NXP Semiconductors
User's Guide

Document Number: MKW41ZBLESW123QSG
Rev. 0,
04/2017

Kinetis MKW41Z Bluetooth® Low Energy Software

Quick Start Guide

Introduction

This document is a brief presentation of the NXP Bluetooth® Low Energy Software for the KW41Z wireless microcontroller platforms version 1.2.3. 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:

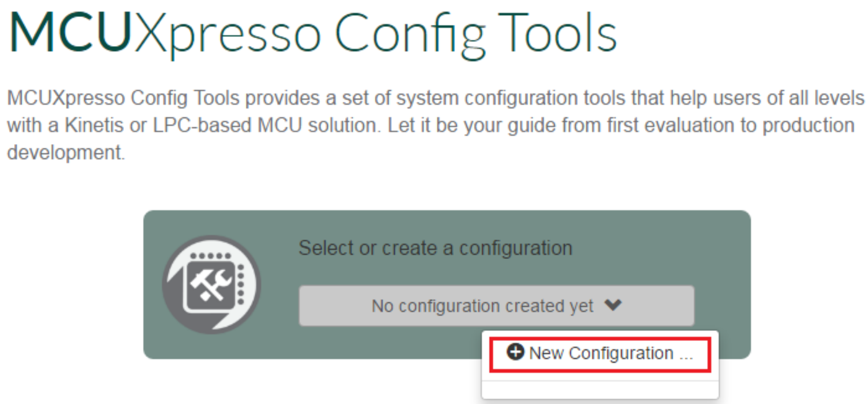

Figure 1: The MCUXpresso welcome screen

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

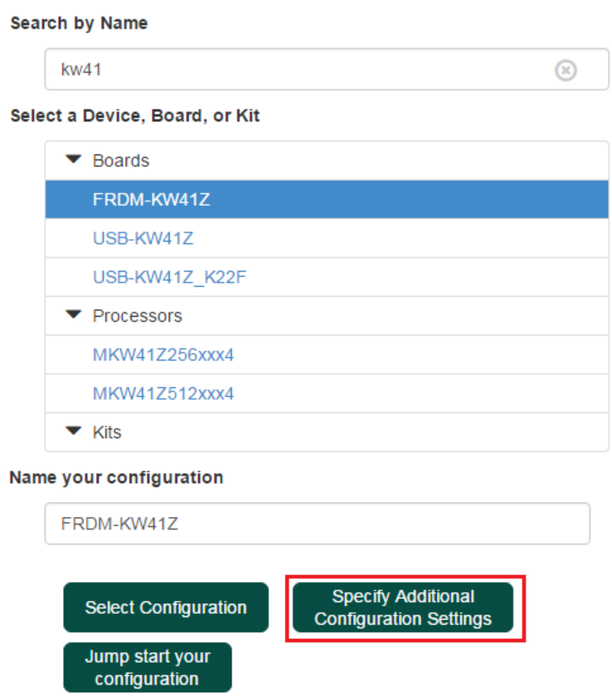

Figure 2: Available configurations

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

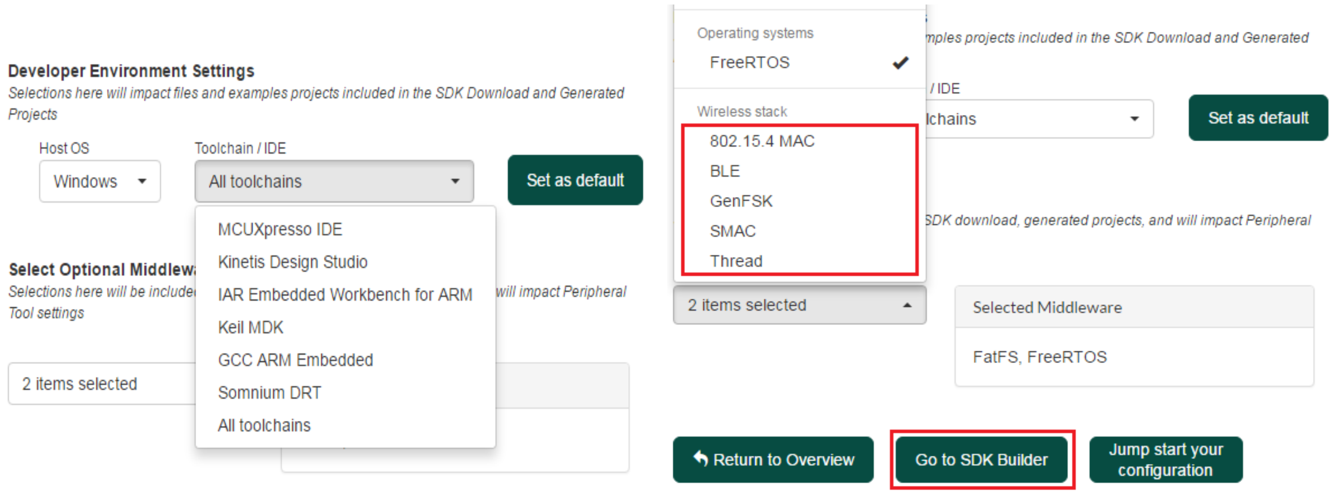

Figure 3: Select toolchain(s) and middleware

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

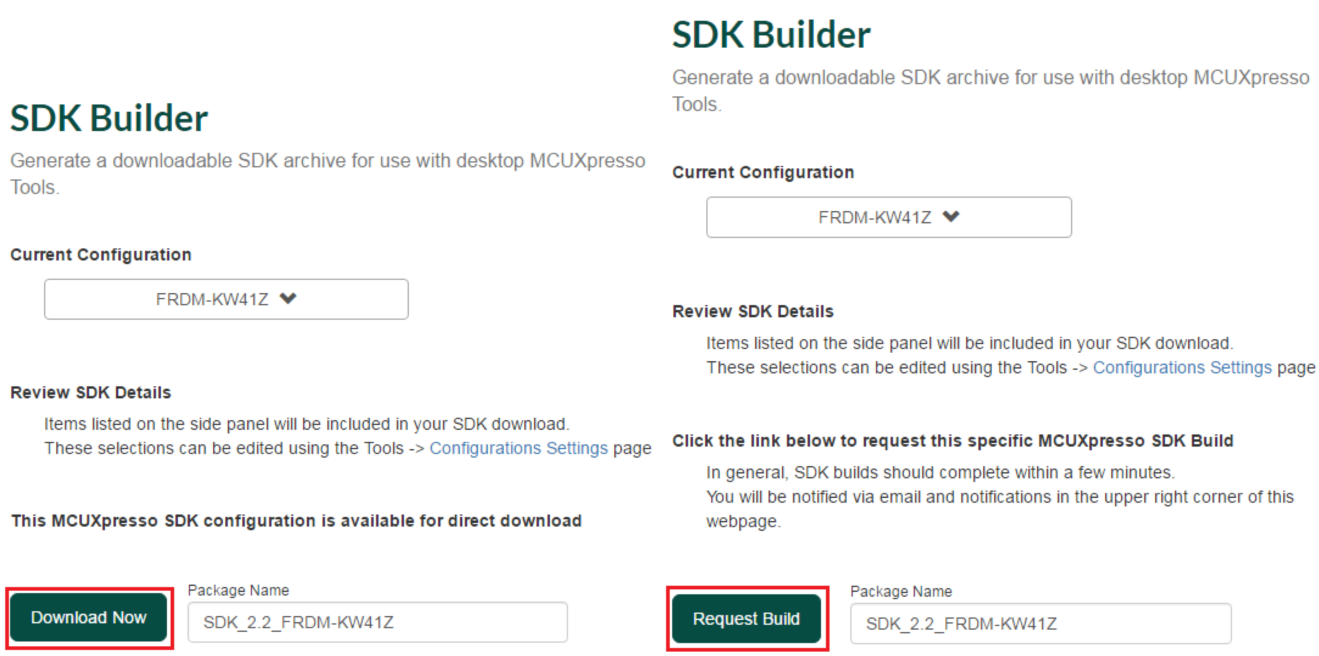

Figure 4: The SDK Builder

If you request a new build, you will receive a notification when process ends. To download the archive, access the SDK Archive manager:

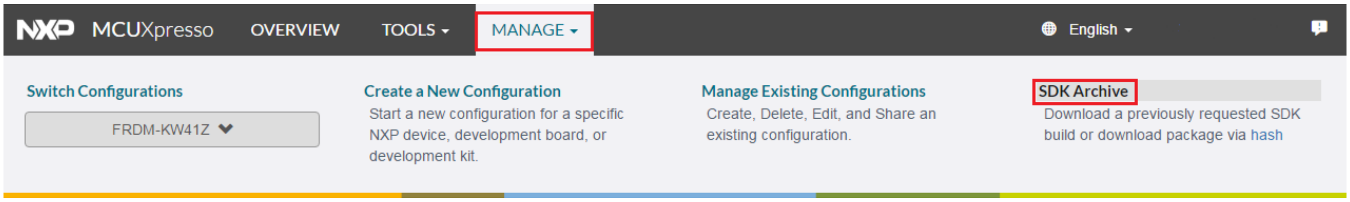

Figure 5: Access the Archive manager

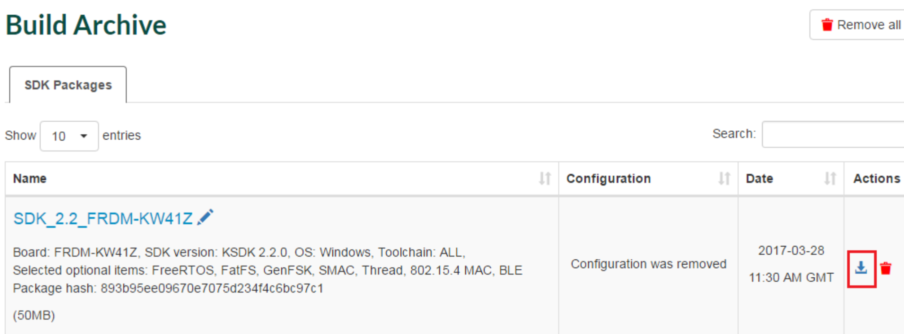

Figure 6: Download the archive

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 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 *heart_rate_sensor* application. All applications can be found using the following placeholders for text:

- <connectivity_path>: represents the root path for the SDK.
- <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; it can be either “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\bluetooth\<demo_app>\<RTOS>\<IDE>\

Kinetis Bluetooth Low Energy Software Demo Application Build Example

Selected app: heart_rate_sensor

Board: frdmkw41z

RTOS: FreeRTOS

Resulting location:

<connectivity_path>\boards\frdmkw41z\wireless_examples\bluetooth\heart_rate_sensor\freertos\<IDE>

NOTE

If your FRDM-KW41Z board is configured for the buck or boost modes of the KW41Z DCDC converter, 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.

The Heart Rate sensor application is configured for low power operation by default. In this case, when the core enters in sleep mode, the debug probe will fail. If you want to disable the low power functionality, the gPWR_UsePowerDownMode define need to be set to 0 in the app_preinclude.h header file.

