

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

Introduction	1
1 Download	2
2 Building the Binaries	5
2.1 Building and Flashing the BLE Software Demo Applications using MCUXpresso IDE...	7
2.2 Building and Flashing the BLE Software Demo Applications using IAR.....	12
2.3 Flashing a Binary Image File Without Using an IDE.....	15
3 Hardware Setup	18
4 Running the Heart Rate sensor application using Kinetis BLE Toolbox mobile application	20



1 Download

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

MCUXpresso Config Tools

MCUXpresso Config Tools provides a set of system configuration tools that help users of all levels with a Kinetis or LPC-based MCU solution. Let it be your guide from first evaluation to production development.

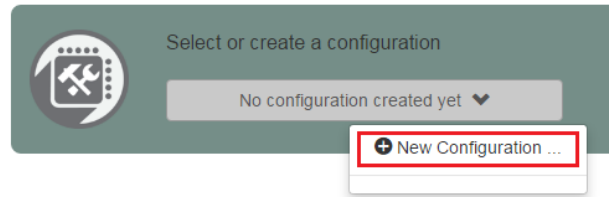


Figure 1: The MCUXpresso welcome screen

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

Search by Name

Select a Device, Board, or Kit

▼ Boards

- FRDM-KW41Z
- USB-KW41Z
- USB-KW41Z_K22F

▼ Processors

- MKW41Z256xxx4
- MKW41Z512xxx4

▼ Kits

Name your configuration

Select Configuration

Specify Additional Configuration Settings

Jump start your configuration

Figure 2: Available configurations

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

Developer Environment Settings
Selections here will impact files and examples projects included in the SDK Download and Generated Projects

Host OS: Windows

Toolchain / IDE: All toolchains (dropdown menu open showing: MCUXpresso IDE, Kinetis Design Studio, IAR Embedded Workbench for ARM, Keil MDK, GCC ARM Embedded, Somnium DRT, All toolchains)

Wireless stack: 802.15.4 MAC (dropdown menu open showing: 802.15.4 MAC, BLE, GenFSK, SMAC, Thread)

Selected Middleware: FatFS, FreeRTOS

Buttons: Return to Overview, **Go to SDK Builder**, Jump start your configuration

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:

SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.

Current Configuration
 FRDM-KW41Z

Review SDK Details
 Items listed on the side panel will be included in your SDK download. These selections can be edited using the Tools -> [Configurations Settings](#) page

This MCUXpresso SDK configuration is available for direct download

Package Name: SDK_2.2_FRDM-KW41Z

Buttons: **Download Now**, Request Build

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:

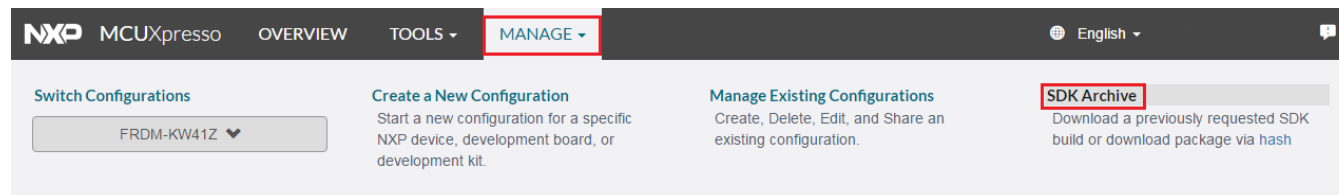


Figure 5: Access the Archive manager

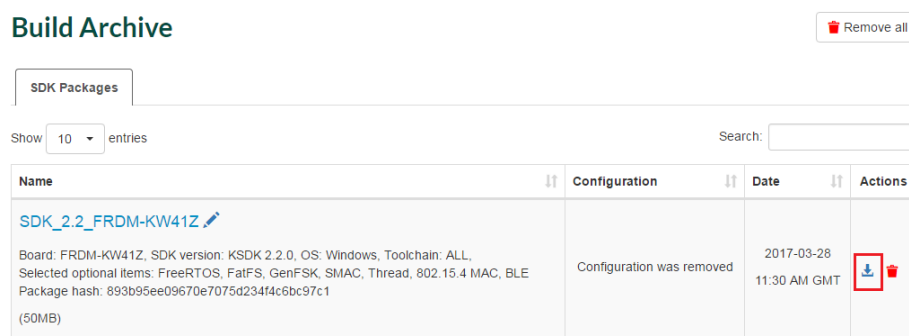


Figure 6: Download the archive

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.

