AN12558

Migration Guide from MKW38A512xxx4 to MKW37A512xxx4

Rev. 0 — 04/2020 Application Note

Contents

1 Introduction

This document describes how to migrate from Kinetis MKW38A512xxx4 to MKW37A512xxx4 MCUs, emphasizing the connectivity software. In this document, MKW38A512xxx4 and MKW37A512xxx4 are referred to as MKW38 and MKW37, respectively. The document is intended for software engineers, software testers, software integrators, and customers designing their own hardware.

2 Hardware considerations

The MKW37 wireless MCU is pin-to-pin compatible with MKW38 and almost all peripherals are the same in both devices. The main difference between the MKW38 and MKW37 MCUs is related to the size and addressing of the on-chip flash memory. Table 1 shows the differences:

Table 1. Differences between MKW38 and MKW37

Device	512 KB P-flash	256 KB P-flash + 256 KB FlexNVM	Capable to emulate 8 KB EEPROM	8 KB program acceleration RAM	2nd instance of LPUART	FlexCAN module
MKW38		X	X		X	X
MKW37	X			X		

2.1 Peripherals instantiation

The KW38/37 devices in the 48-pin HVQFN package are pin-to-pin compatible. MKW37 does not support LPUART1 and FlexCAN. Therefore, the signals of the last instances are not available for multiplexing in this device. The **bold** alternatives are only included for the MKW38 MCU.

Table 2. MKW37/38 instance comparative

Pin Name	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
PTA16		PTA16/ LLWU_P4	SPI1_SOUT	**LPUART1_ RTS_b**		TPM0_CH0				
PTA17		PTA17/ LLWU_P5	SPI1_SIN	**LPUART1_ RX**	**CAN0_TX**	TPM_ CLKIN1				

Table continues on the next page...

Table 2. MKW37/38 instance comparative (continued)

PTA18		PTA18/ LLWU_P6	SPI1_SCK	**LPUART1_ TX**	**CAN0_RX**	TPM2_CH0				
PTA19	ADC0_SE5	PTA19/ LLWU_P7	SPI1_PCS0	**LPUART1_ CTS_b**		TPM2_CH1				
PTB0		PTB0/ LLWU_P8/ RF_ RFOSC_ EN		I2C0_SCL	CMP0_OUT	TPM0_CH1		CLKOUT	**CAN0_TX**	
PTB1	ADC0_SE1/ CMP0_IN5	PTB1/ RF_ PRIORITY	DTM_RX	I2C0_SDA	LPTMR0_ ALT1	TPM0_CH2		CMT_IRO	**CAN0_RX**	
PTB3	ADC0_SE2/ CMP0_IN4	PTB3/ ERCLK32K/ RF_ACTIVE	**LPUART1_ RTS_b**	TPM0_CH1	CLKOUT	TPM1_CH1		RTC_ CLKOUT	TPM2_CH1	
PTB16	EXTAL32K	PTB16	**LPUART1_ RX**	I2C1_SCL		TPM2_CH0				
PTB17	XTAL32K	PTB17	**LPUART1_ TX**	I2C1_SDA		TPM2_CH1				
PTB18	ADC0_SE4/ CMP0_IN2	PTB18	**LPUART1_ CTS_b**	I2C1_SCL	TPM_ CLKIN0	TPM0_CH0		NMI_b		
PTC3		PTC3/ LLWU_P11	RX_ SWITCH	I2C1_SDA	LPUART0_ TX	TPM0_CH1		DTM_TX	SPI1_SIN	**CAN0_TX**

Table continues on the next page...

Table 2. MKW37/38 instance comparative (continued)

PTC4		PTC4/ LLWU_P12/ RF_ACTIVE	ANT_A	EXTRG_IN	LPUART0_ CTS_b	TPM1_CH0		I2C0_SCL	SPI1_PCS0	**CAN0_RX**
PTC16		PTC16/ LLWU_P0/ RF_ STATUS	SPI0_SCK	I2C0_SDA	LPUART0_ RTS_b	TPM0_CH3			**LPUART1_ RTS_b**	
PTC17		PTC17/ LLWU_P1/ RF_EXT_ OSC_EN	SPI0_SOUT	I2C1_SCL	LPUART0_ RX			DTM_RX	**LPUART1_ RX**	
PTC18		PTC18/ LLWU_P2	SPI0_SIN	I2C1_SDA	LPUART0_ TX			DTM_TX	**LPUART1_ TX**	
PTC19		PTC19/ LLWU_P3/ RF_ EARLY_ WARNING	SPI0_PCS0	I2C0_SCL	LPUART0_ CTS_b				**LPUART1_ CTS_b**	

NOTE

Table 2 is not a full description of the MKW37/38 pinout. It can be found in the MKW39/38/37 reference manual.

2.2 MCU on-chip memory

2.2.1 MKW38 flash memory

The MKW38 flash is partitioned into:

- One 256-KB Program Flash (P-Flash) array divided into 2-KB sectors. The flash address ranges from 0x0000_0000 to 0x0003_FFFF.
- One 256-KB FlexNVM array divided into 2-KB sectors. The flash address ranges from 0x1000_0000 to 0x1003_FFFF with an alias memory with the address range from 0x0004_0000 to 0x0007_FFFF. With FlexRAM, you can configure the FlexNVM as either basic data flash or EEPROM flash records to support.