2.1 Building and Flashing the BLE 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.

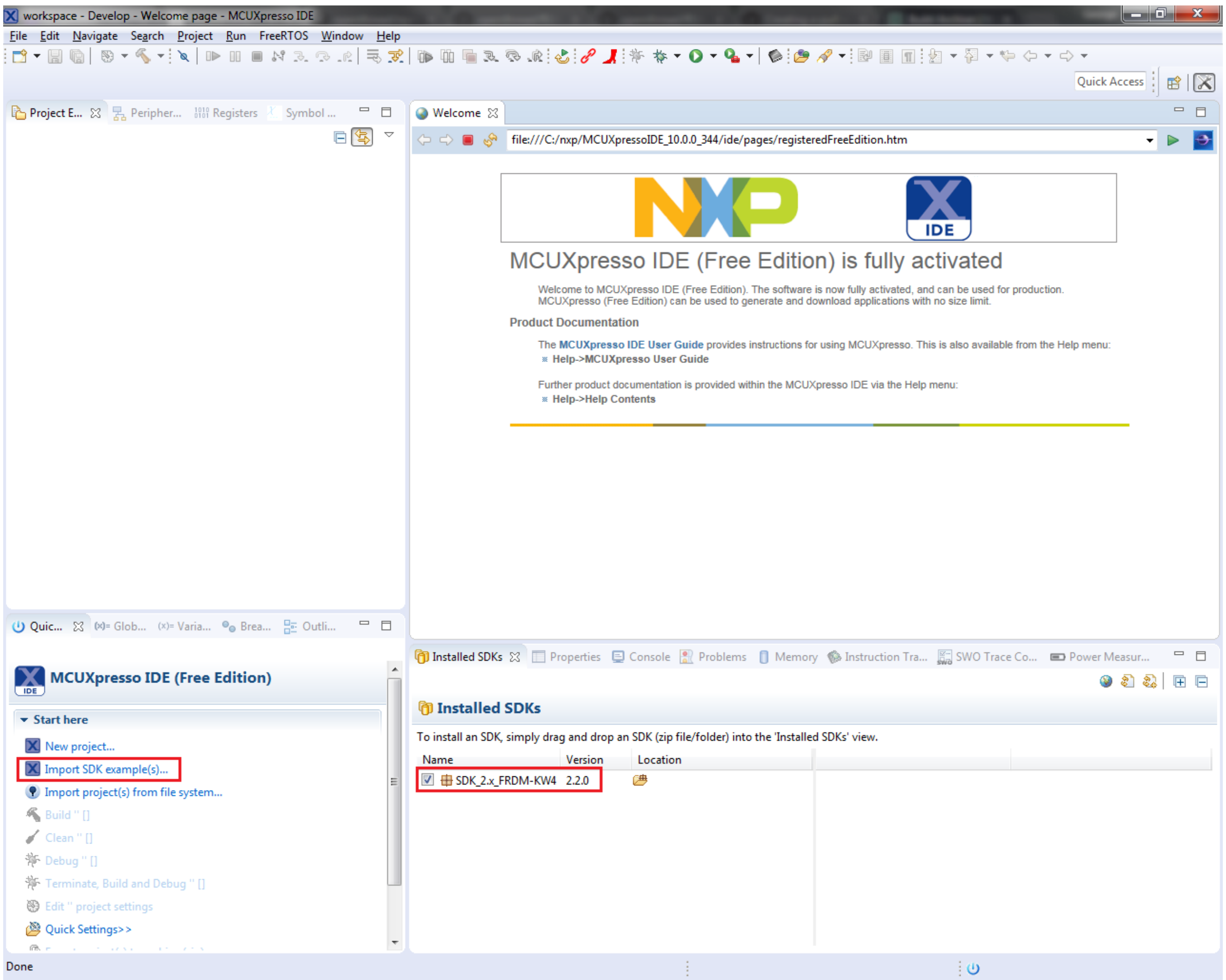

Figure 7: Installed SDKs

Step 3:

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

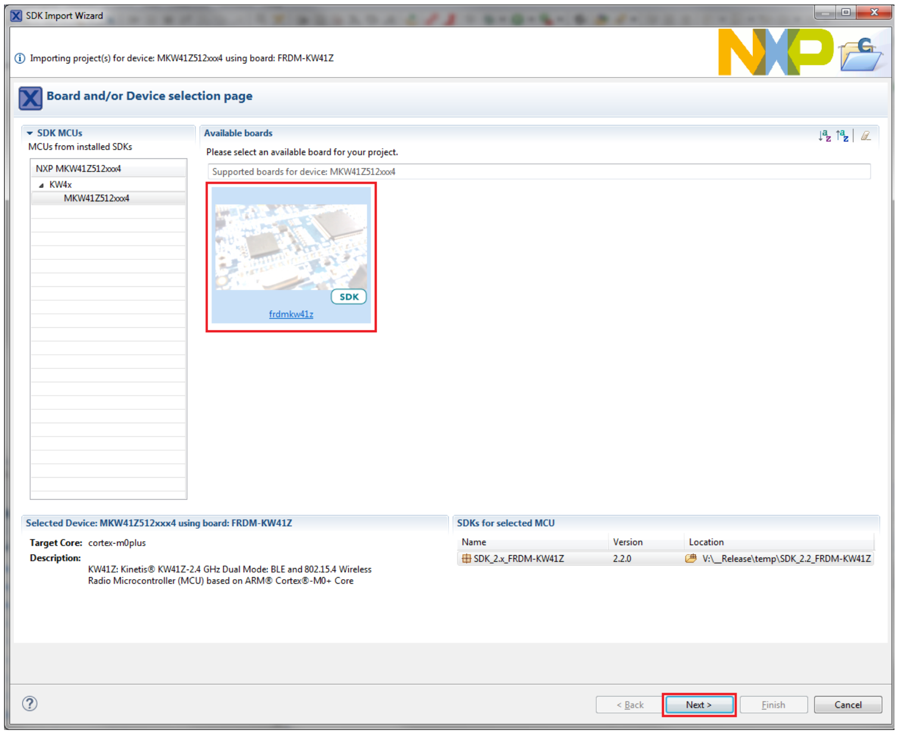

Figure 8: Select the board

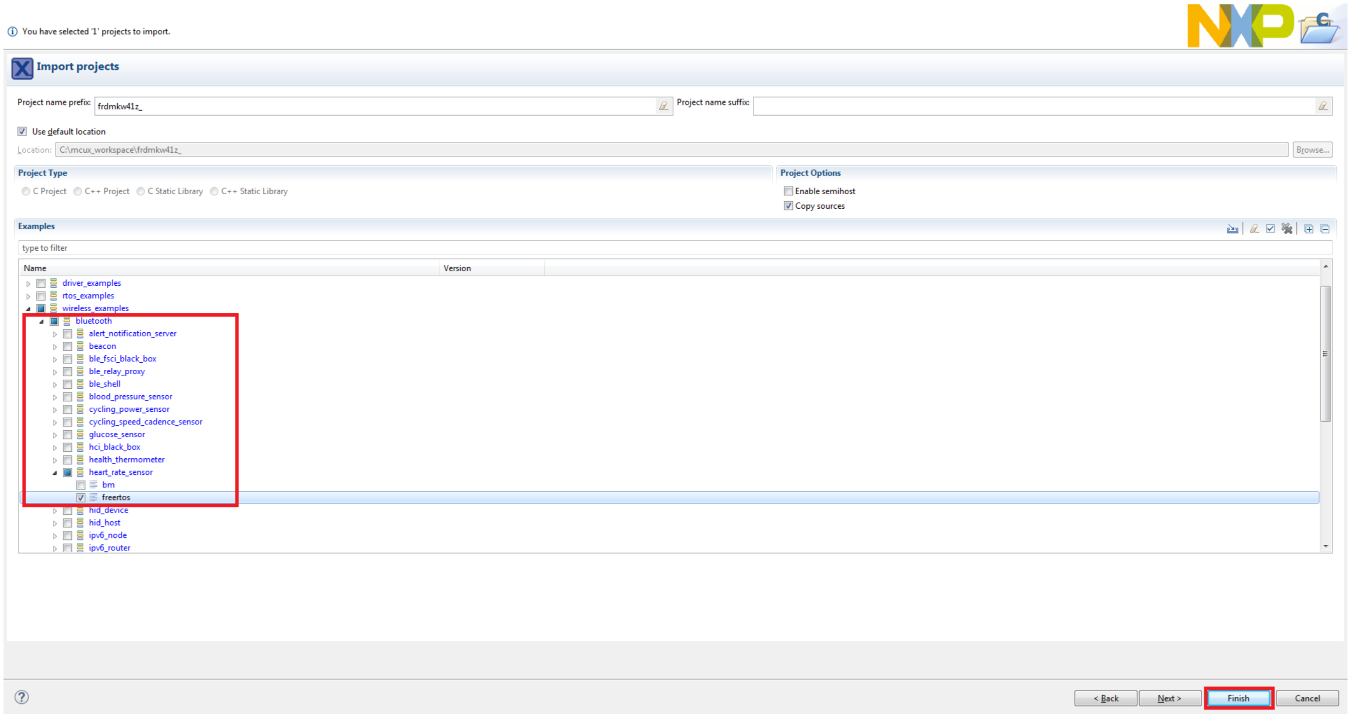

Figure 9: Select the example(s)

Step 4:

Build the heart_rate_sensor project.

