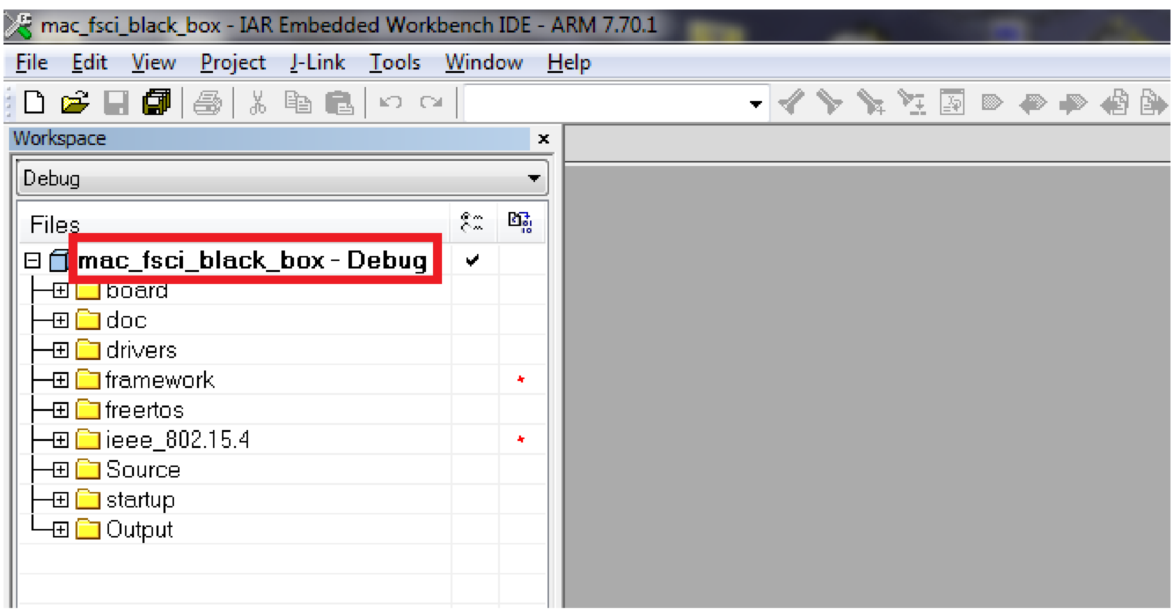


**Figure 13: mac_fsci_black_box FreeRTOS IAR project**

Step 4:

Build the mac_fsci_black_box project.

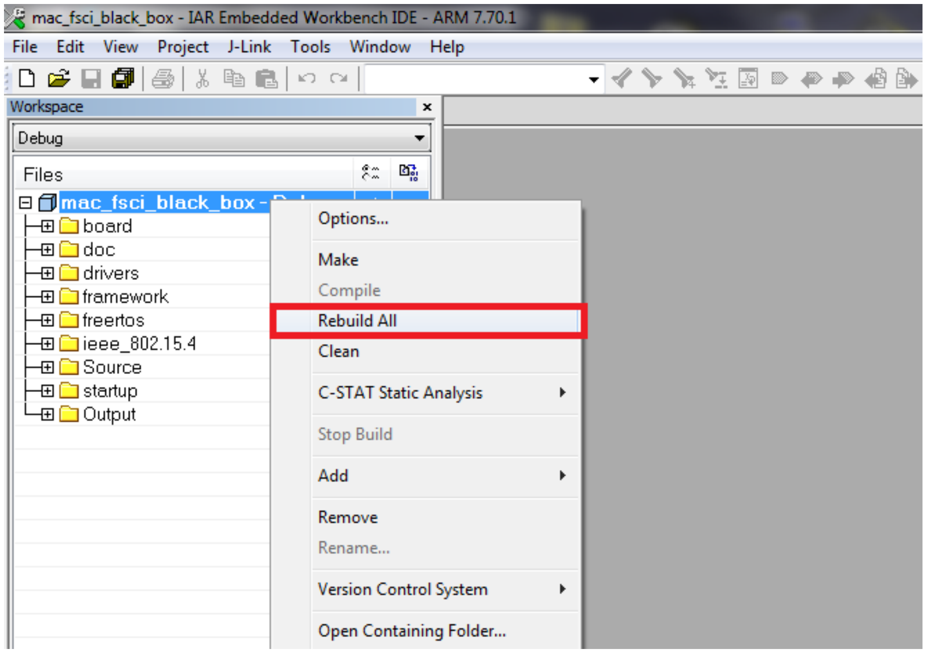


**Figure 14: mac_fsci_black_box build**

Step 5

Make the appropriate debugger settings in the project options window:

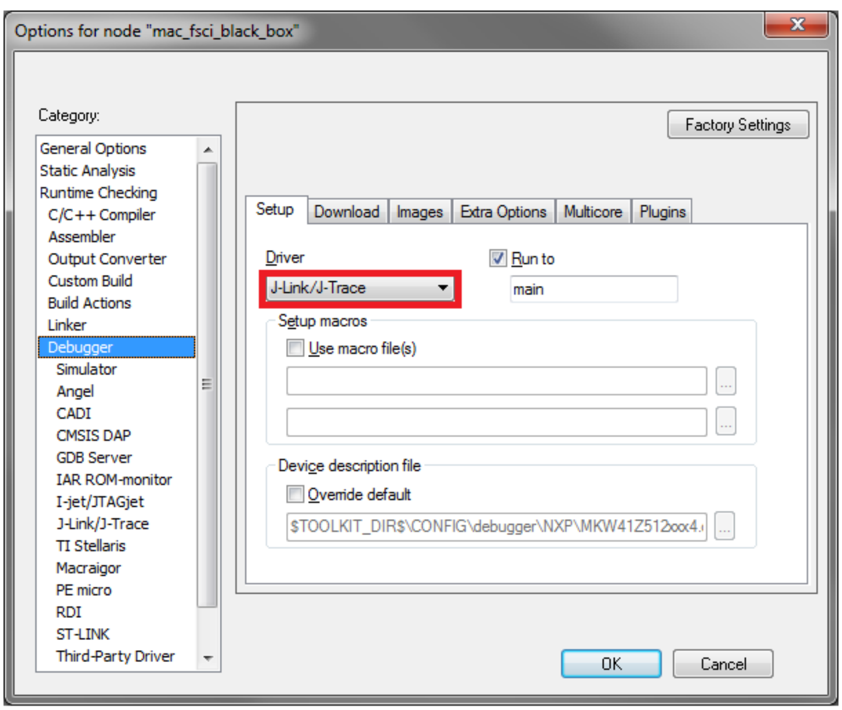


**Figure 15: Debugger Settings**

Step 6:

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

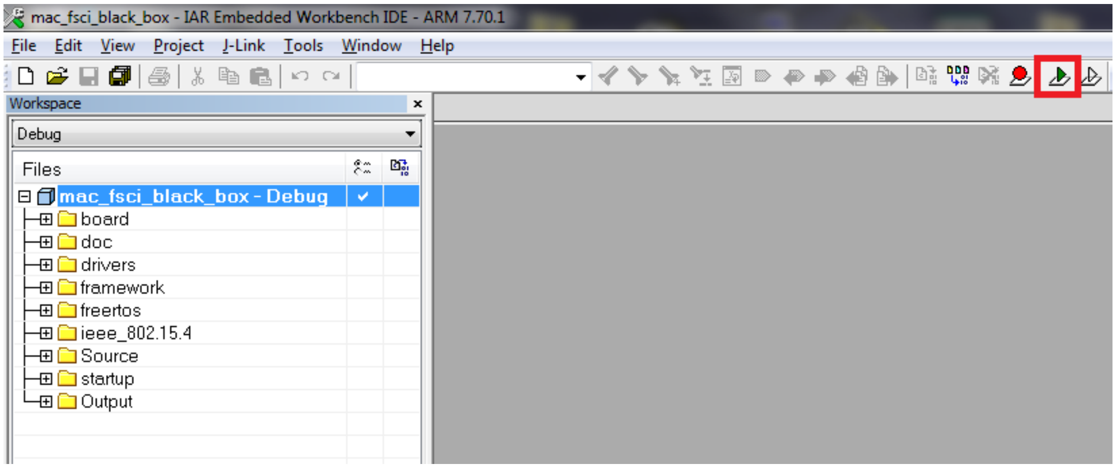


Figure 16: **mac_fsci_black_box Download and Debug**

NOTE

The projects are configured to use “J-Link / J-Trace” as the default debugger. Please make sure that your board’s OpenSDA chip contains a J-Link firmware or that the debugger selection corresponds to the physical interface used to interface to the board. See the section below for more information.

## 2.3 Flashing a Binary Image File Without Using an IDE

The MKW41Z connectivity software package contains in the *<connectivity_path>\tools\wireless\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 FRDMKW41Z board, the best approach is to use the OpenSDA on-board interface J-Link Mass Storage Device functionality, by simply dragging and dropping the binary image in the mass storage drive exposed by this OpenSDA firmware.

In order to flash the firmware on the USBKW41Z, 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 follow the steps 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.

Step 1: Select MKW41Z512xxx4 device.

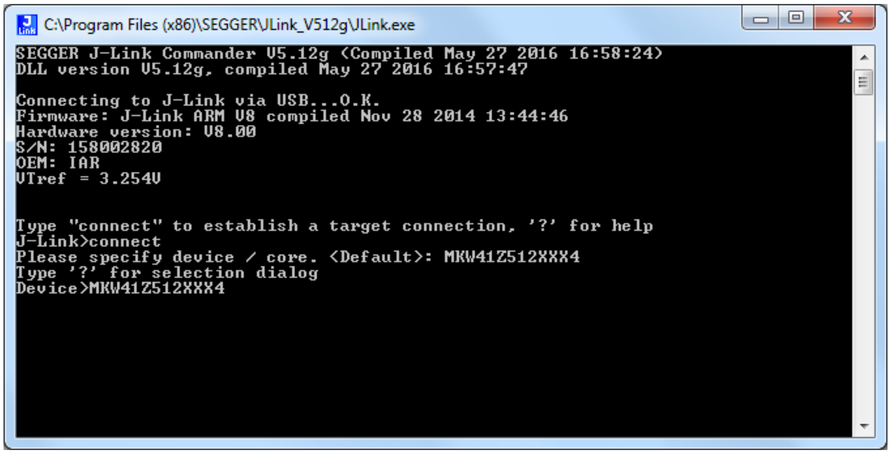


**Figure 17: MKW41Z512xxx4 device selection**

Step 2: Select SWD target interface.

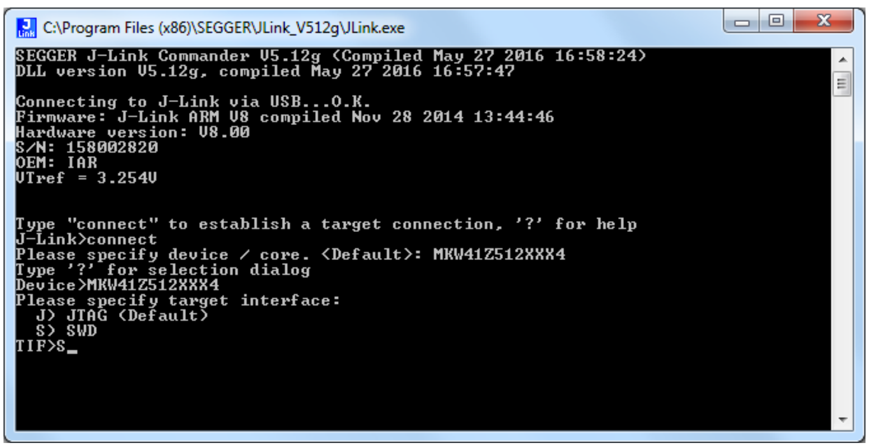


**Figure 18: SWD interface selection**

Step 3: Press “Enter” to select the default interface speed.

