

Migration Guide from MKW36Z512xxx4 to MKW35Z512xxx4

1. Introduction

This document describes how to migrate from Kinetis MKW36Z512xxx4 to MKW35Z512xxx4 microcontrollers emphasizing in connectivity software. In this document MKW36Z512xxx4 and MKW35Z512xxx4 are referred as MKW36 and MKW35 respectively. The document is intended for software engineers, software testers, software integrators, and customers designing their own hardware.

Contents

| | | |
|------|---|----|
| 1. | Introduction..... | 1 |
| 2. | Hardware Considerations..... | 2 |
| 2.1. | Peripherals Instantiation..... | 2 |
| 2.2. | MCU On-chip Memory..... | 3 |
| 3. | Software Development Kit Download and Install..... | 5 |
| 4. | Software Migration in IAR Embedded Workbench..... | 8 |
| 4.1. | Changes Required in Project Options and Settings..... | 8 |
| 5. | Software Migration in MCUXpresso IDE..... | 16 |
| 5.1. | Changes Required in Project Properties..... | 16 |
| 6. | Build and Run Bluetooth LE Connectivity Stack Examples..... | 26 |

2. Hardware Considerations

The MKW35 wireless MCU is pin to pin compatible with MKW36, and almost all peripherals are in accord in both devices. The main difference between MKW36 and MKW35 microcontrollers is related to the size and addressing of the On-chip Flash Memory. The [Table 1](#) shows the differences:

Table 1. Differences between MKW36 and MKW35

| Device | 512 KB P-Flash | 256 KB P-Flash + 256 KB FlexNVM | Capable to emulate 8 KB EEPROM | 2 nd instance of LPUART | FlexCAN module |
|--------|----------------|---------------------------------|--------------------------------|------------------------------------|----------------|
| MKW36 | | X | X | X | X |
| MKW35 | X | | | | |

2.1. Peripherals Instantiation

KW36/35 devices are pin to pin compatible for both available packages (40-pin QFN and 48-pin LQFN). MKW35 does not support LPUART1 and FlexCAN, therefore, the signals of the last instances are not available for multiplexing in this device. **Bold** alternatives are only included for MKW36 microcontroller.

Table 2. MKW35/36 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/ RF_RESET | SPI1_SIN | LPUART1_ RX | CAN0_TX | TPM_ CLKIN1 | | | | |
| 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/ RF_RESET | | 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 | LPUART1_ RTS_b | | CLKOUT | TPM1_CH1 | | RTC_ CLKOUT | | |
| PTB16 | EXTAL32K | PTB16 | LPUART1_ RX | I2C1_SCL | | TPM2_CH0 | | | | |
| PTB17 | XTAL32K | PTB17 | LPUART1_ TX | I2C1_SDA | | TPM2_CH1 | | BSM_CLK | | |
| 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 |
| PTC4 | | PTC4/ LLWU_P12/ BLE_RF_ ACTIVE | | EXTRG_IN | LPUART0_ CTS_b | TPM1_CH0 | | BSM_DATA | 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 | BSM_ FRAME | | DTM_RX | | LPUART1_ RX |
| PTC18 | | PTC18/ LLWU_P2 | SPI0_SIN | I2C1_SDA | LPUART0_ TX | BSM_DATA | | DTM_TX | | LPUART1_ TX |
| PTC19 | | PTC19/ LLWU_P3/ RF_ EARLY_ WARNING | SPI0_PCS0 | I2C0_SCL | LPUART0_ CTS_b | BSM_CLK | | | | LPUART1_ CTS_b |

NOTE

Table 2 is not a full description of the MKW35/36 pinout.

NOTE

There is a change in the number of available digital pins between 40-pin and 48-pin packages due to different number of pins. For example, in the 48-pin package, there is a total of 25 digital pins while in the 40-pin package, 18 digital pins are available.

2.2. MCU On-chip Memory

2.2.1 MKW36 Flash Memory

Migration Guide from MKW36Z512xxx4 to MKW35Z512xxx4, User's Guide, Rev. 1, 06/2019

MKW36 Flash is partitioned into:

- One 256 KB Program Flash (P-Flash) array divided into 2 KB sectors. Flash address range from 0x0000_0000 to 0x0003_FFFF.
- One 256 KB FlexNVM array divided in 2 KB sectors, flash address range from 0x1000_0000 to 0x1003_FFFF with an Alias memory with address range 0x0004_0000 to 0x0007_FFFF. FlexNVM memory can be used as Program Flash (P-Flash), Data Flash (D-Flash) and EEPROM Flash.