Figure 1. KW38 flash memory map

2.2.2 MKW37 flash memory

The MKW37 flash does not support the FlexNVM nor the alias memory. Therefore, the P-Flash array increases to:

- 2x256 KB P-Flash array starting from 0x0000_0000 to 0x0007 FFFF with the address divided into 2-KB sectors.

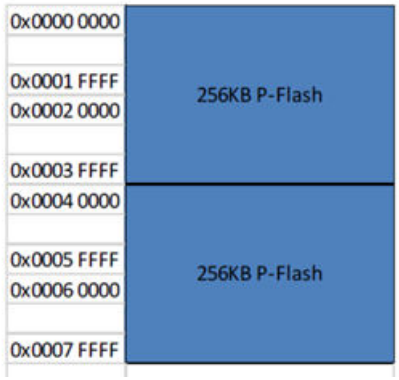

Figure 2. KW37 flash memory map

3 Software Development Kit download and Install

This chapter describes how to download the Software Development Kit (SDK) for the MKW37A512xxx4, because it is used as a starting point to migrate MKW38 code to MKW37 MCUs. The steps to download the SDK package are as follows:

1. Go to the MCUXpresso web page (mcuxpresso.nxp.com).
2. Log in with your registered account.
3. Search for the “KW37A” device. Then click on the suggested processor and click the “Build MCUXpresso SDK” button.

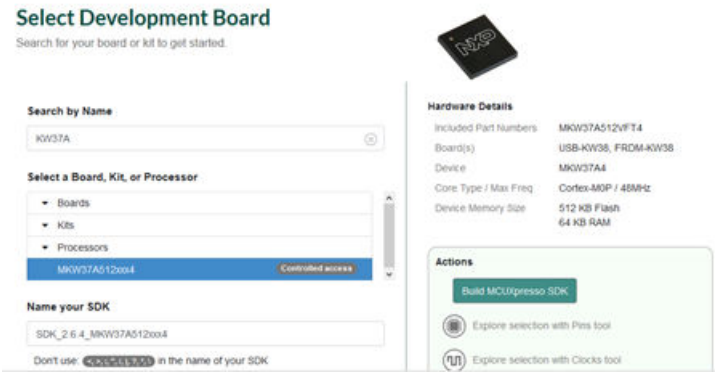

Figure 3. Building MKW37 SDK package

4. The next page appears. Select “All toolchains” in the “Toolchain / IDE” box and provide a name to identify the package.

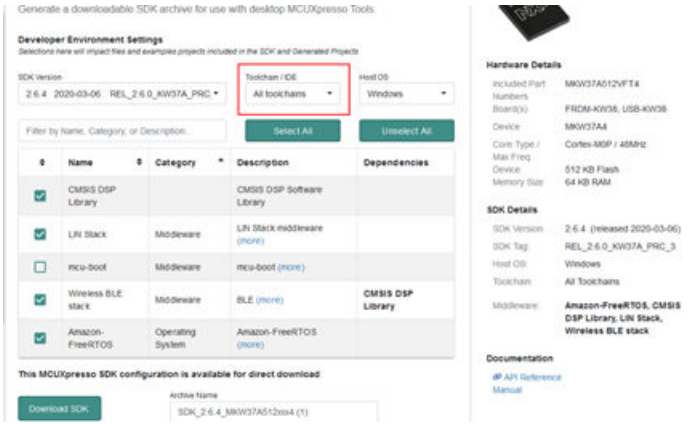

Figure 4. Selecting SDK version

5. Click the “Download SDK” button. This starts the building process of the desired SDK. It takes a few minutes until the system gets the package into your profile in the MCUXpresso web page.
6. When the SDK is ready to download, the “Software Terms and Conditions” are displayed. Accept them and the download process starts automatically.
7. If the download does not start automatically, click the “Download SDK Archive” button.

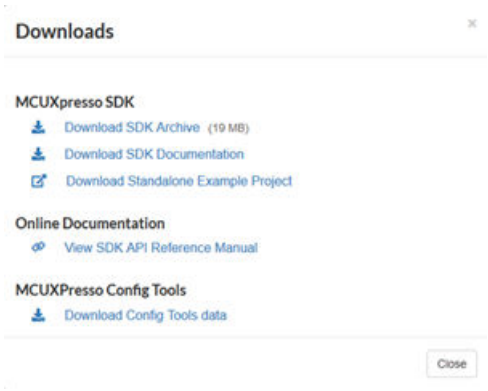

Figure 5. MKW37 SDK displayed in dashboard

8. If the above picture is not displayed, click the “Download SDK archive and documentation” button from “MCUXpresso SDK Dashboard”.

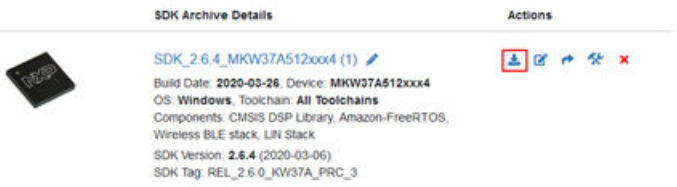

Figure 6. MKW37 SDK displayed in dashboard

At this point, you have downloaded and installed the SDK package for MKW37A devices.

4 Software migration in IAR Embedded Workbench IDE

This chapter describes how to migrate MKW38 example code to MKW37 MCUs in the IAR Embedded Workbench IDE. The Heart Rate Sensor project is used as a base in this document, because it has easy-to-understand examples and involves the Bluetooth LE connectivity software stack (included in the SDK).