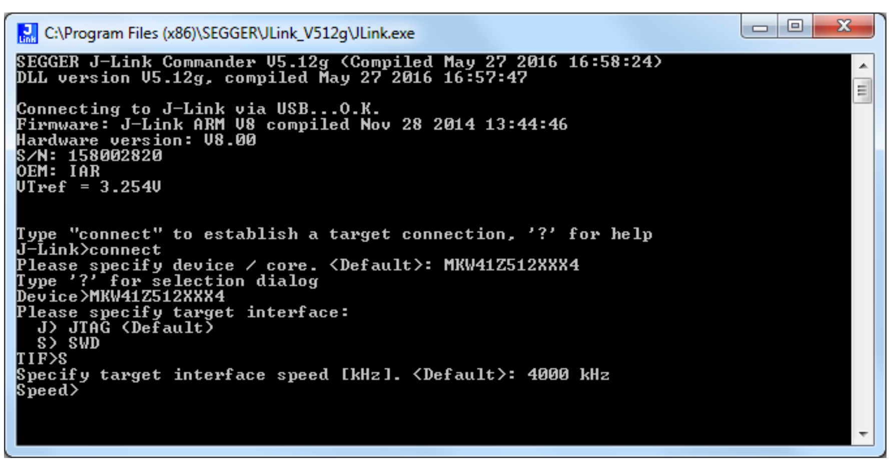


**Figure 19: SWD interface speed selection**

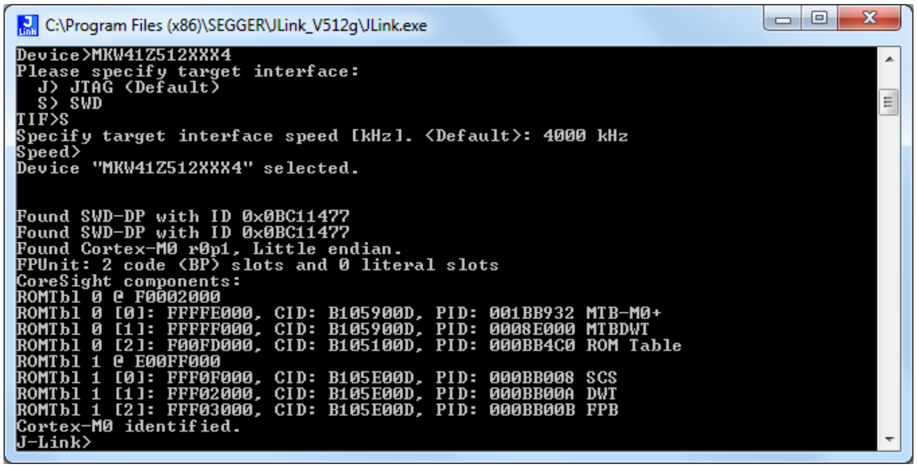


**Figure 20: Cortex-M0 identified**

Step 4: Type `loadbin app.bin 0` in order to flash the binary file (assuming application name is *app.bin*).

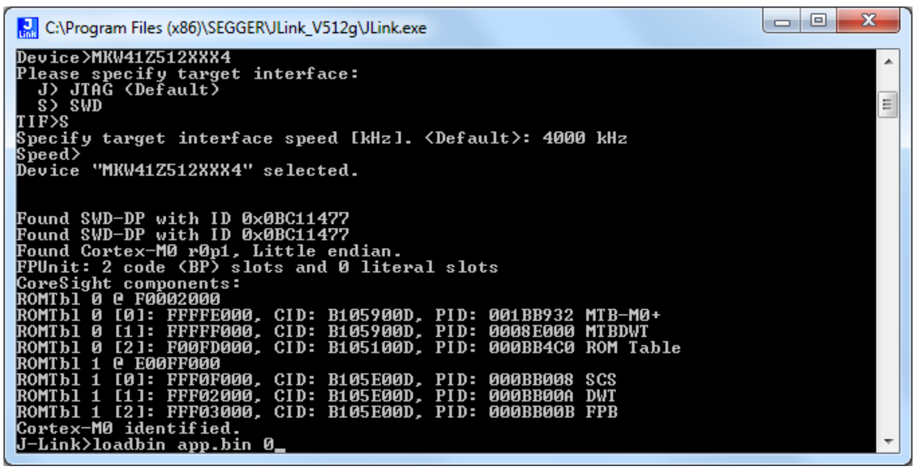


**Figure 21: Load binary file**

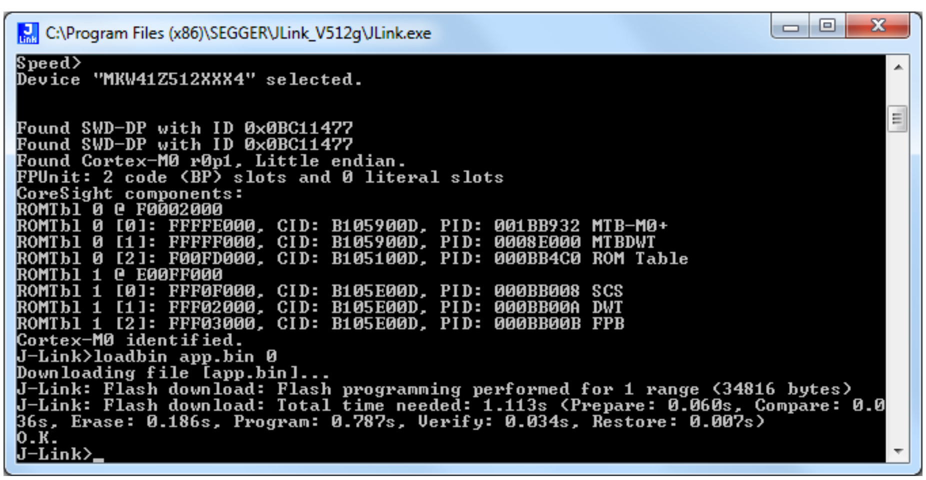


**Figure 22: Download completed successfully**

# 3 Hardware Setup

The hardware setup in this example uses either a FRDMKW41Z or USBKW41Z development platform, shown in the figure below:

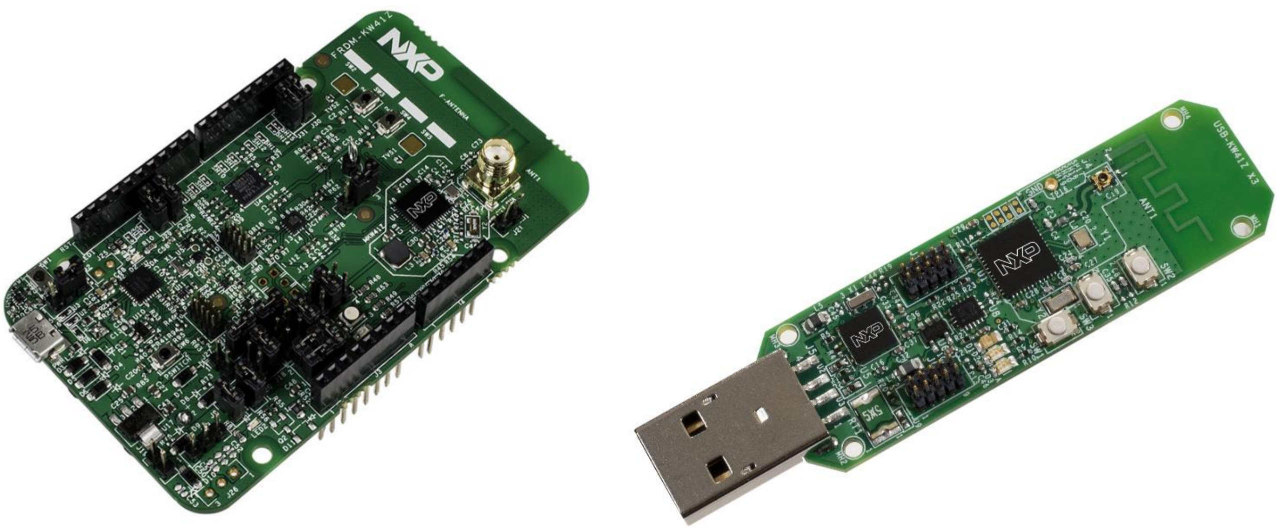

**Figure 23: FRDMKW41Z and USBKW41Z**

The FRDMKW41Z and USBKW41Z boards should have their OpenSDA USB ports connected to a Windows PC. The OpenSDA chip on the boards 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.nxp.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/

J-LINK/J-TRACE is the default interface selected in the IAR Embedded Workbench for ARM® projects with FRDMKW41Z and USBKW41Z included in this release.

The FRDM-KW41Z board can be configured via jumpers to be in the two modes of the DCDC converter inside the KW41Z microcontroller or to bypass it entirely, as shown in the figure below:

**Power Configuration**

| | PWR_CFG J18 | PSW_CFG J16 | DCDC_CFG J17 | REG_CFG J22 |
|---|---|---|---|---|
| **Bypass Mode (auto start)**<br>VDCDC_IN (1.71 to 3.6V)<br>Operation 1.8V - 3.6 V | 1-2 | 1-2 | 3-4 | 1-3<br>2-4<br>5-6 |
| **Boost Mode (auto start)**<br>VDCDC_IN (0.9V to 1.8V)<br>Single Battery Operation | 2-4 | 3-4 | 1-2<br>5-6 | 5-6 |
| **Buck Mode (manual start)**<br>VDCDC_IN (1.8V to 4.2V)<br>Coin Cell Battery Operation | 2-4 | 5-6<br>press SW6 to start | 3-4 | 5-6 |
| **Buck Mode (auto start)**<br>VDCDC_IN (1.8V to 4.2V)<br>Coin Cell Battery Operation | 2-4 | 3-4 | 3-4 | 5-6 |