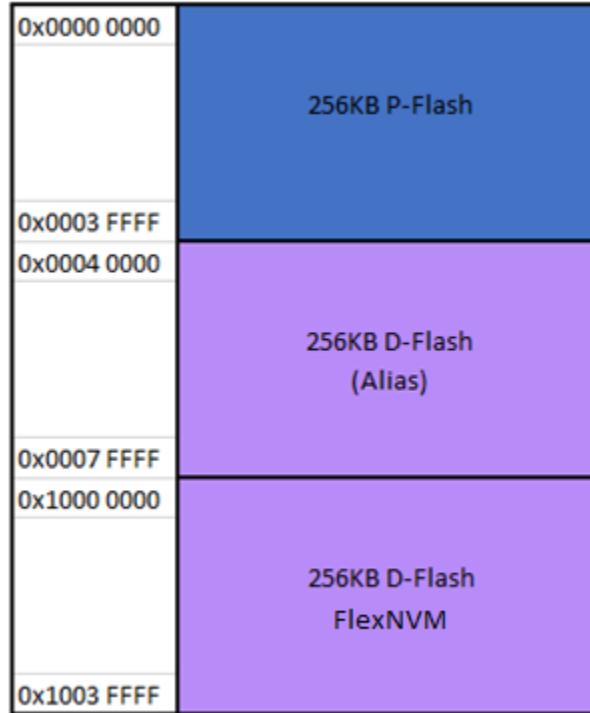


Figure 1. KW36 Flash Memory Map

2.2.2 MKW35 Flash Memory

MKW35 Flash does not support FlexNVM nor Alias memory, therefore, Program Flash array increase to:

- 2x256 KB Program Flash (P-Flash) array starting from 0x0000_0000 to 0x0007_FFFF address divided into 2 KB sectors.

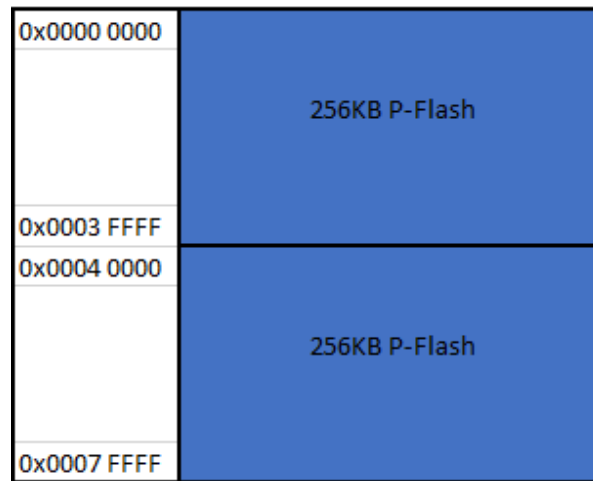


Figure 2. KW35 Flash Memory Map

3. Software Development Kit Download and Install

This chapter provides all the steps needed to download the SDK (Software Development Kit) for MKW35Z512xxx4 due it is used as starting point to migrate a MKW36 code to MKW35 MCU. Below are the steps to download the SDK package:

1. Go to MCUXpresso web page (mcuxpresso.nxp.com).
2. Log in with your registered account.
3. Search for “KW35Z” device. Then click on the suggested processor and click on “Build MCUXpresso SDK”.

Select Development Board

Search for your board or kit to get started.

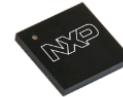
Search by Name

Select a Device, Board, or Kit

- ▼ Boards
- ▼ Kits
- ▼ Processors
- MKW35Z512xxx4

Name your SDK

Don't use: < > * / : ; ' " \ | in the name of your SDK



Hardware Details

| | |
|-----------------------|---------------------------|
| Included Part Numbers | MKW35Z512VHT4 |
| Board(s) | FRDM-KW36 |
| Device | MKW35Z4 |
| Core Type / Max Freq | Cortex-M0P / 48MHz |
| Memory Size | 512 KB Flash 64 KB RAM |

Actions

- [Build MCUXpresso SDK](#)
- Explore selection with Clocks tool
- Explore selection with Pins tool

Figure 3. Building MKW35 SDK package

- The next page is displayed. Select “All toolchains” in the “Toolchain / IDE” box and provide name to identify the package.