4.1 Changes required in project options and settings

NOTE

In this section, a “bare-metal” version of the project is used. However, the same steps apply for FreeRTOS projects. Some paths related to “bare-metal” projects may differ if using FreeRTOS versions.

1. Open the heart rate sensor project located at the following path:

   <SDK_root>/boards/frdmkw38/wireless_examples/bluetooth/hrs/bm/ iar/heart_rate_sensor_bm.eww

2. Select the project in the workspace and press Alt + F7 to open project options.

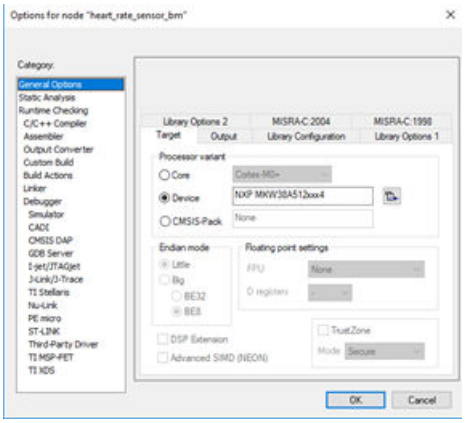

Figure 7. Project options window

3. In the "General Options/Target" window, click the icon next to the device name and select the appropriate device (NXP->KinetisKW->KW3x ->NXP MKW37A512xxx4). Click the "OK" button.

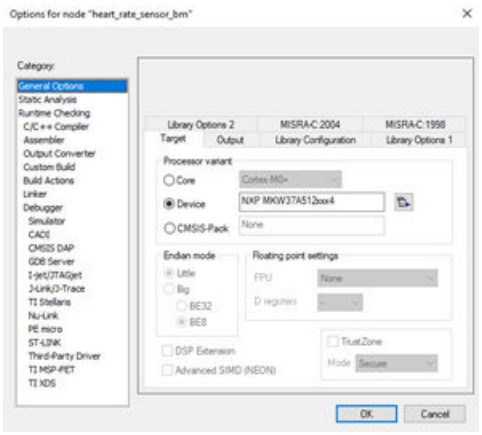

Figure 8. Selecting MKW37 device

4. Create a new folder named *MKW37A4* at the following paths:

   <SDK_root>/middleware/wireless/framework/DCDC/Interface

   <SDK_root>/middleware/wireless/framework/DCDC/Source

   <SDK_root>/middleware/wireless/framework/LowPower/Interface

   <SDK_root>/middleware/wireless/framework/LowPower/Source

   <SDK_root>/middleware/wireless/framework/XCVR

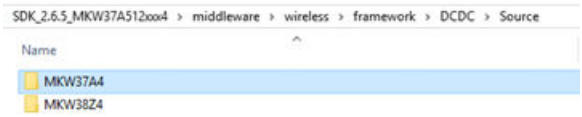

Figure 9. Creating MKW37A4 folders

5. Copy all files inside the *MKW38A4* folders located at the above mentioned paths and paste them into the *MKW37A4* folders.

Figure 10. Adding MKW3A4 files

6. Select the heart rate sensor in the workspace and press Alt+F7 to open the "Project Options" window. In the "C/C++ Compiler/Preprocessor" window, rename the paths related to the *MKW38A/Z4* folders to the *MKW37A4* folders, modifying the name in the additional "Include Directories" text box. The results must look as follows:

   $PROJ_DIR$/../../../../../../../devices/MKW37A4

   $PROJ_DIR$/../../../../../../../devices/MKW37A4/utilities

   $PROJ_DIR$/../../../../../../../devices/MKW37A4/drivers

   $PROJ_DIR$/../../../../../../../middleware/wireless/framework/LowPower/Interface/MKW37A4

   $PROJ_DIR$/../../../../../../../devices/MKW37A4/utilities/str

   $PROJ_DIR$/../../../../../../../devices/MKW37A4/utilities/debug_console

   $PROJ_DIR$/../../../../../../../middleware/wireless/framework/DCDC/Interface/MKW37A4

$PROJ_DIR$/../../../../../../../middleware/wireless/framework/XCVR/MKW37A4/drv

$PROJ_DIR$/../../../../../../../middleware/wireless/framework/XCVR/MKW37A4/drv/nb2p4ghz

$PROJ_DIR$/../../../../../../../middleware/wireless/framework/XCVR/MKW37A4/drv/nb2p4ghz/configs/gen35

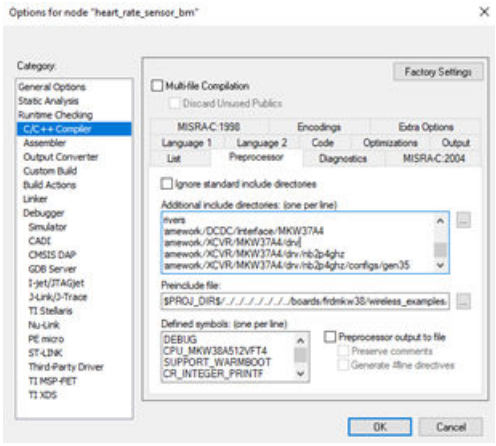

Figure 11. Updating the MKW37 files

7. Rename the CPU_MKW38A512VFT4 macro to CPU_MKW37A512VFT4 and delete the FRDM_KW38 macro in the "Defined Symbols" text box. Click the "OK" button.