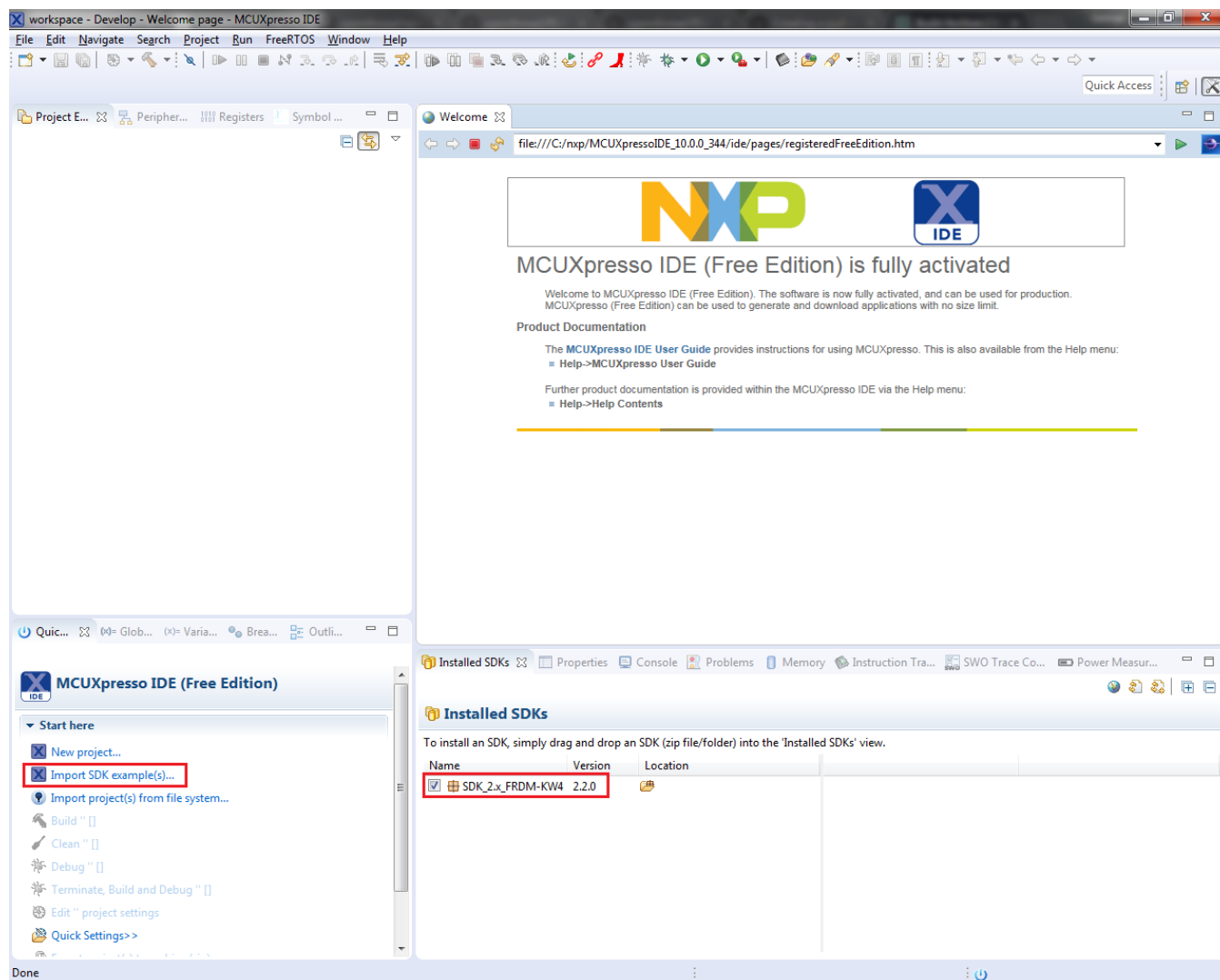


Figure 7: Installed SDKs

Step 3:

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

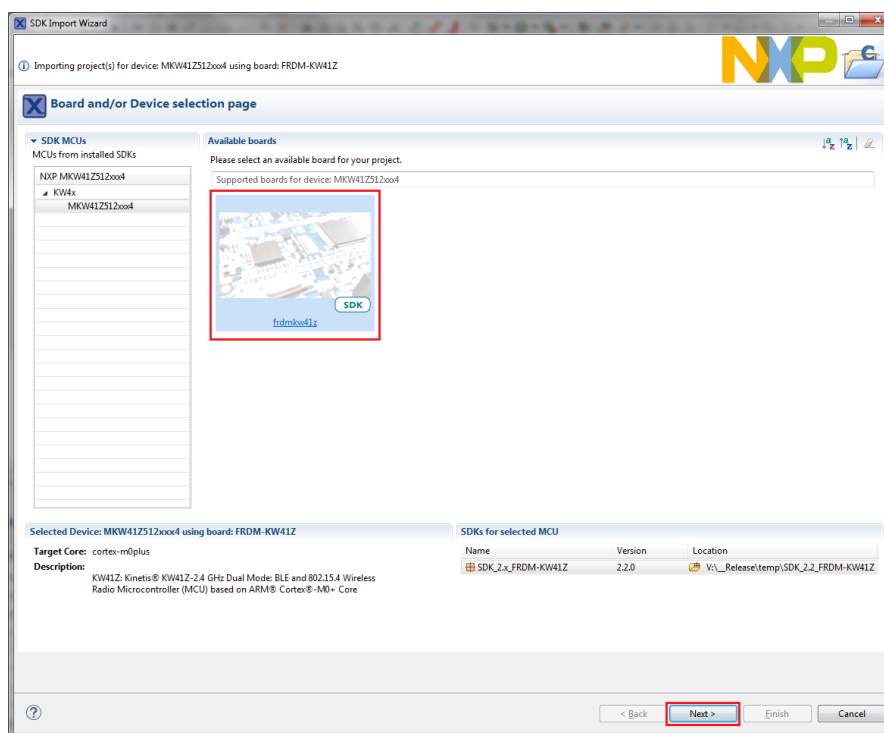


Figure 8: Select the board

You have selected 1 projects to import.