SDK Builder

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

Developer Environment Settings

Selections here will impact files and examples projects included in the SDK and Generated Projects

Host OS:

Toolchain / IDE:

Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

[Add software component](#)

Click the link below to request this specific MCUXpresso SDK Build

In general, SDK builds should complete within a few minutes. You will be notified via email and notifications in the upper right corner of this webpage.



Hardware Details

| | |
|-----------------------|---------------------------|
| Included Part Numbers | MKW35Z512VHT4 |
| Board(s) | FRDM-KW36 |
| Device | MKW35Z4 |
| Core Type / Max Freq | Cortex-M0P / 48MHz |
| Memory Size | 512 KB Flash 64 KB RAM |

SDK Details

| | |
|--------------|---|
| SDK Version: | KSDK 2.2.0 (released 2018-09-11) |
| SDK Tag: | release_conn_ksdk_2.2_kw35a_1.3.3_RFP_RC3.1 |
| Host OS: | Windows |
| Toolchain: | All Toolchains |
| Middleware: | FatFS, FreeRTOS, BLE, GenFSK |

Figure 4. Selecting version of SDK

- Click on the “Request Build” button. This will start the building process of the desired SDK. It will take few minutes until the system get the package into your profile in MCUXpresso web page.
- Once the SDK is ready to download, it is displayed in the MCUXpresso Dashboard as shown in [Figure 5](#)



Figure 5. MKW35 SDK displayed in Dashboard

7. Click on download icon as shown in [Figure 5](#) A new window appears to download the package.

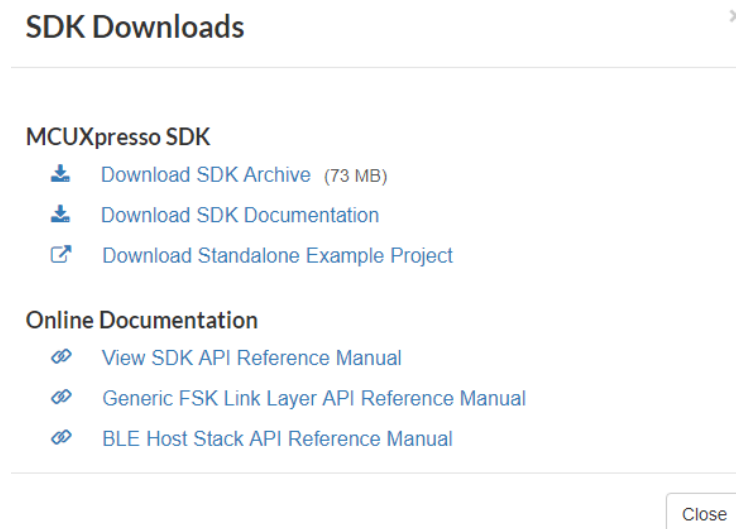


Figure 6. MKW35 SDK displayed in Dashboard

8. Accept the license agreement and save the SDK in your desired location.
9. If MCUXpresso IDE is used, drag and drop the MKW35 SDK zip folder in “Installed SDK’s” perspective to install the package.

At this point, you have downloaded and installed the SDK package for MKW35Z device.

4. Software Migration in IAR Embedded Workbench

This chapter provides all the steps needed to migrate a MKW36 example code to MKW35 MCU in IAR Embedded Workbench IDE. The Heart Rate Sensor project is used as base in this document because it has easy to understand example and involves the Bluetooth LE connectivity software stack (included in SDK).

4.1. Changes Required in Project Options and Settings

NOTE

In this section “bare metal” version of the project was used, however, the same steps apply for “freertos” projects. Some paths related to “bm” projects may differ if “freertos” versions were used.

1. Open the heart rate sensor project located at the following path:
`<SDK_root>/boards/frdmkw36/wireless_examples/bluetooth/hrs/bm/iar/hrs_bm.eww`
2. Select the project in the workspace and press Alt + F7 to open project options.