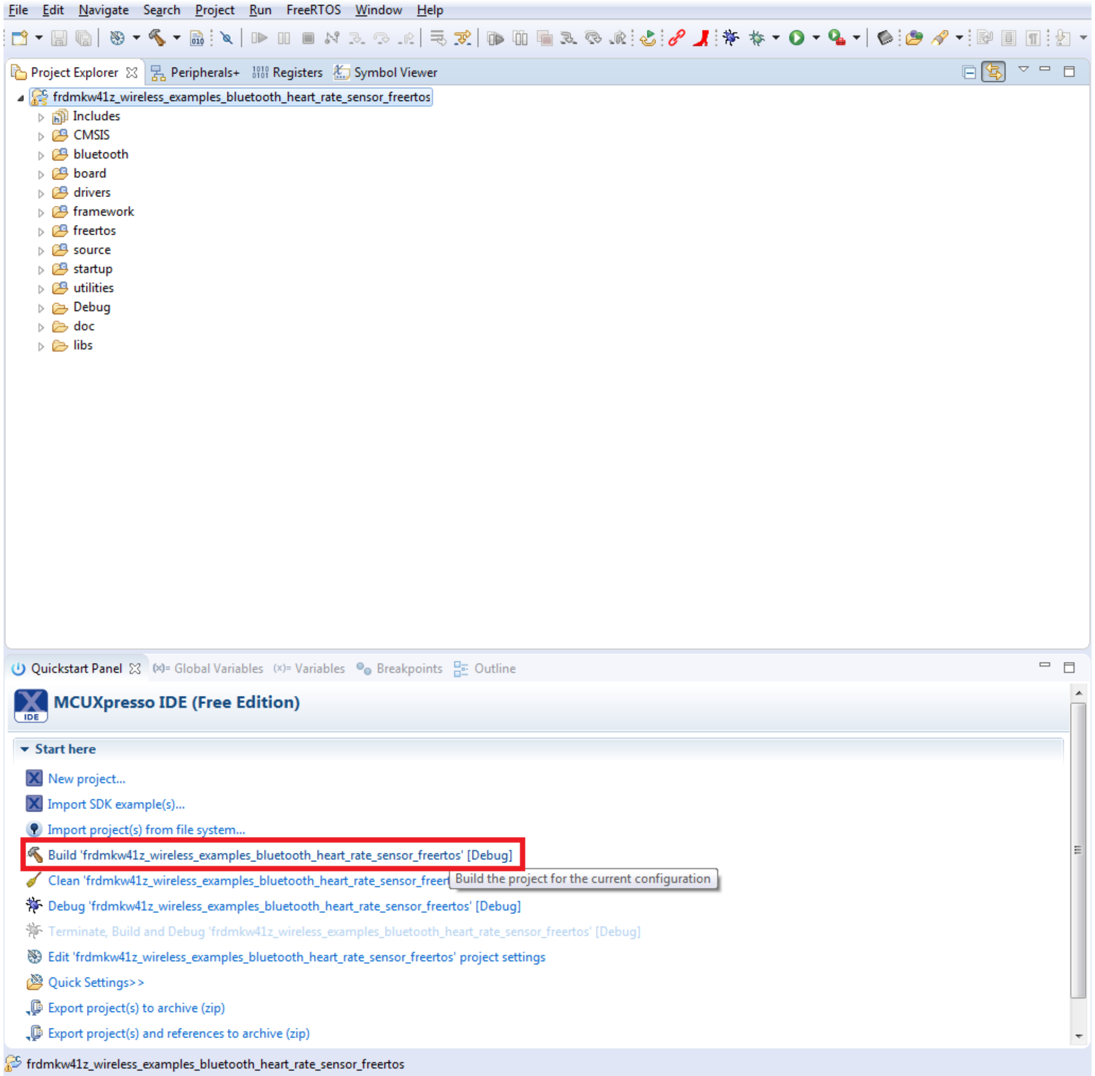

Figure 10: "heart_rate_sensor" FreeRTOS build

Step 5:

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

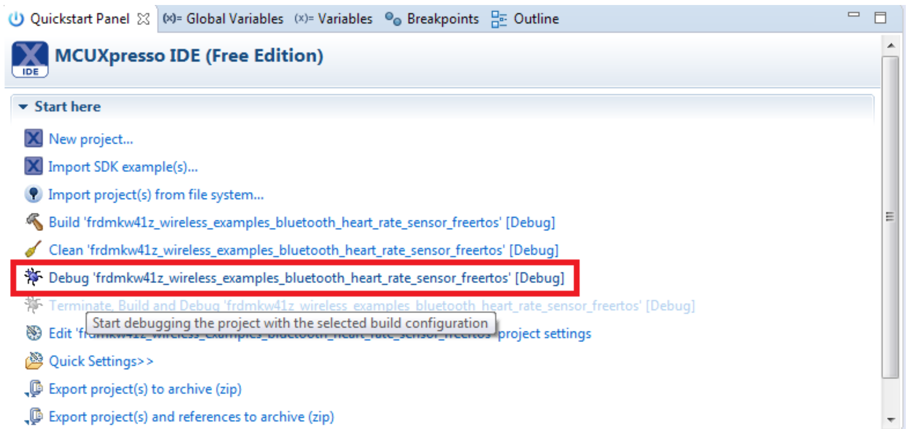

Figure 11: “heart_rate_sensor” Debug

2.2 Building and Flashing the BLE Software Demo Applications using IAR

Step 1:

Navigate to the resulting location in the SDK root directory.

Step 2:

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

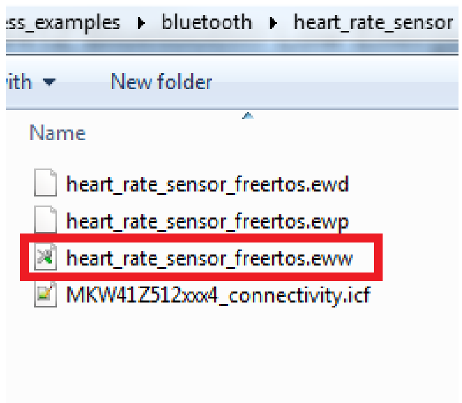

Figure 12: "Heart Rate Sensor" IAR demo project location

Step 3:

Select the Heart Rate Sensor project.

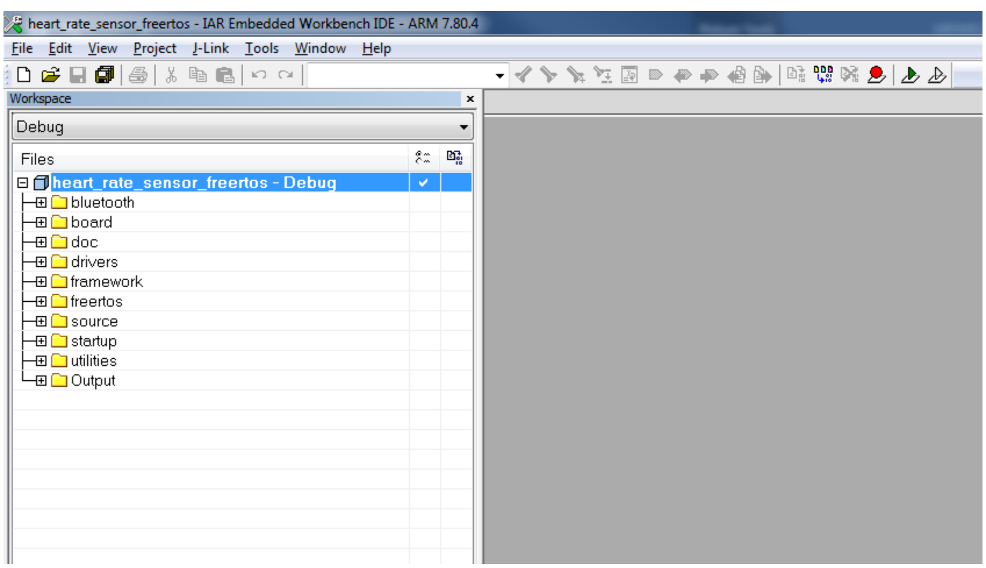

Figure 13: Heart Rate Sensor FreeRTOS IAR project

Step 4:

Build the Heart Rate Sensor project.