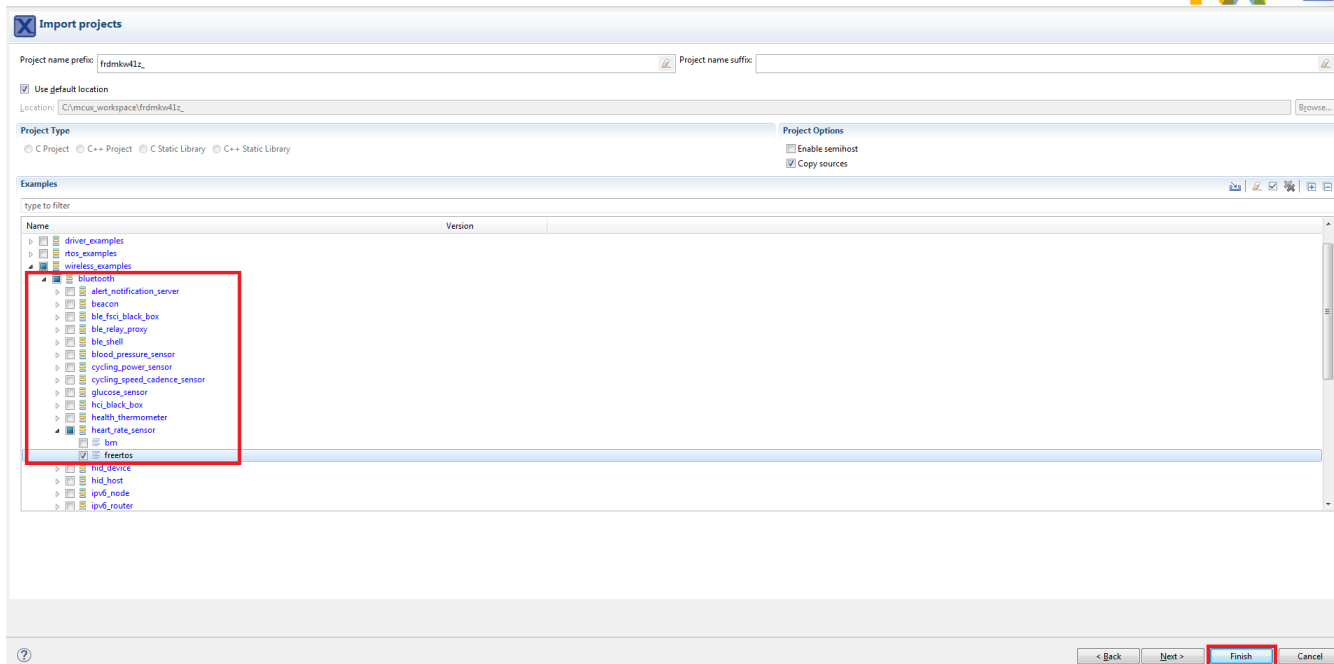


Figure 9: Select the example(s)

Step 4:

Build the heart_rate_sensor project.

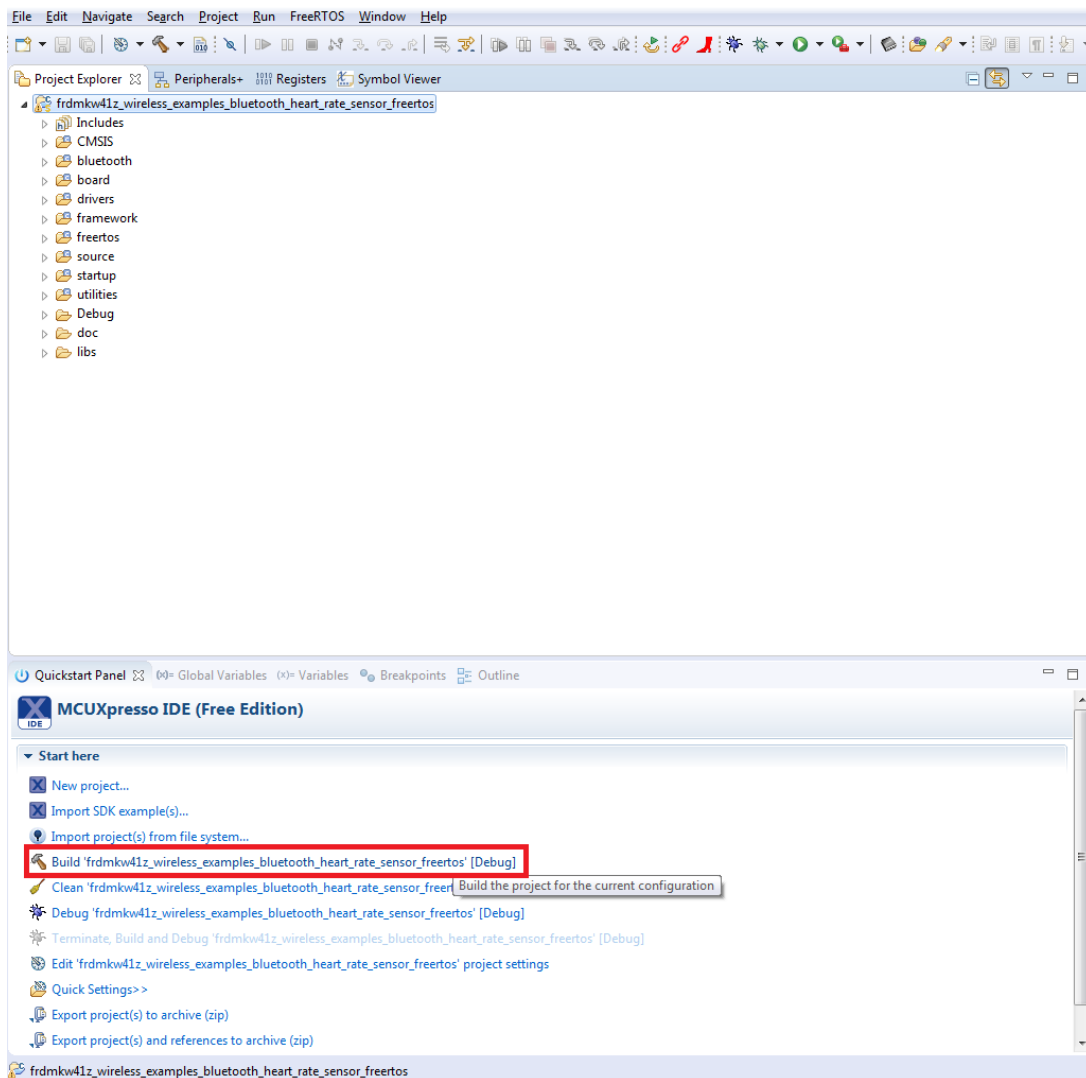


Figure 10: “heart_rate_sensor” FreeRTOS build

Step 5:

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

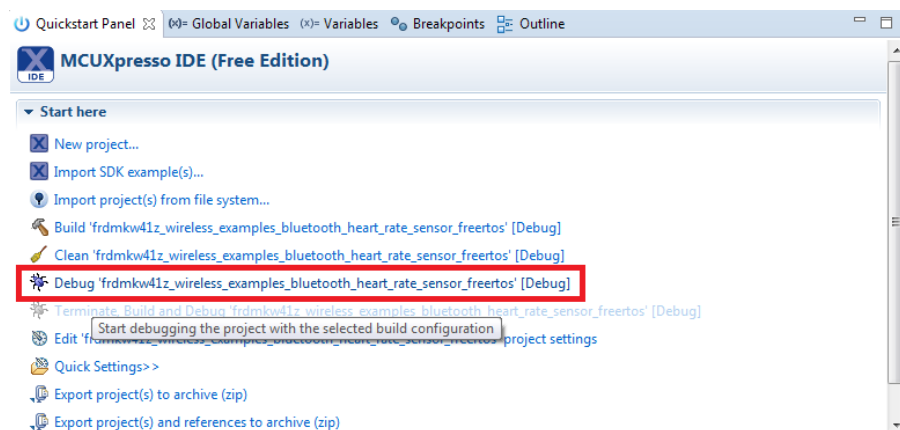


Figure 11: “heart_rate_sensor” Debug

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):

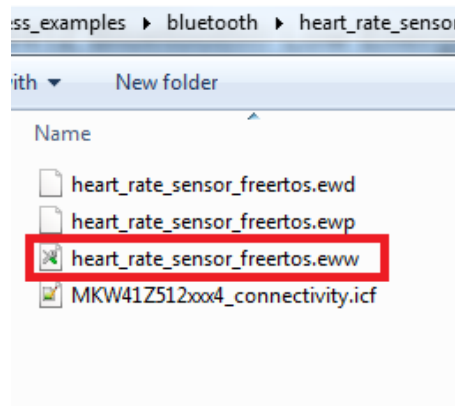


Figure 12: “Heart Rate Sensor” IAR demo project location

Step 3:

Select the Heart Rate Sensor project.

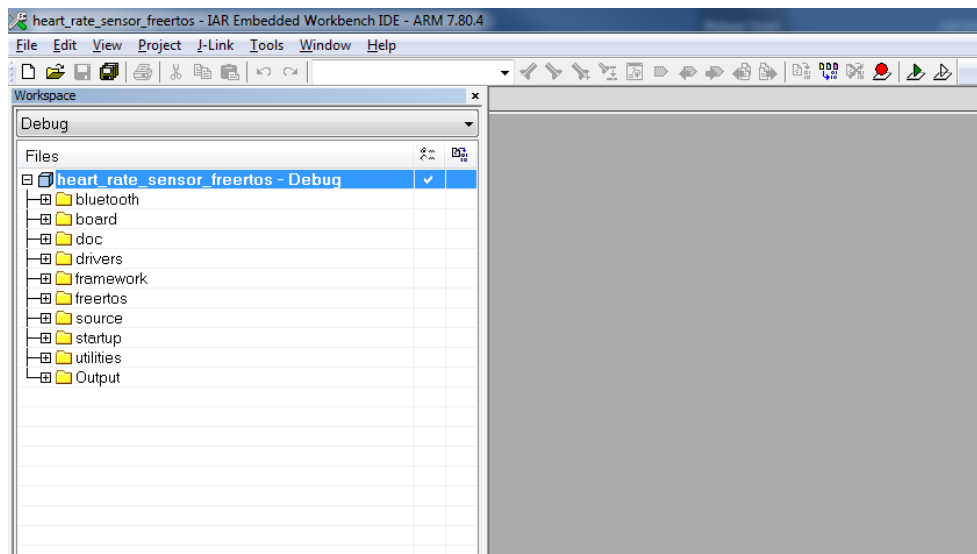


Figure 13: Heart Rate Sensor FreeRTOS IAR project

Step 4:

Build the Heart Rate Sensor project.

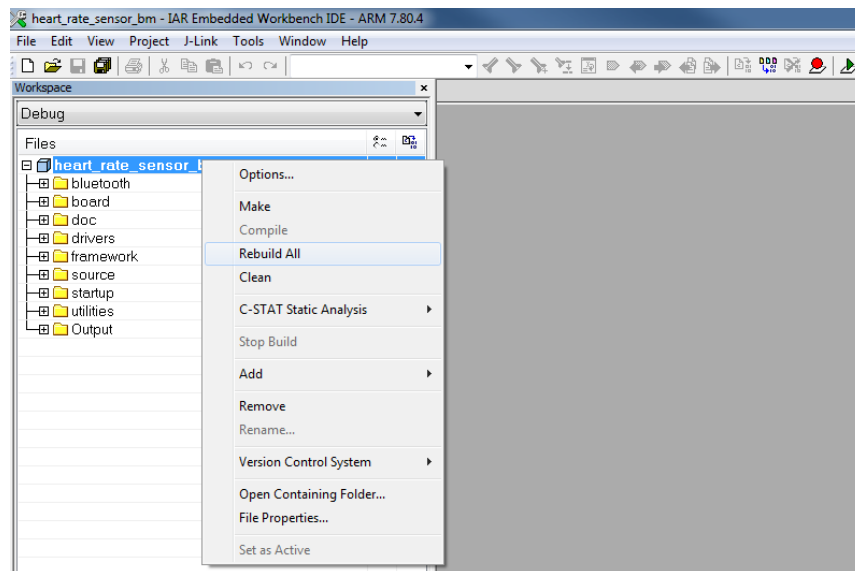


Figure 14: Heart Rate Sensor build

Step 5

Make the appropriate debugger settings in the project options window:

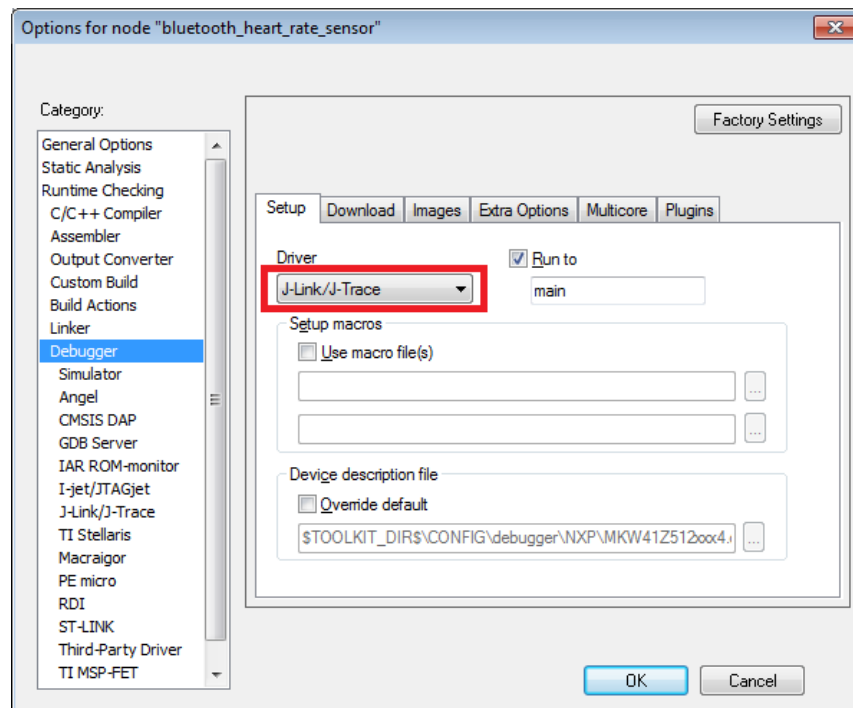


Figure 15: Debugger Settings

Step 6:

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

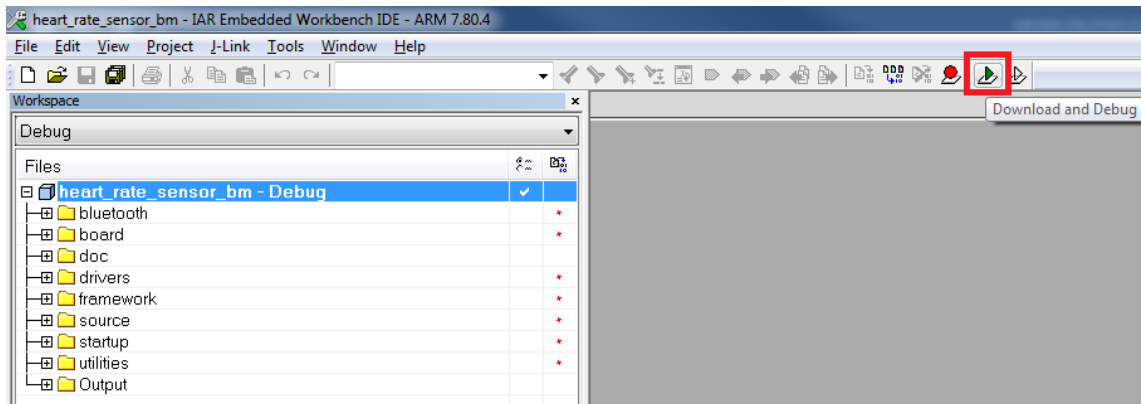


Figure 16: Heart Rate Sensor Download and Debug

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

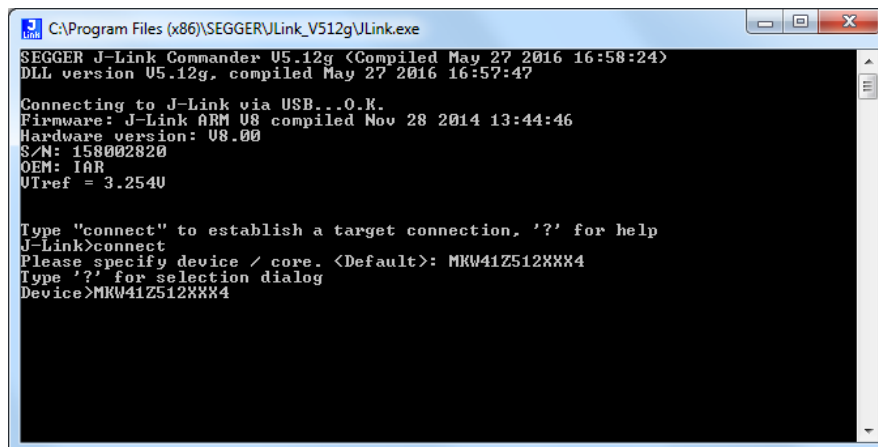
The MKW41Z connectivity software package contains in the `<connectivity_path>\tools\wireless\binaries` folder a series of pre-compiled binary applications that can be flashed onto a development board.

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.