**Figure 24: FRDM-KW41Z Jumper Configuration for DCDC Modes**

# 4 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".

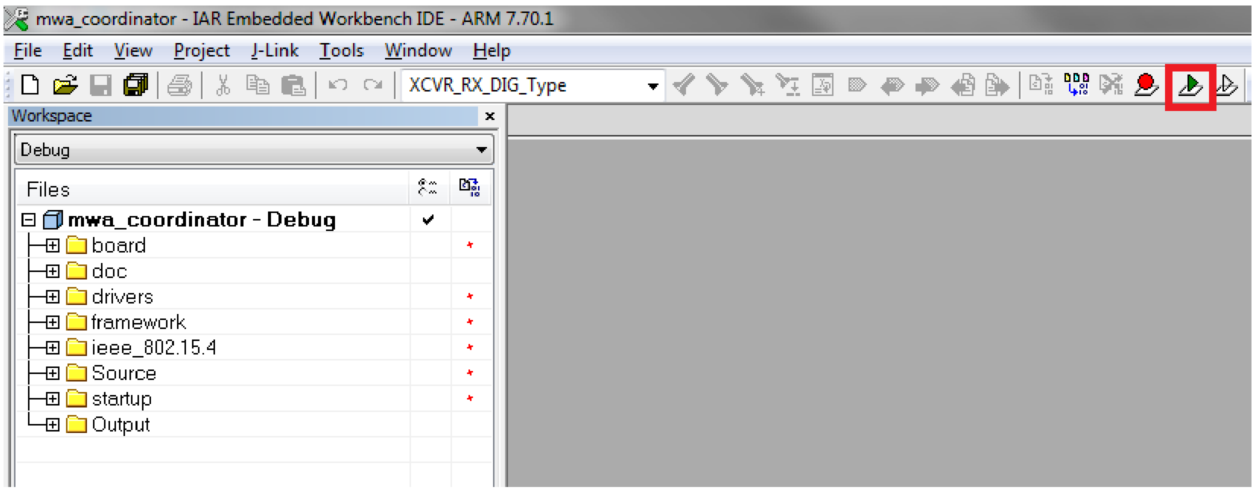


**Figure 25: MyWirelessApp coordinator project**

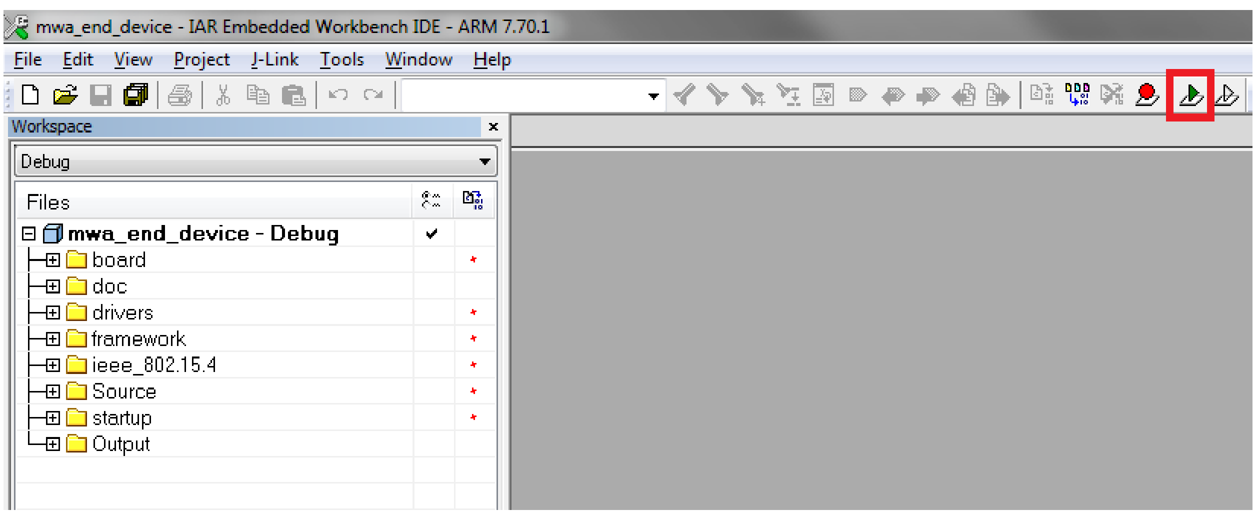


**Figure 26: MyWirelessApp end device project**

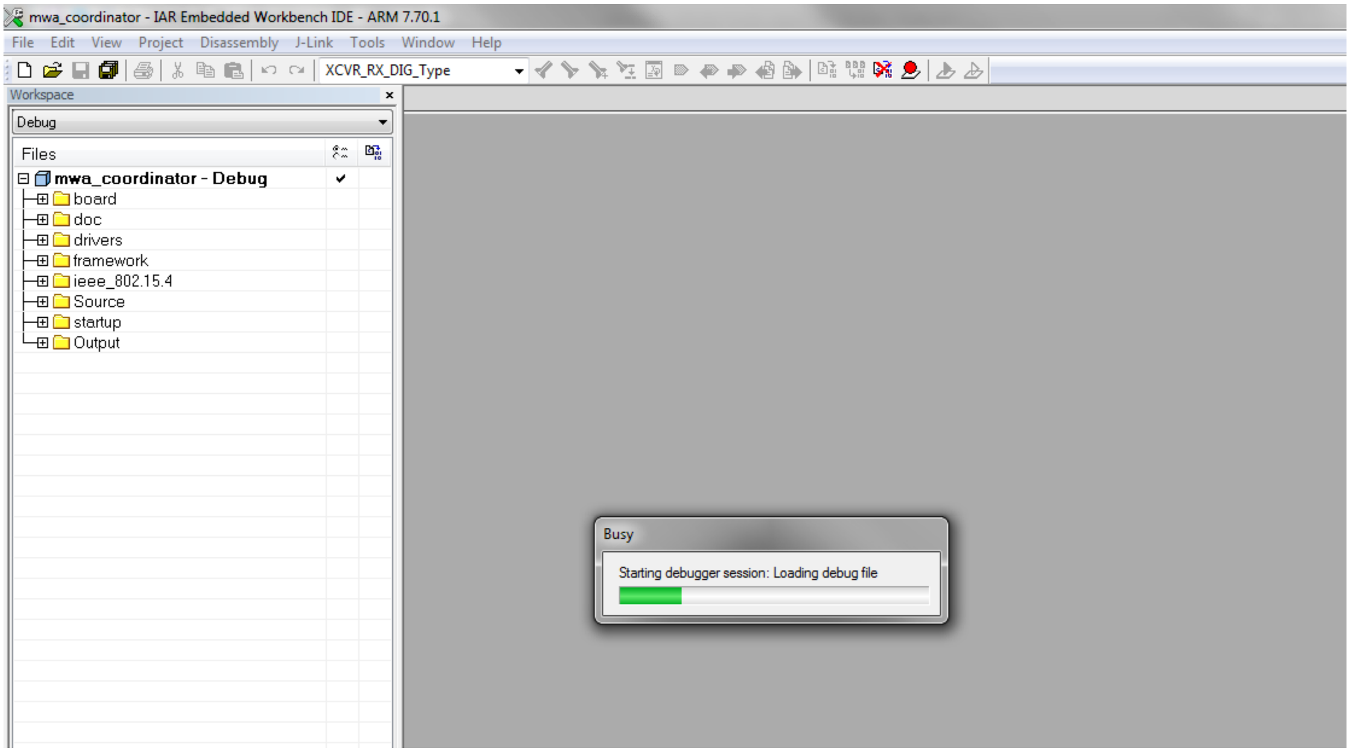


Figure 27: MyWirelessApp Coordinator loading stage example