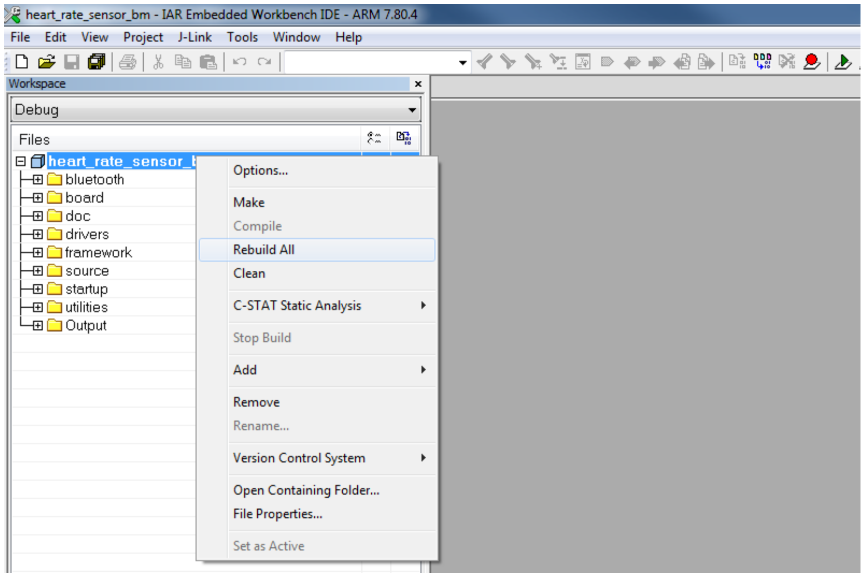

Figure 14: Heart Rate Sensor build

Step 5

Make the appropriate debugger settings in the project options window:

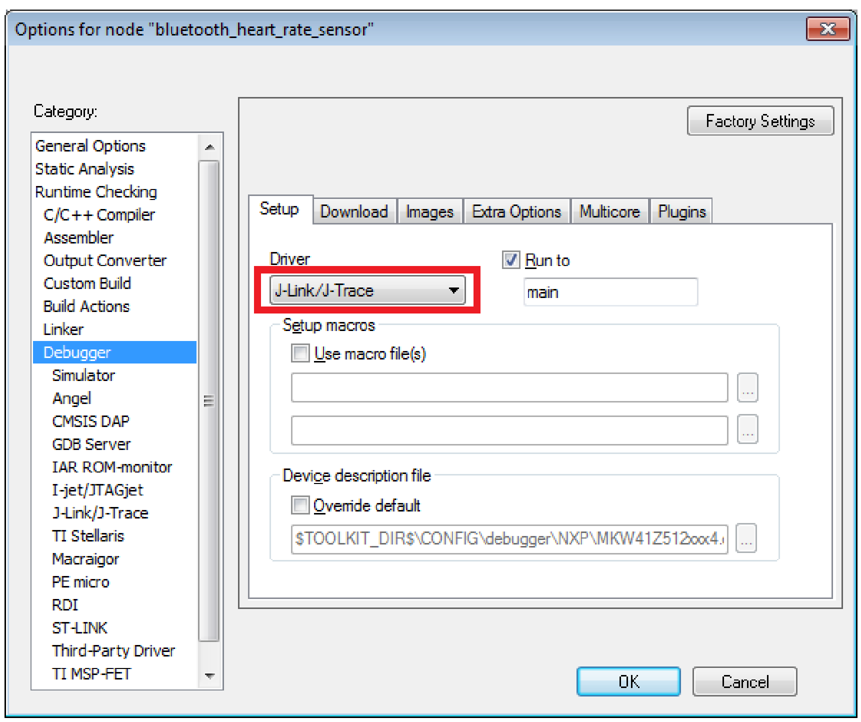

Figure 15: Debugger Settings

Step 6:

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

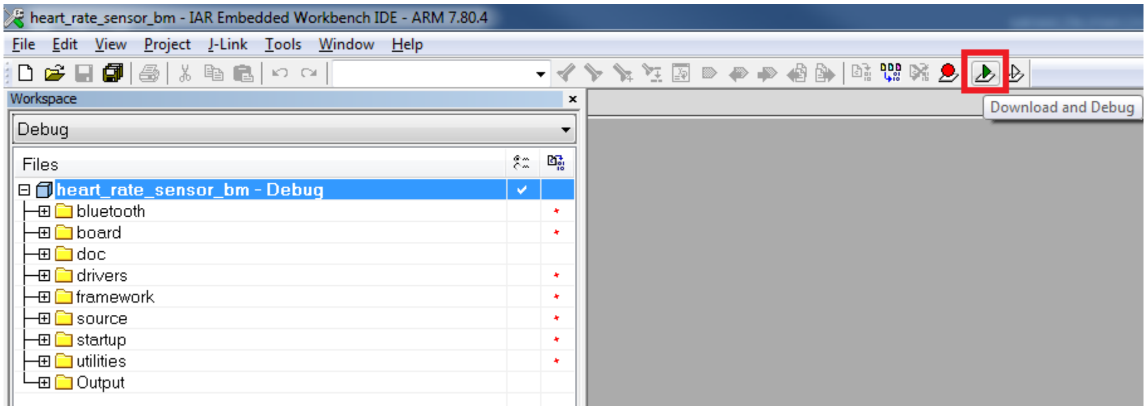

Figure 16: Heart Rate Sensor Download and Debug

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.

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.

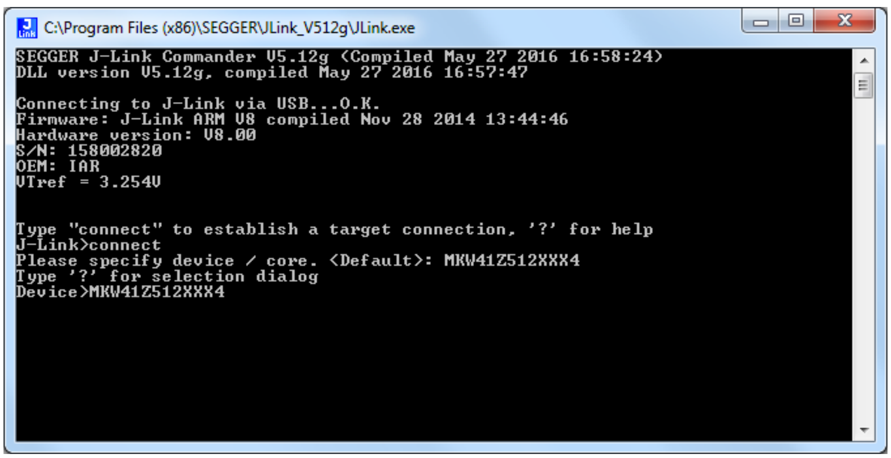

Figure 17: MKW41Z512xxx4 device selection

Step 2: Select SWD target interface.