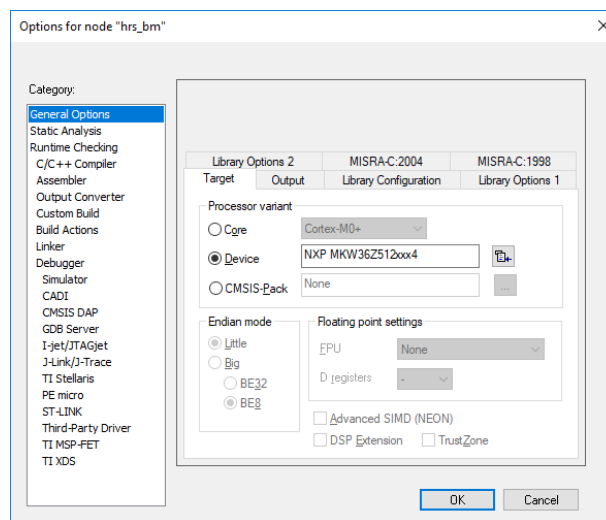


Figure 7. Project options window

3. In the General Options/Target window click the icon next to the device name and select the appropriate device NXP/KinetisKW/KW3x/NXP MKW35Z512xxx4, then click the OK button.

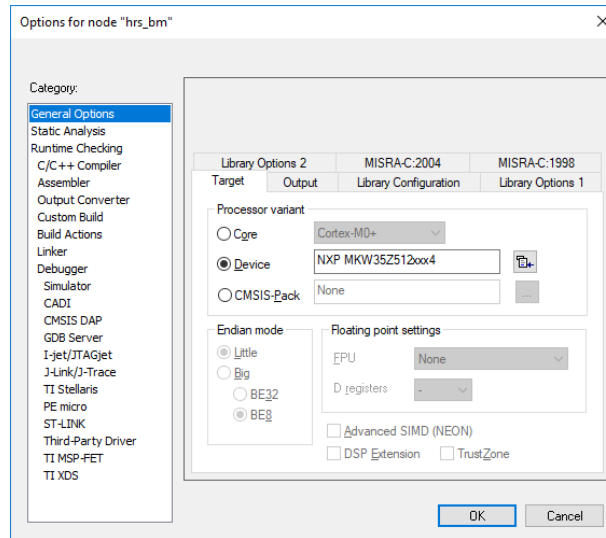


Figure 8. Selecting MKW35 device

4. Create a new folder with the name MKW35Z at following paths:

<SDK_root>/middleware/wireless/framework_5.4.2/DCDC/Interface

<SDK_root>/middleware/wireless/framework_5.4.2/DCDC/Source

<SDK_root>/middleware/wireless/framework_5.4.2/LowPower/Interface

<SDK_root>/middleware/wireless/framework_5.4.2/LowPower/Source

<SDK_root>/middleware/wireless/framework_5.4.2/XCVR

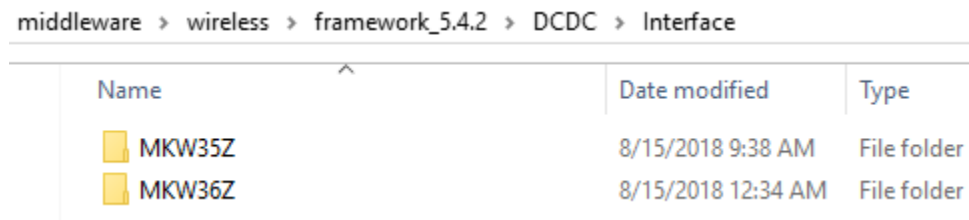


Figure 9. Creating MKW35Z folders

5. Copy all files inside MKW36Z folders located at the above mentioned paths and paste in MKW35Z folders.

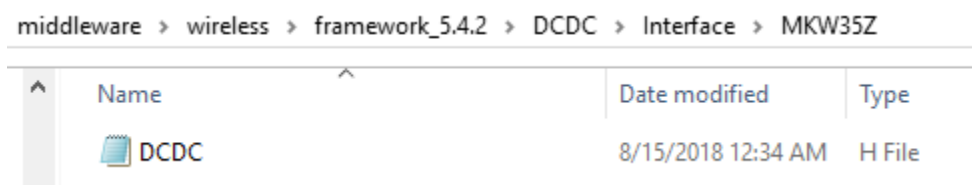


Figure 10. Creating the MKW35 files

Migration Guide from MKW36Z512xxx4 to MKW35Z512xxx4, User's Guide, Rev. 1, 06/2019