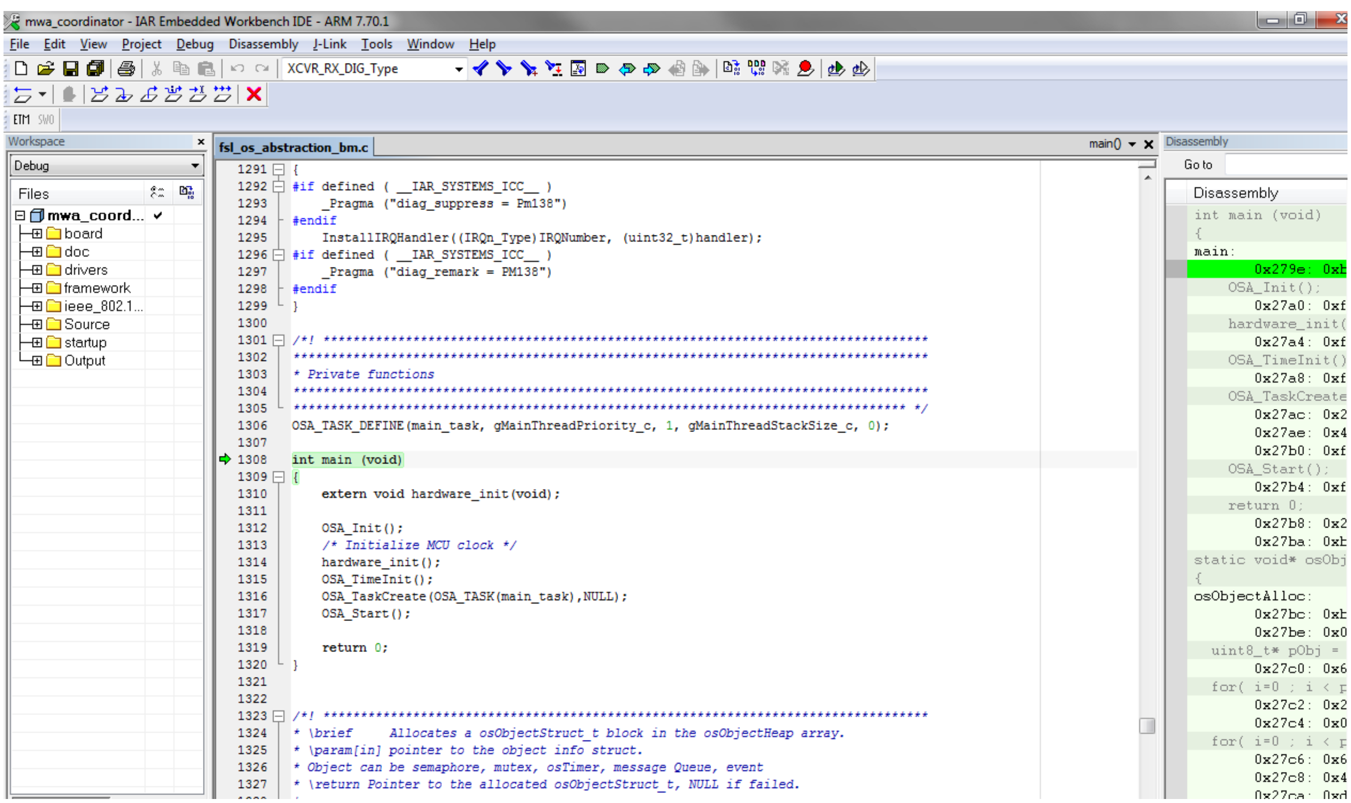


Figure 28: MyWirelessApp Coordinator application loaded

Step 2:

After loading both application check “Device Manager” to get the serial ports numbers. These should appear with the prefix “jLink”.

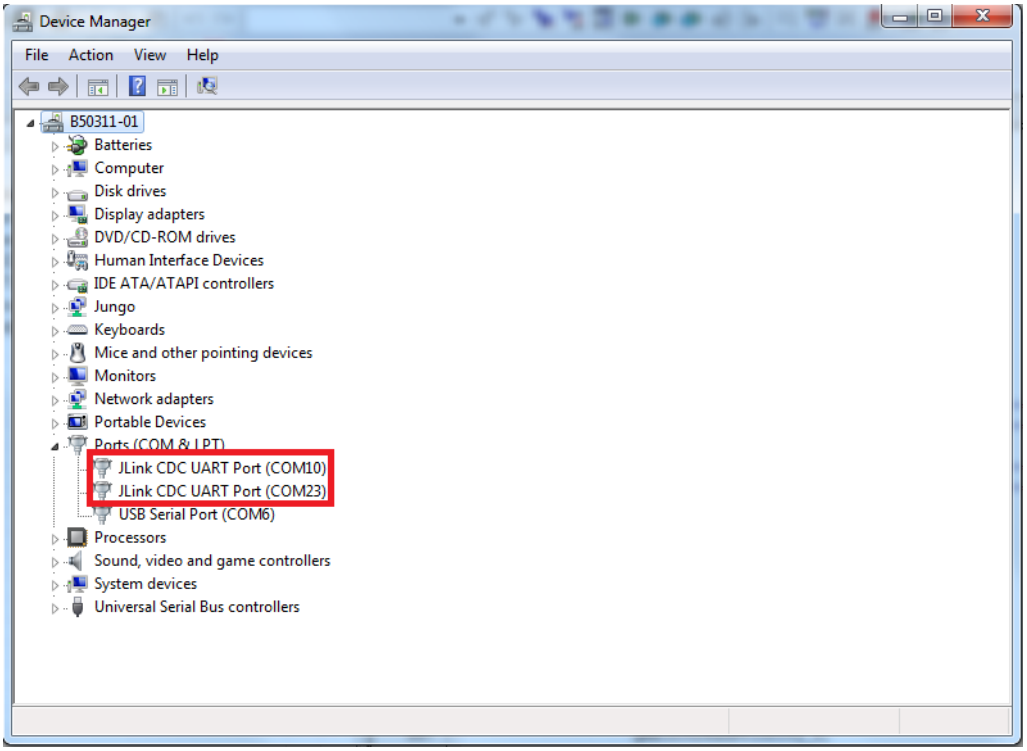


Figure 29: **Device Manager serial port look up**

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.