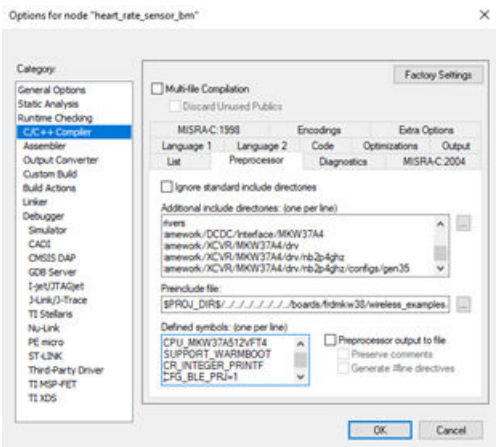

Figure 12. Defining MKW37 macros

8. Expand the startup folder, select all files, right-click, and select the “Remove” option. Right-click the folder and select “Add/Add files”. Add the *startup_MKW37A4.s* file located at this path:

   <SDK_root>/devices/MKW37A4/iar/startup_MKW37A4.s

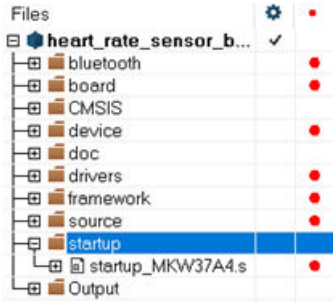

Figure 13. Replacing startup system file

9. Expand the *device* folder, select all files, right-click, and select the “Remove” option. Right-click the folder and select “Add/Add files”. Add the *fsl_device_registers.h, MKW37A4.h, MKW37A4_features.h, system_MKW37A4.h*, and *system_MKW37A4.c* files at the following path:

   <SDK_root>/devices/MKW37A4

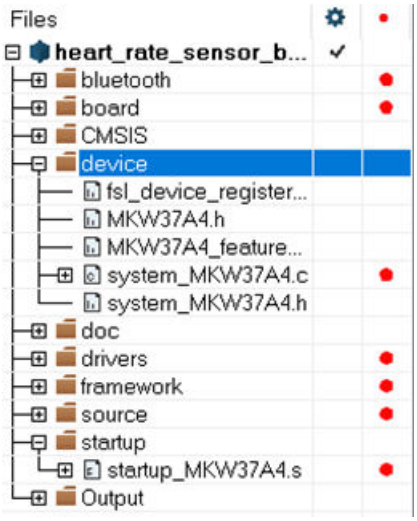

Figure 14. Replacing device files

10. Expand the drivers folder, select the *fsl_flexcan.h* and *fsl_flexcan.c* files, right-click, and select the “Remove” option.
11. Some files must be updated. Open the *clock_config.c* file in the *board* folder. Delete the CLOCK_SetLpuart1Clock call to function in the BOARD_BootClockRUN function.

```
CLOCK_SetLpuart0Clock(SIM_LPUART_CLK_SEL_OSCERCLK_CLK);
/* Set LPUART1 clock source. */
// Delete this line: CLOCK_SetLpuart1Clock(SIM_LPUART_CLK_SEL_OSCERCLK_CLK);
/* Set TPM clock source. */
CLOCK_SetTpmClock(SIM_TPM_CLK_SEL_OSCERCLK_CLK);
```

Figure 15. Updating clock_config.c file

12. If necessary, open the *app_preinclude.h* file in the *source* directory in the workspace. This file contains important information about the board, such as the number of switches and LEDs, timers, power consumption settings, and so on. Examine and update this file to fit into your own custom board.

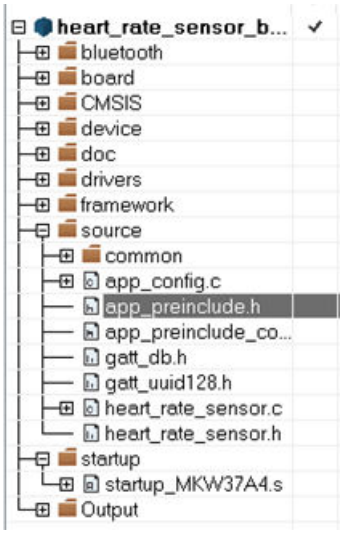

Figure 16. Location of preinclude file

13. If necessary, open the pin mux files (*pin_mux.c* and *pin_mux.h*) and the gpio files (*gpio_pins.c* and *gpio_pins.h*) in the *board* directory in the workspace. These files contain alternatives, options, and multiplexing information of the pins. Examine and update them to fit into your own custom board.

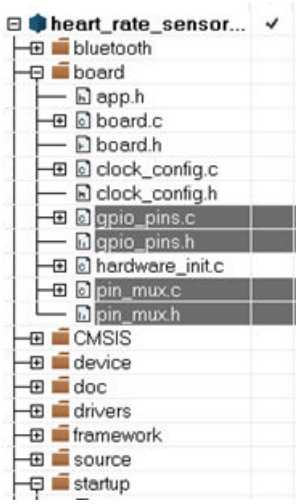

Figure 17. Location of pin mux and gpio files

14. Select the project in the workspace, right-click, and click "clean".
15. Rebuild the project. At this point, the project is already migrated and you can customize the application for your own purpose using MKW37 devices.

5 Software migration in MCUXpresso IDE

This chapter shows how to migrate MKW38 example code to MKW37 MCUs in the MCUXpresso IDE. The Heart Rate Sensor project is used as a base because it has an easy-to-understand example and involves the Bluetooth LE connectivity software stack (included in the SDK).