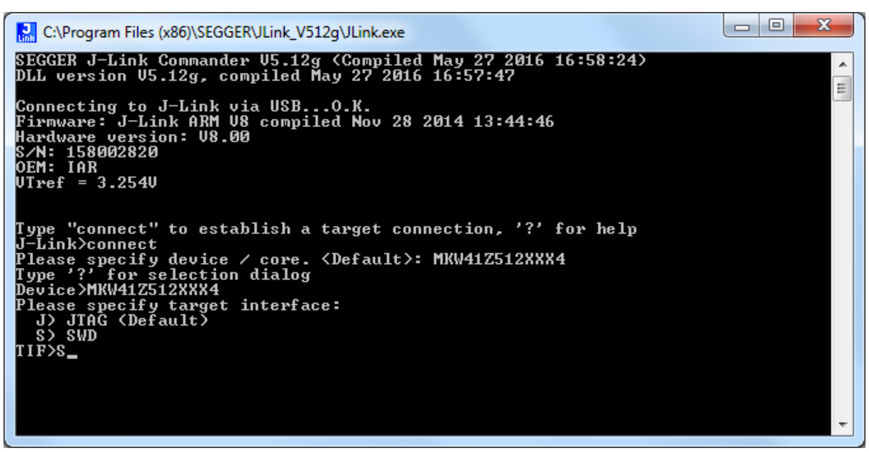

Figure 18: SWD interface selection

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

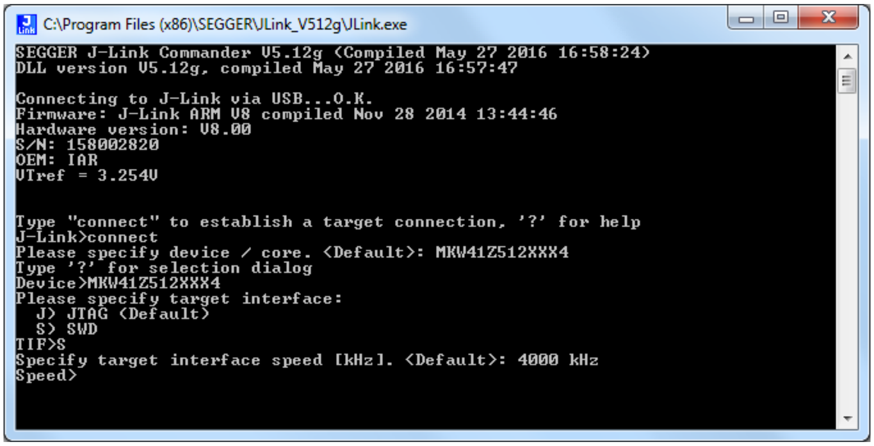

Figure 19: SWD interface speed selection

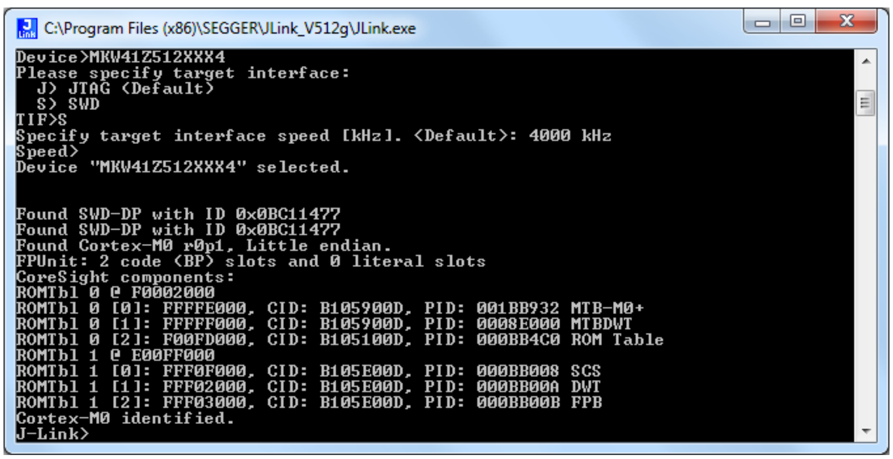

Figure 20: Cortex-M0 identified

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

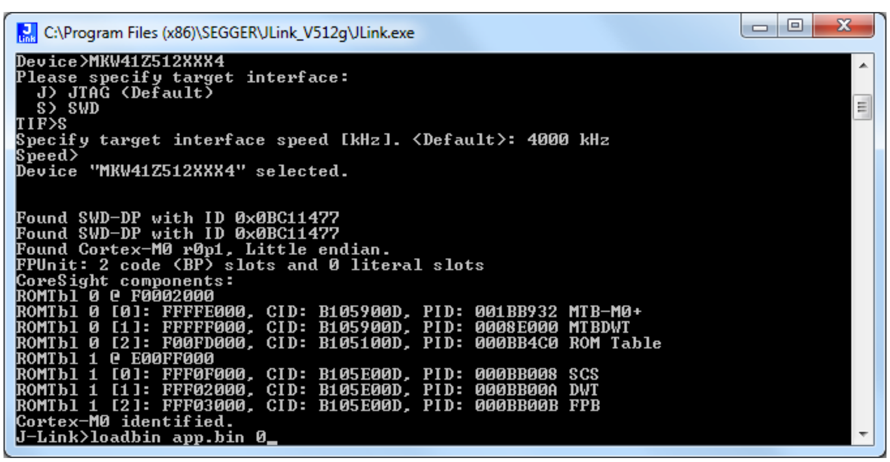

Figure 21: Load binary file

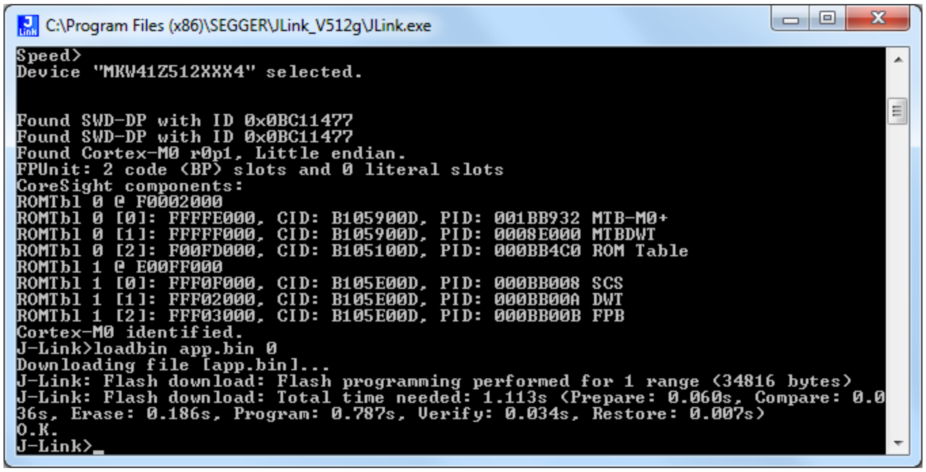

Figure 22: Download completed successfully

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) VDCDC_IN (1.71 to 3.6V) Operation 1.8V - 3.6 V	1-2	1-2	3-4	1-3 2-4 5-6
Boost Mode (auto start) VDCDC_IN (0.9V to 1.8V) Single Battery Operation	2-4	3-4	1-2 5-6	5-6
Buck Mode (manual start) VDCDC_IN (1.8V to 4.2V) Coin Cell Battery Operation	2-4	5-6 press SW6 to start	3-4	5-6
Buck Mode (auto start) VDCDC_IN (1.8V to 4.2V) Coin Cell Battery Operation	2-4	3-4	3-4	5-6

Figure 24: FRDM-KW41Z Jumper Configuration for DCDC Modes

4 Running the Heart Rate sensor application using Kinetis BLE Toolbox mobile application