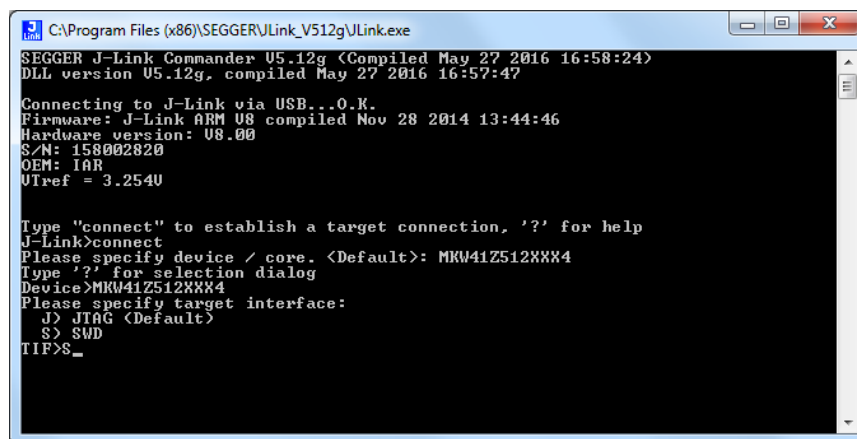
```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
SEGGER J-Link Commander V5.12g (Compiled May 27 2016 16:58:24)
DLL version V5.12g, compiled May 27 2016 16:57:47

Connecting to J-Link via USB...O.K.
Firmware: J-Link ARM V8 compiled Nov 28 2014 13:44:46
Hardware version: V8.00
S/N: 158002820
OEM: IAR
Vtref = 3.254V

Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. (Default): MKW41Z512XXX4
Type '?' for selection dialog
Device>MKW41Z512XXX4
```

Figure 17: MKW41Z512xxx4 device selection

Step 2: Select SWD target interface.



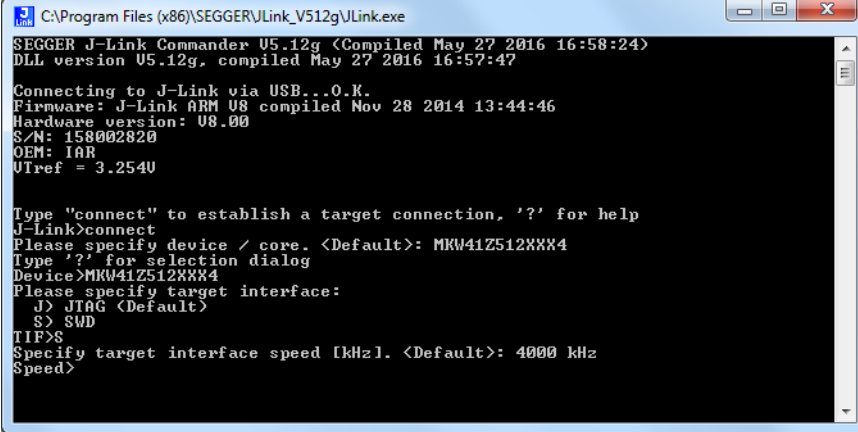
```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
SEGGER J-Link Commander V5.12g (Compiled May 27 2016 16:58:24)
DLL version V5.12g, compiled May 27 2016 16:57:47

Connecting to J-Link via USB...O.K.
Firmware: J-Link ARM V8 compiled Nov 28 2014 13:44:46
Hardware version: V8.00
S/N: 158002820
OEM: IAR
Vtref = 3.254V

Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. (Default): MKW41Z512XXX4
Type '?' for selection dialog
Device>MKW41Z512XXX4
Please specify target interface:
J) JTAG (Default)
S) SWD
TIF>S_
```

Figure 18: SWD interface selection

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

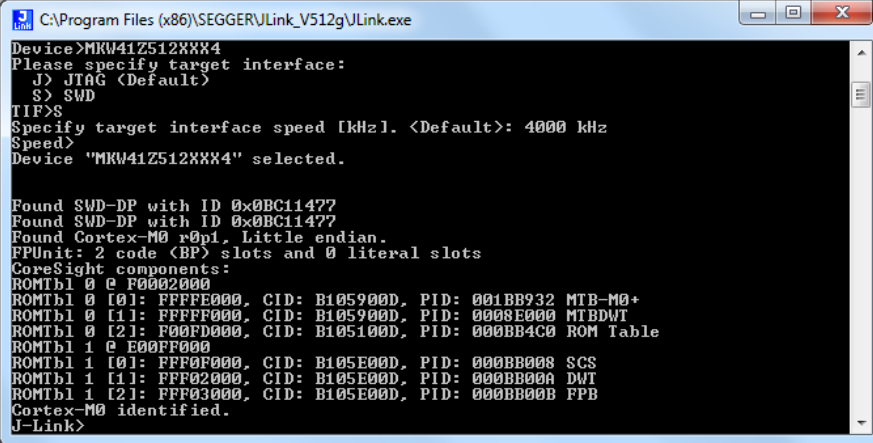


```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
SEGGER J-Link Commander V5.12g [Compiled May 27 2016 16:58:24]
DLL version V5.12g, compiled May 27 2016 16:57:47

Connecting to J-Link via USB...O.K.
Firmware: J-Link ARM V8 compiled Nov 28 2014 13:44:46
Hardware version: V8.00
S/N: 158002820
OEM: IAR
UTref = 3.254V

Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: MKW41Z512XXX4
Type '?' for selection dialog
Device>MKW41Z512XXX4
Please specify target interface:
  J) JTAG <Default>
  S) SWD
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
```