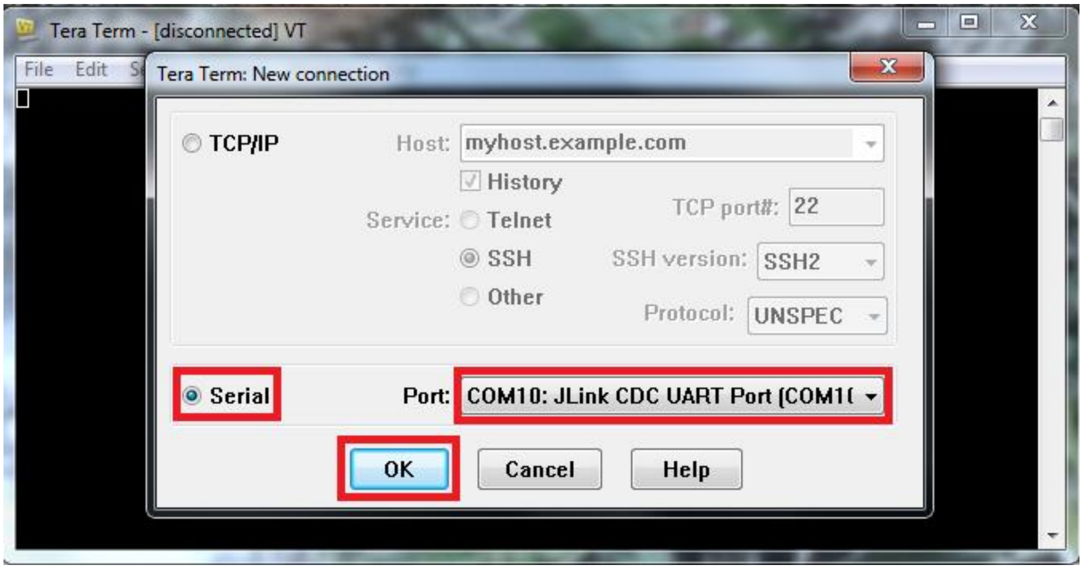


Figure 30: Select JLink serial connection COM port

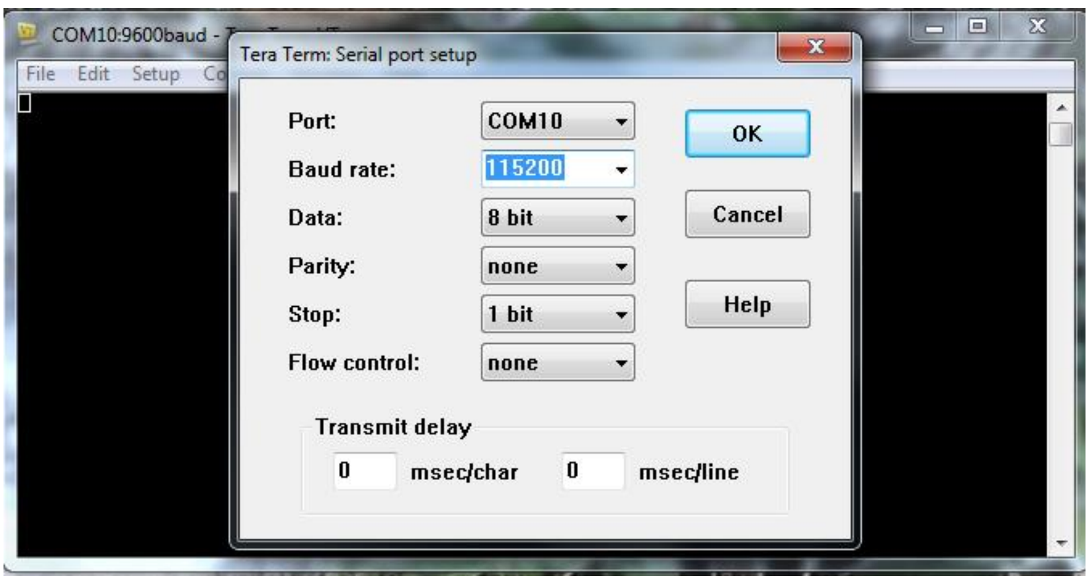


**Figure 31: Setting correct baud rate**

Step 4:

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

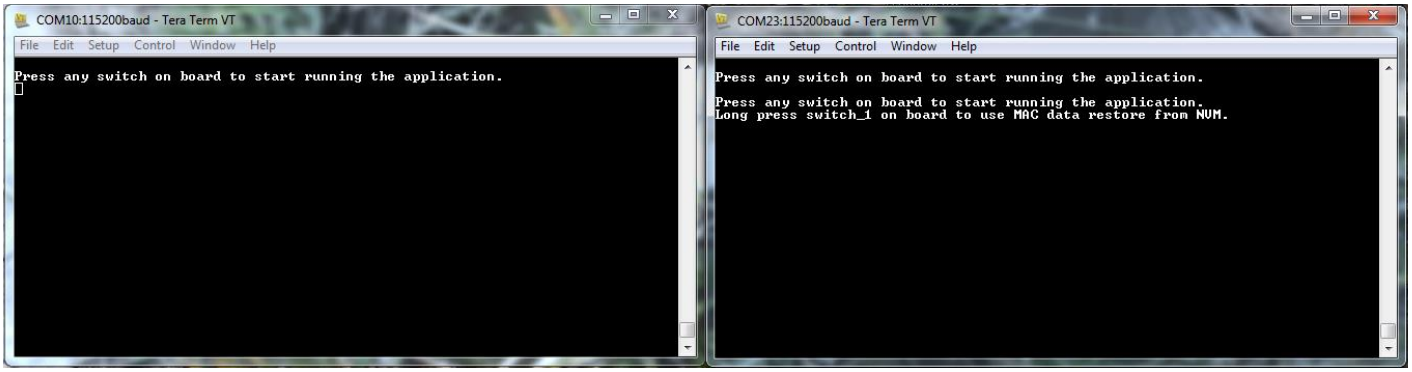


**Figure 32: Both applications after a reset**

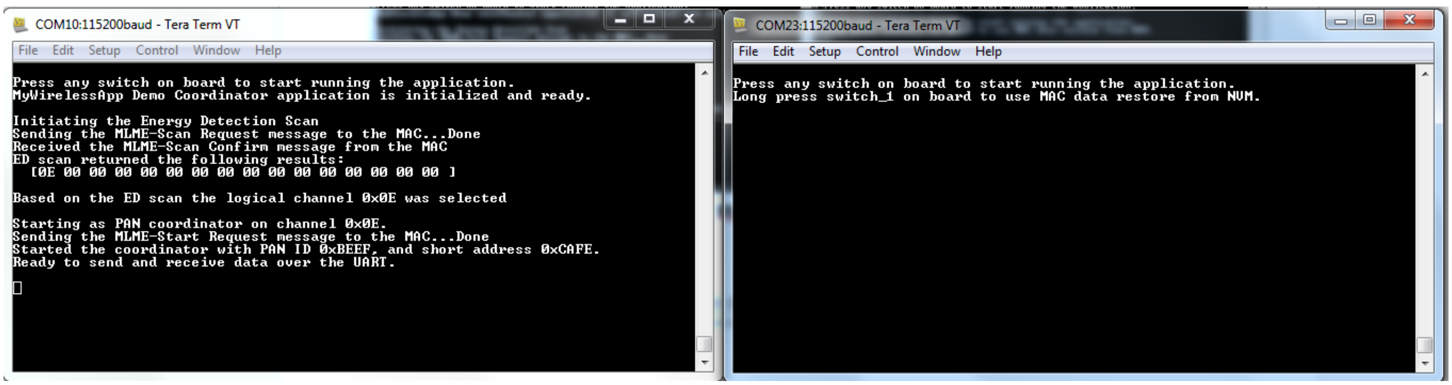


**Figure 33: MyWirelessApp coordinator started**

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.

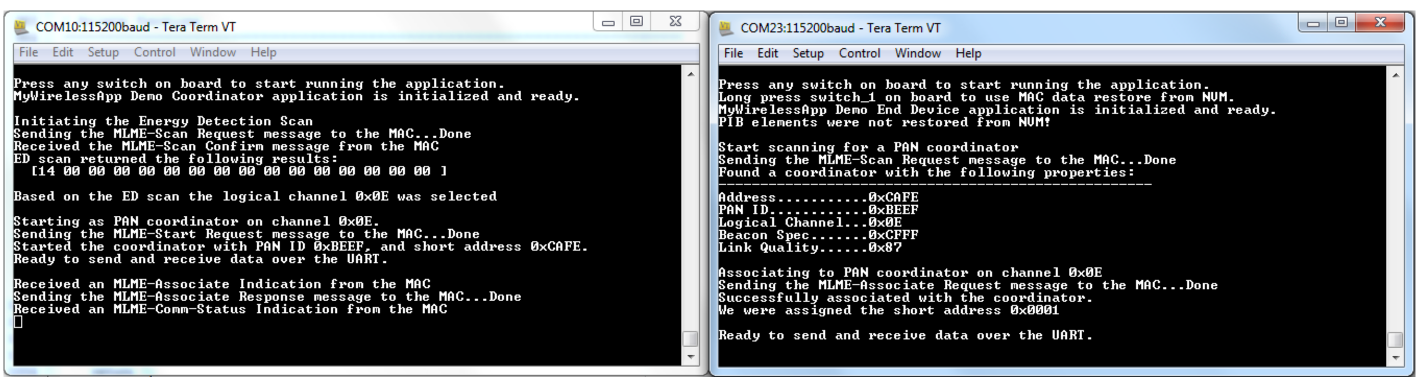


**Figure 34: MyWirelessApp end device associated to coordinator**

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

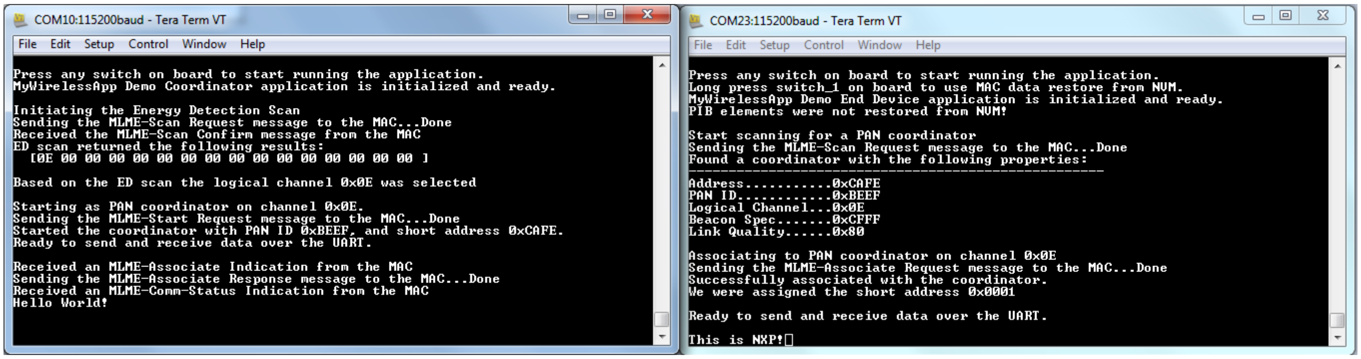


**Figure 35: MyWirelessApp message exchange**

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*).

*How to Reach Us:*

**Home Page:**
www.nxp.com

**Web Support:**
www.nxp.com/support



Document number: MKW41Z802154SW532QSG
Rev. 0
03/2017