5.1 Changes required in project properties

NOTE

In this section, a “bare-metal” version of the project is used. However, the same steps apply for the FreeRTOS projects. Some paths related to the “bare-metal” projects may differ if using the FreeRTOS versions.

1. Import the KW39 SDK into the MCUXPresso IDE:
   a. Open the MCUXpresso IDE.
   b. Go to the “Installed SDKs" tab and drag and drop the KW37 SDK.

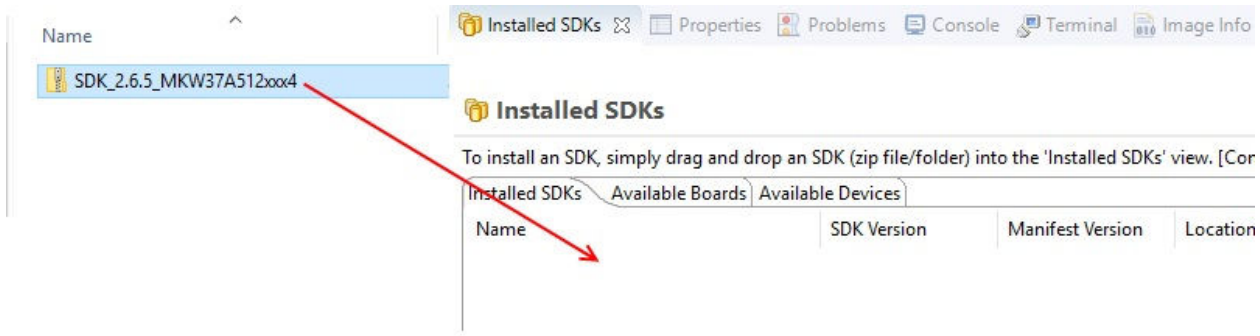

Figure 18. Importing SDK into MCUXpresso IDE

2. Find the "Quickstart Panel" in the lower left-hand corner.

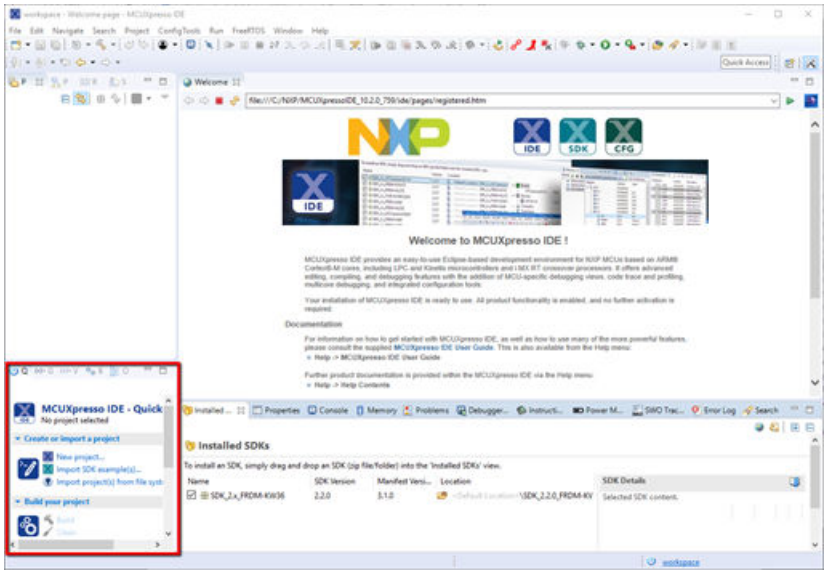

Figure 19. Quickstart Panel perspective in MCUXpresso IDE

3. Click the “Import SDK examples(s)…” option.

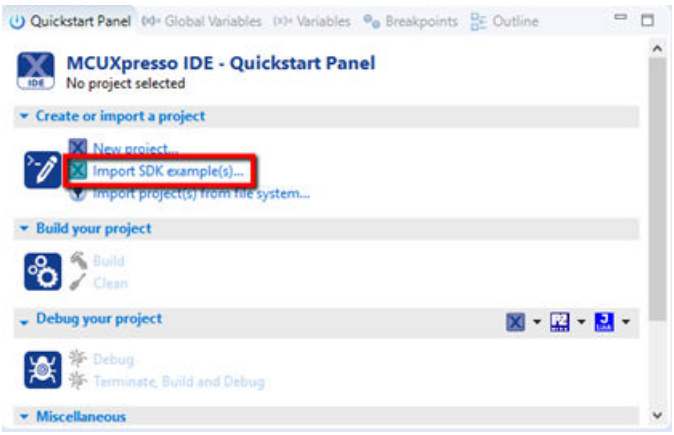

Figure 20. Importing an SDK example

4. Click the "frdmkw38" board and then click "Next".

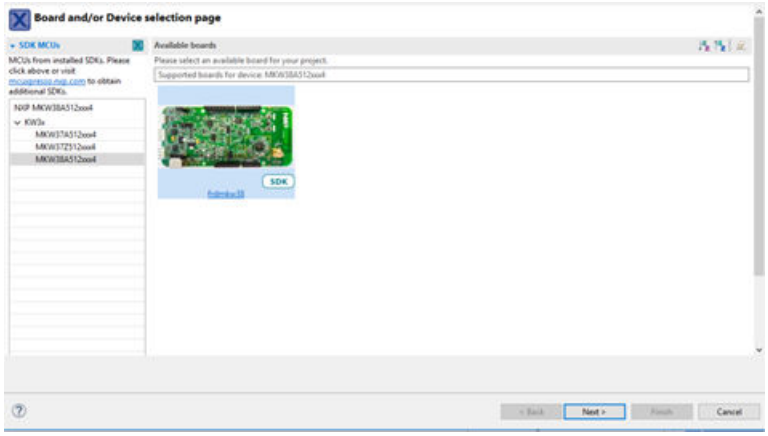

Figure 21. Select FRDM-KW38 board

5. Use the arrow button to expand the list and locate the “hrs” project (wireless_examples -> bluetooth -> hrs) and select the "bm" version of the project. In the “SDK Debug Console”, select “UART” and click “Finish”.