Figure 19: SWD interface speed selection

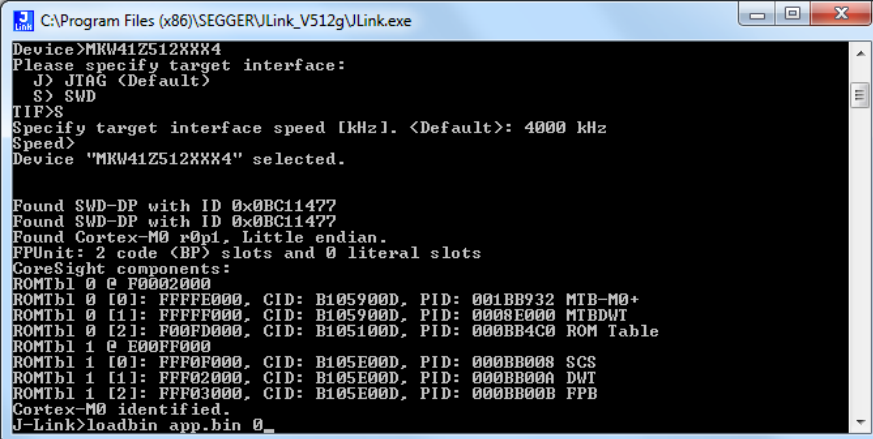


```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
Device>MKW41Z512XXX4
Please specify target interface:
  J) JTAG <Default>
  S) SWD
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
Device "MKW41Z512XXX4" selected.

Found SWD-DP with ID 0x0BC11477
Found SWD-DP with ID 0x0BC11477
Found Cortex-M0 r0pl, Little endian.
FPUnit: 2 code <BP> slots and 0 literal slots
CoreSight components:
ROMTbl 0 @ F0002000
ROMTbl 0 [0]: FFFFE000, CID: B105900D, PID: 001BB932 MTB-M0+
ROMTbl 0 [1]: FFFF0000, CID: B105900D, PID: 0008E000 MTBDWT
ROMTbl 0 [2]: F00FD000, CID: B105100D, PID: 000BB4C0 ROM Table
ROMTbl 1 @ E00FF000
ROMTbl 1 [0]: FFF0F000, CID: B105E00D, PID: 000BB008 SCS
ROMTbl 1 [1]: FFF02000, CID: B105E00D, PID: 000BB00A DWT
ROMTbl 1 [2]: FFF03000, CID: B105E00D, PID: 000BB00B FPB
Cortex-M0 identified.
J-Link>
```

Figure 20: Cortex-M0 identified

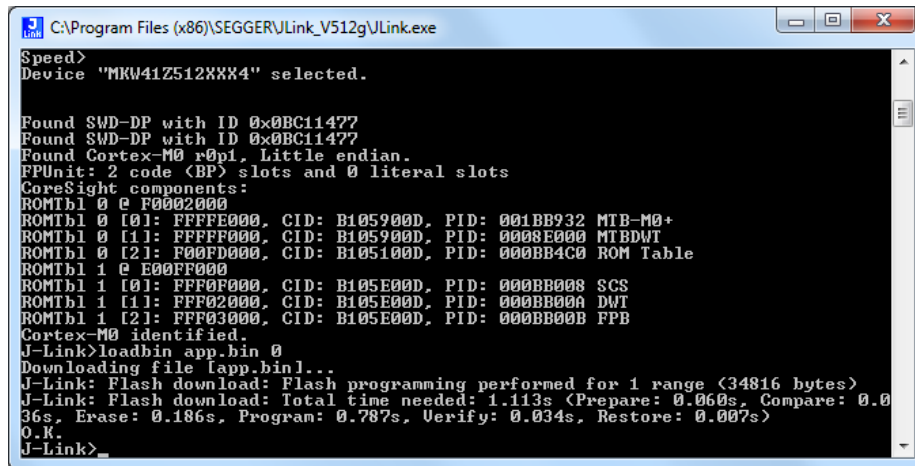
Step 4: Type **loadbin app.bin 0** to flash the binary file (assuming application name is *app.bin*).



```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
Device>MKW41Z512XXX4
Please specify target interface:
  J) JTAG <Default>
  S) SWD
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
Device "MKW41Z512XXX4" selected.

Found SWD-DP with ID 0x0BC11477
Found SWD-DP with ID 0x0BC11477
Found Cortex-M0 r0pl, Little endian.
FPUnit: 2 code <BP> slots and 0 literal slots
CoreSight components:
ROMTbl 0 @ F0002000
ROMTbl 0 [0]: FFFFE000, CID: B105900D, PID: 001BB932 MTB-M0+
ROMTbl 0 [1]: FFFF0000, CID: B105900D, PID: 0008E000 MTBDWT
ROMTbl 0 [2]: F00FD000, CID: B105100D, PID: 000BB4C0 ROM Table
ROMTbl 1 @ E00FF000
ROMTbl 1 [0]: FFF0F000, CID: B105E00D, PID: 000BB008 SCS
ROMTbl 1 [1]: FFF02000, CID: B105E00D, PID: 000BB00A DWT
ROMTbl 1 [2]: FFF03000, CID: B105E00D, PID: 000BB00B FPB
Cortex-M0 identified.
J-Link>loadbin app.bin 0_
```