This section does not cover the installation of the *Kinetis BLE Toolbox* mobile application and it assumes you have already installed the latest version from the application store.

Step 1:

Flash the board with the Heart Rate sensor application as previously described. If low power support is enabled, no LED's will be flashing.

Step 2:

Make sure your phone's Bluetooth Device is enabled.

Step 3:

Open the *Kinetis BLE Toolbox* application and select the *Heart Rate* icon.

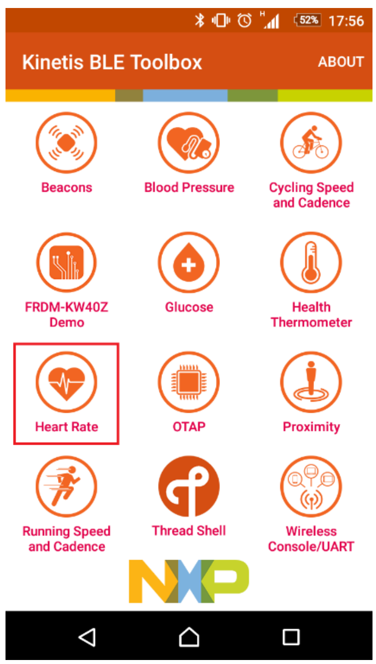

Figure 25: Kinetis BLE Toolbox - opening Heart Rate application

Step 4:

Press **SW4** on the FRDM-KW41Z board to start advertising. The device should become visible for the Heart Rate application.

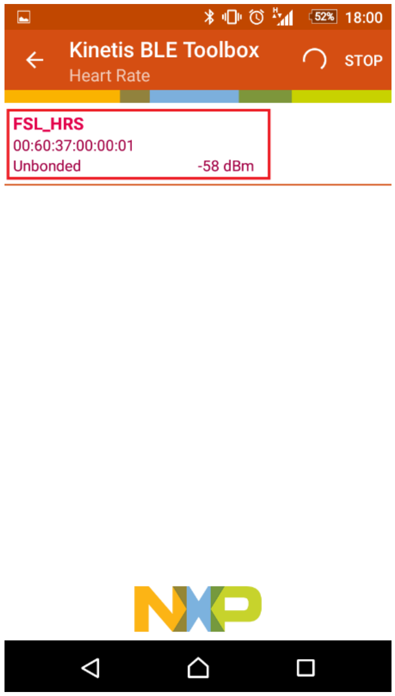

Figure 26: Selecting the scanned device

Step 5:

Select the device that appears in the *Heart Rate* tab to connect to it. After connecting, the mobile application starts plotting the randomly generated heart rate values sent by the FRDM-KW41Z device.

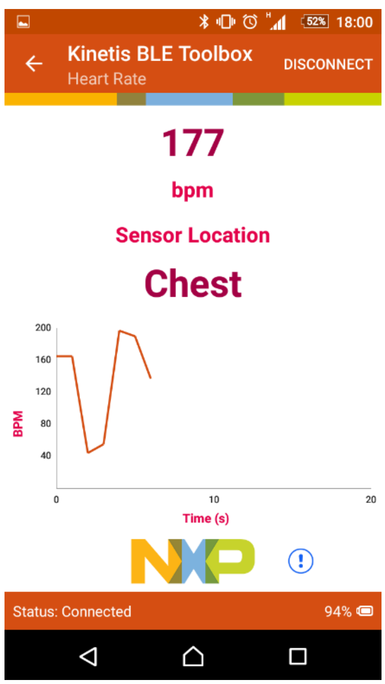

Figure 27: Heart Rate sensor device functionality in connected mode

How to Reach Us:

Home Page:
www.nxp.com

Web Support:
www.nxp.com/support

.

Document number: MKW41ZBLESW123QSG
Rev. 0
04/2017