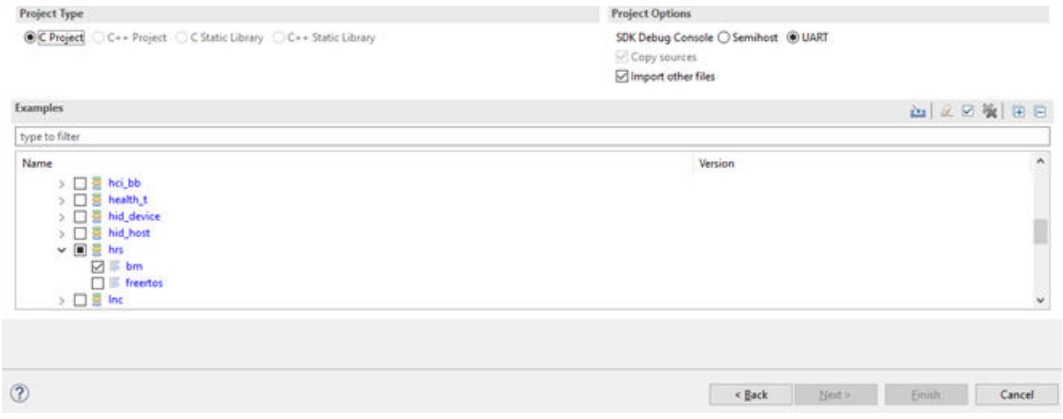

Figure 22. Opening heart rate sensor project

6. Go to "Project/Properties". Expand the "C/C++ Build/MCU" settings and select the "MKW37A512xxx4" MCU. Click the "Apply and Close" button to save the configuration.

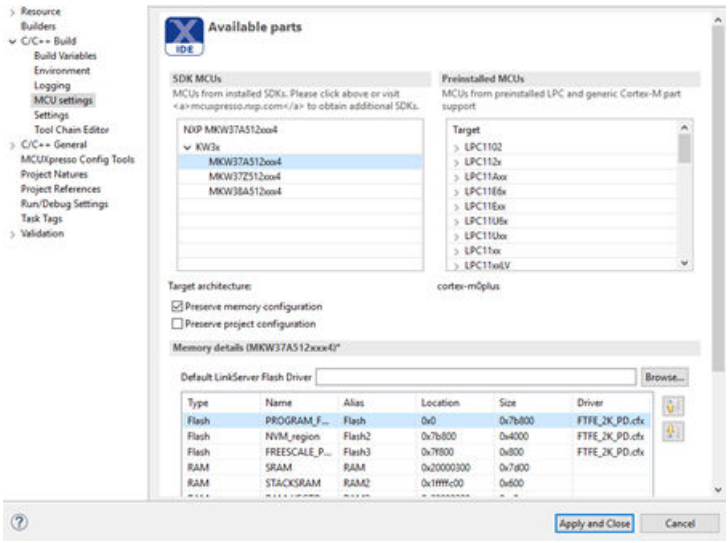

Figure 23. Selecting MKW37 MCU

7. Rename the *MKW38Z4* folder to *MKW37A4* at the following path (Default MCUXpresso root in Documents/ MCUXpressoIDE<version>/workspace):

   <MCUXpresso_hrs_project_root>/framework/XCVR

8. Select the heart rate sensor project in the workspace and press F5 to refresh the modified folders.

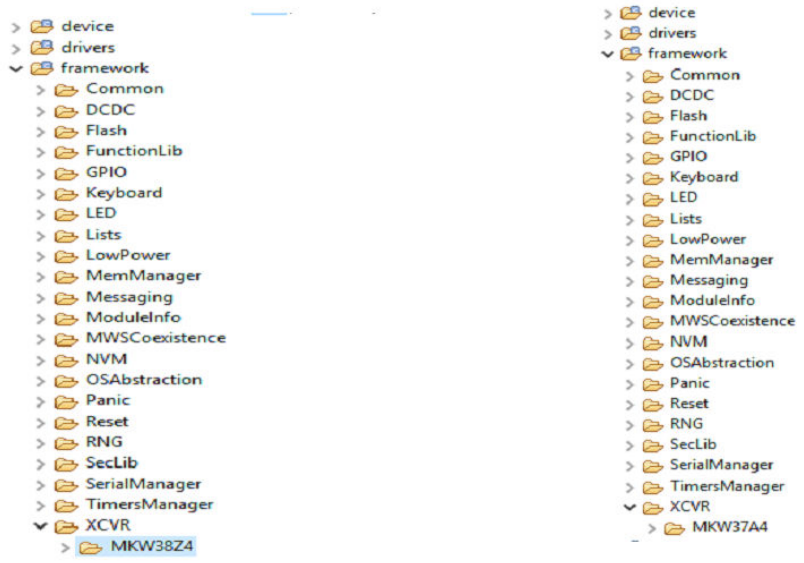

Figure 24. Updating project folders in the workspace

9. Open the "Project/Properties" window in the MCUXpresso IDE. Go to "C/C++ Build/Settings" and select the *MCU C Compliler/Includes* folder in the "Tool Settings" window. Edit all paths related to the MKW38 MCU according to the *MKW37* folders created before. The results must look as follows:

   ../framework/XCVR/MKW37A4

   ../framework/XCVR/MKW37A4/nb2p4ghz/configs

   ../framework/XCVR/MKW37A4/nb2p4ghz

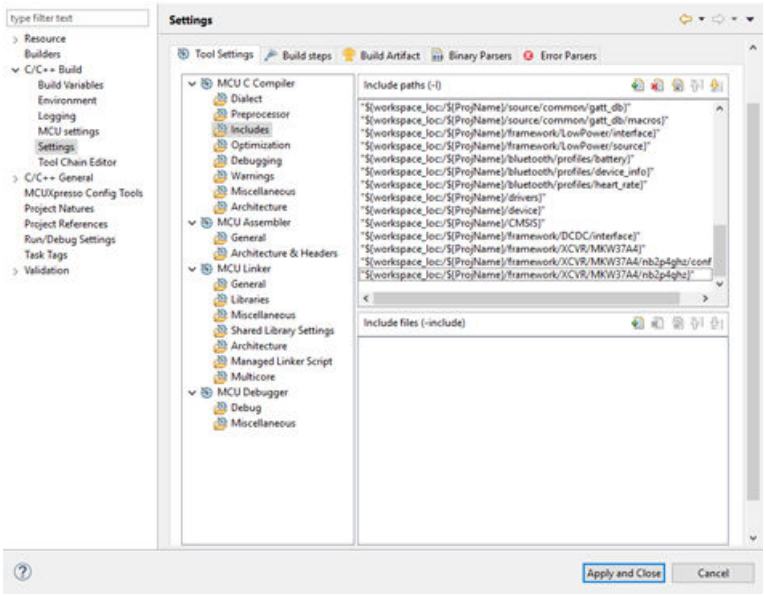

Figure 25. Updating include paths

10. Select the *MCU Assembler/General* folder in "Tool Settings". Edit the paths related to the MKW38 MCU. The results must look as follows:

    ../framework/XCVR/MKW37A4

    ../framework/XCVR/MKW37A4/nb2p4ghz/configs

    ../framework/XCVR/MKW37A4/nb2p4ghz

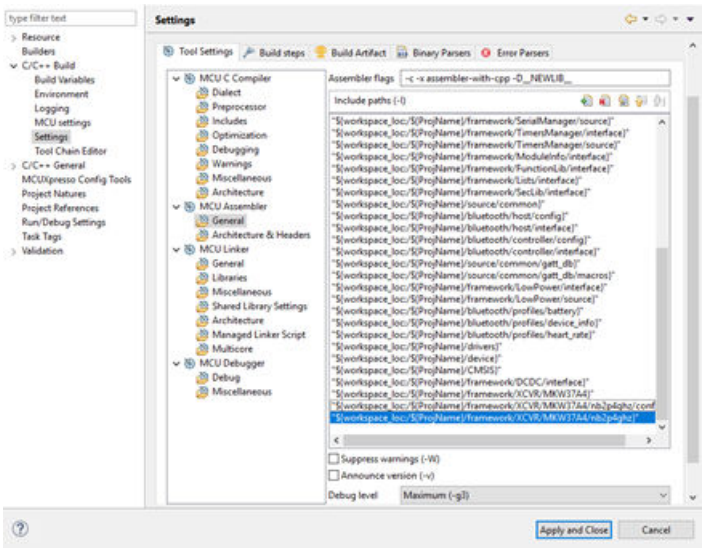

Figure 26. Updating assembler paths

11. Open the *MCU C Compiler/Preprocessor* window, edit the CPU_MKW38A512VFT4 and CPU_MKW38A512VFT4_cm0plus macros to CPU_MKW37A512VFT4 and CPU_MKW37A512VFT4_cm0plus, respectively. Delete the FRDM_KW38 macro. Then click the "Apply and Close" button.

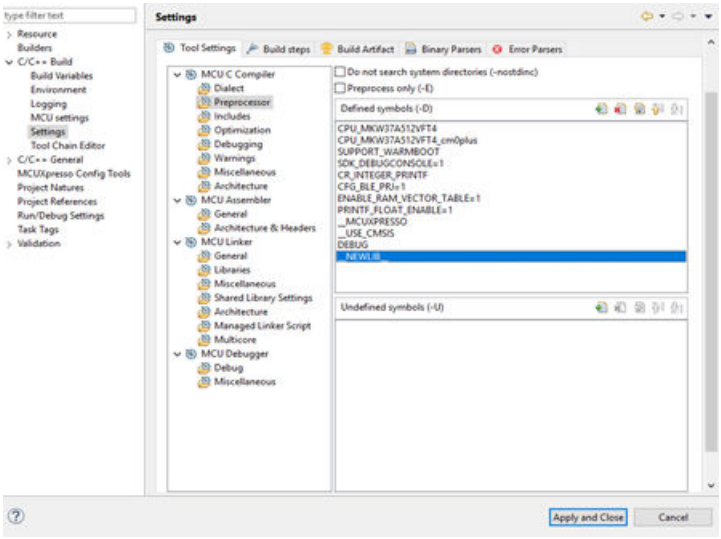

Figure 27. Updating preprocessor macros

12. Delete the *fsl_device_registers.h* , *MKW38A4.h* , *MKW38A4_features.h* , *system_MKW38A4.h*, and *system_MKW38A4c.* files located in the *device* folder in the workspace.
13. Unzip the MKW37A4 SDK package and search for the *fsl_device_registers.h*, *MKW37A4.h*, *MKW37A4_features.h,system_MKW37A4.h*, and *system_MKW37A4.c* files in this folder at the following paths:

    <SDK_folder_root>/devices/MKW37A4/fsl_device_registers.h

    <SDK_folder_root>/devices/MKW37A4/MKW37A4.h

    <SDK_folder_root>/devices/MKW37A4/MKW37A4_features.h

    <SDK_folder_root>/devices/MKW37A4/system_MKW37A4.h

    <SDK_folder_root>/devices/MKW37A4/system_MKW37A4.c

    Drag and drop these files into the *device* folder in the workspace.

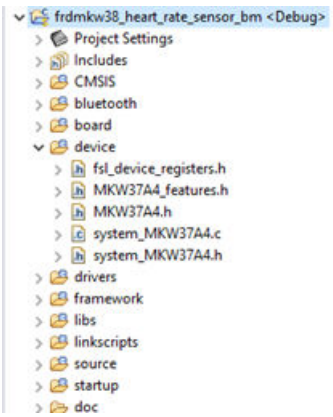

Figure 28. Updating device files

14. Replace the *startup_mkw38a4.c* file located in the *startup* folder in the workspace to *startup_mkw37a4.c*. This file is located in the MKW37A SDK at the following path:

    <SDK_folder_root>/devices/MKW37A4/mcuxpresso/startup_mkw37a4.c

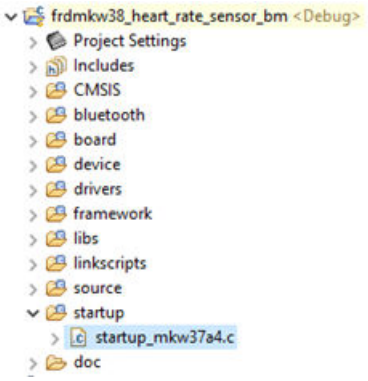

Figure 29. Updating the startup file

15. Some files must be updated. Open the *clock_config.c* file in the *board* folder. Delete the CLOCK_SetLpuart1Clock call to a function in the BOARD_BootClockRUN function.

```
/* Set LPUART0 clock source. */
CLOCK_SetLpuart0Clock(SIM_LPUART_CLK_SEL_OSCERCLK_CLK);
/* Set LPUART1 clock source. */
//Delete this line: CLOCK_SetLpuart1Clock(SIM_LPUART_CLK_SEL_OSCERCLK_CLK);
/* Set TPM clock source. */
CLOCK_SetTpmClock(SIM_TPM_CLK_SEL_OSCERCLK_CLK);
```

Figure 30. Updating clock_config.c file

16. If necessary, open the *app_preinclude.h* file under the source directory in the workspace. This file contains important information about the board, such as the number of switches and LEDs, timers, power consumption settings, and so on. Examine and update this file to fit into your own custom board.

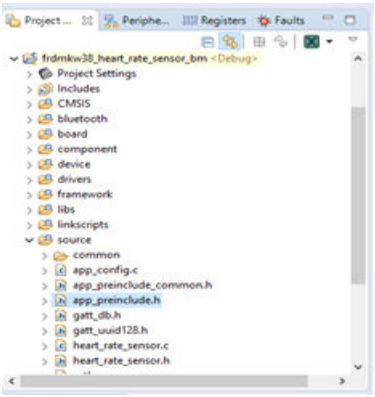

Figure 31. Location of preinclude file

17. If necessary, open the pin mux files (*pin_mux.c* and *pin_mux.h*) and the gpio files (*gpio_pins.c* and *gpio_pins.h*) in the *board* directory in the workspace. These files contain alternatives, options, and multiplexing information of the pins. Examine and update them to fit into your own custom board.

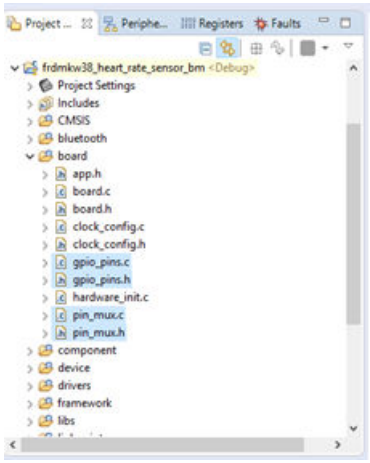

Figure 32. Location of pin mux and gpio files

18. Expand the *drivers* folder, select the *fsl_flexcan.h* and *fsl_flexcan.c* files, right-click and select the “Delete” option.
19. Build the project. At this point, the project is already migrated and you are ready to run the demo using MKW37.

6 Build and run Bluetooth LE connectivity stack examples

All the examples referenced in the *Bluetooth LE Demo Applications User’s Guide* are compatible with the MKW37 device, following the modifications described in this document.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 04/2020

Document identifier: AN12558

arm