Figure 21: Load binary file



```
C:\Program Files (x86)\SEGGER\JLink_V512g\JLink.exe
Speed>
Device "MKW41Z512XXXX4" selected.

Found SWD-DP with ID 0x0BBC11477
Found SWD-DP with ID 0x0BBC11477
Found Cortex-M0 r0p1, Little endian.
FPUnit: 2 code <BP> slots and 0 literal slots
CoreSight components:
ROMTbl 0 @ F0002000
ROMTbl 0 [0]: FFFFE000, CID: B105900D, PID: 001BB932 MTB-M0+
ROMTbl 0 [1]: FFFF0000, CID: B105900D, PID: 0008E000 MTBDWT
ROMTbl 0 [2]: F00FD000, CID: B105100D, PID: 000BB4C0 ROM Table
ROMTbl 1 @ E00FF000
ROMTbl 1 [0]: FFF0F000, CID: B105E00D, PID: 000BB008 SCS
ROMTbl 1 [1]: FFF02000, CID: B105E00D, PID: 000BB00A DWT
ROMTbl 1 [2]: FFF03000, CID: B105E00D, PID: 000BB00B FPB
Cortex-M0 identified.
J-Link>loadbin app.bin 0
Downloading file {app.bin}...
J-Link: Flash download: Flash programming performed for 1 range (34816 bytes)
J-Link: Flash download: Total time needed: 1.113s (Prepare: 0.060s, Compare: 0.036s, Erase: 0.186s, Program: 0.787s, Verify: 0.034s, Restore: 0.007s)
O.K.
J-Link>
```

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:

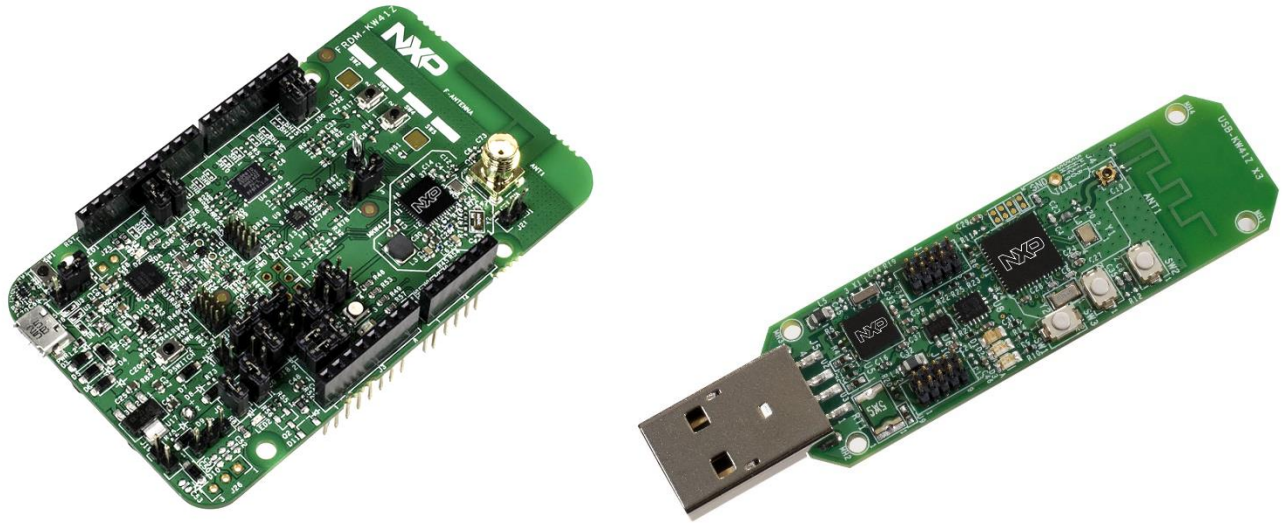


Figure 23: FRDMKW41Z and USBKW41Z

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.

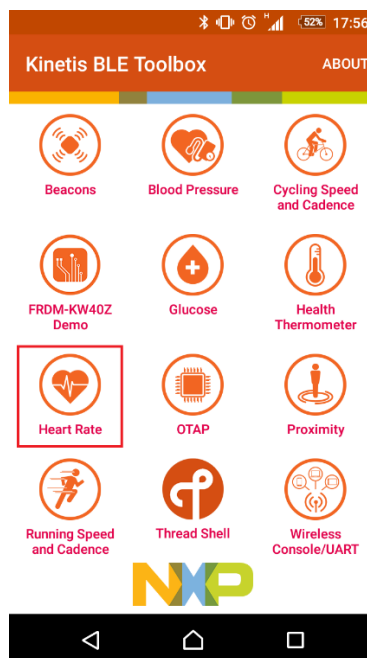


Figure 25: Kinetis BLE Toolbox - opening Heart Rate application

Step 4:

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

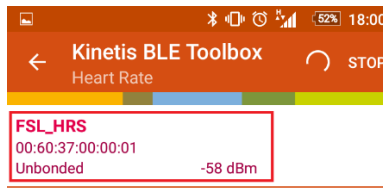


Figure 26: Selecting the scanned device

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.

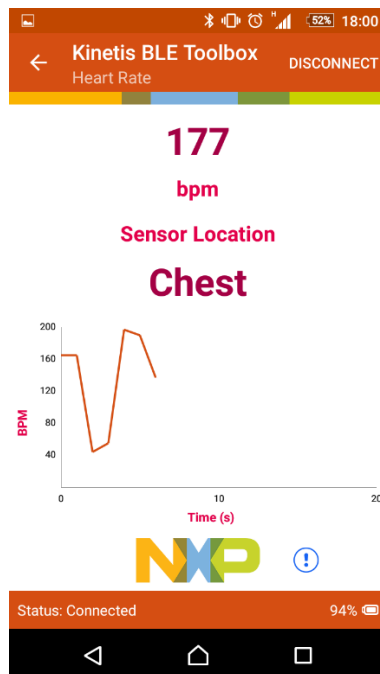


Figure 27: Heart Rate sensor device functionality in connected mode

How to Reach Us:

Home Page:

www.nxp.com

Web Support:

www.nxp.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BLUETOOTH® LOW ENERGY is a trademark of the Institute of Electrical and Electronics Engineers, Inc. (IEEE). This product is not endorsed or approved by the IEEE. All other product or service names are the property of their respective owners. ARM, the ARM powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ZigBee is a registered trademark of ZigBee Alliance, Inc. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document number: MKW41ZBLESW123QSG
Rev. 0
04/2017