6. Select the heart rate sensor in the workspace and press Alt+F7 to open project options window. In C/C++ Compiler/Preprocessor window, rename the paths related to MKW36Z folders to MKW35Z folders modifying a 5 number instead of 6 in the additional include directories text box. The results must look similar as shown below:

```
$PROJ_DIR$/middleware/wireless/framework_5.4.2/LowPower/Interface/MKW35Z
$PROJ_DIR$/../../../../../../../../devices/MKW35Z4/drivers
$PROJ_DIR$/../../../../../../../../middleware/wireless/framework_5.4.2/DCDC/Interface/MKW35Z
$PROJ_DIR$/../../../../../../../../middleware/wireless/framework_5.4.2/XCVR/MKW35Z4
$PROJ_DIR$/../../../../../../../../devices/MKW35Z4
$PROJ_DIR$/../../../../../../../../devices/MKW35Z4/utilities
```

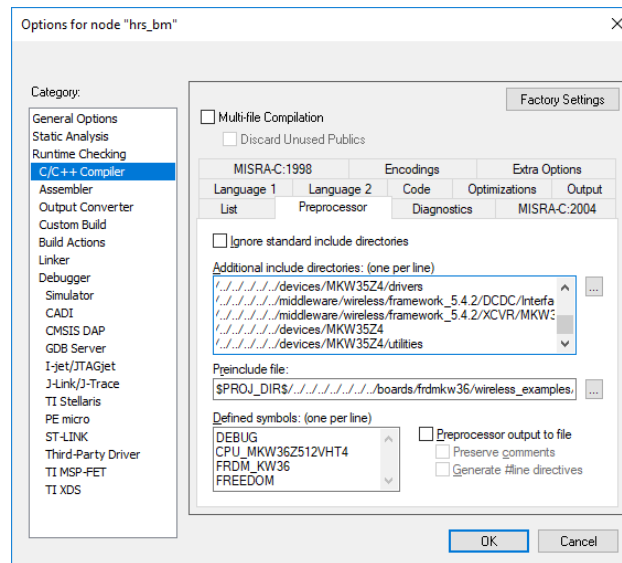


Figure 11. Updating the MKW35 files

7. Rename the CPU_MKW36Z512VHT4 macro as CPU_MKW35Z512VHT4 and delete FRDM_KW36 and FREEDOM macros in defined symbols text box. Click the OK button.

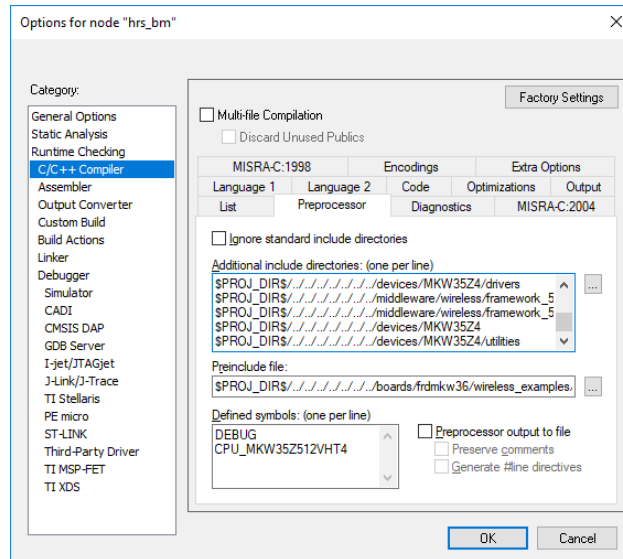


Figure 12. Defining MKW35 macros

- Expand the startup folder, select all files, click the right mouse button and select the “Remove” option. Click right mouse button on the folder and select “Add/Add files”. Add the startup_MKW35Z4.s located at this path:

<SDK_root>/devices/MKW35Z4/iar/startup_MKW35Z4.s

Also, add system_MKW35Z4.c and system_MKW35Z4.h into the startup folder. Both files are located at next path:

<SDK_root>/devices/MKW35Z4

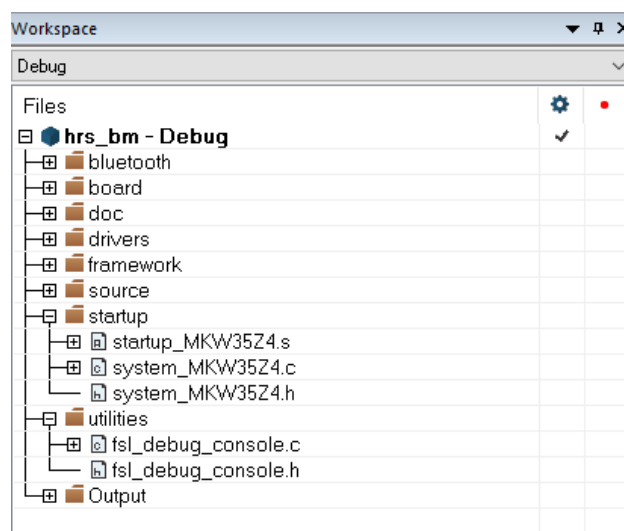


Figure 13. Replacing startup and system files

Migration Guide from MKW36Z512xxx4 to MKW35Z512xxx4, User's Guide, Rev. 1, 06/2019

- Some files need to be updated. Go to PWRLib.c file (in workspace framework/LowPower/Source). Replace LPUART0_LPUART1_IRQn by LPUART0_IRQn in the PWRLib_StopUpdateWakeupReason function.

```
void PWRLib_StopUpdateWakeupReason(void)
{
    uint32_t iser;
    uint32_t ispr;

    iser = NVIC->ISER[0];
    ispr = NVIC->ISPR[0];

    if(( iser & ispr) & (1 << LPUART0_IRQn)) //Replace LPUART0_LPUART1_IRQn
    {
        PWRLib_MCU_WakeupReason.Bits.FromPSwitch_UART = 1;
    }
}
```

Figure 14. Updating PWRLib.c file

- Open clock_config.c file in board folder. Delete 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

- If it is 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 LED's, timers, power consumption settings, etc. Examine and update this file to fit into own custom board.

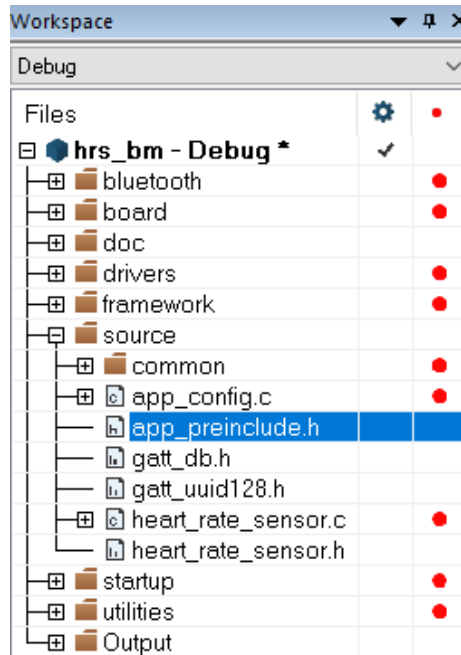


Figure 16. Location of preinclude file

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

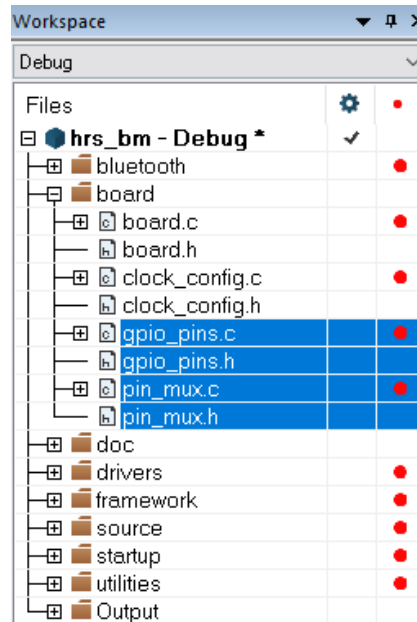


Figure 17. Location of pin mux and gpio files

13. Save the MKW35Z512xxx4_connectivity.icf file located at:

<SDK_root>/middleware/wireless/framework_5.4.4/Common/devices/MKW35Z4/iar

Into the folder of the heart rate sensor project located at the following path:

<SDK_root>/boards/frdmkw36/wireless_examples/bluetooth/hrs/bm/iar

14. Open the project options window (Alt+F7). In Linker/Config window click the icon next to linker path and select the linker configuration file MKW35Z512xxx4_connectivity.icf.

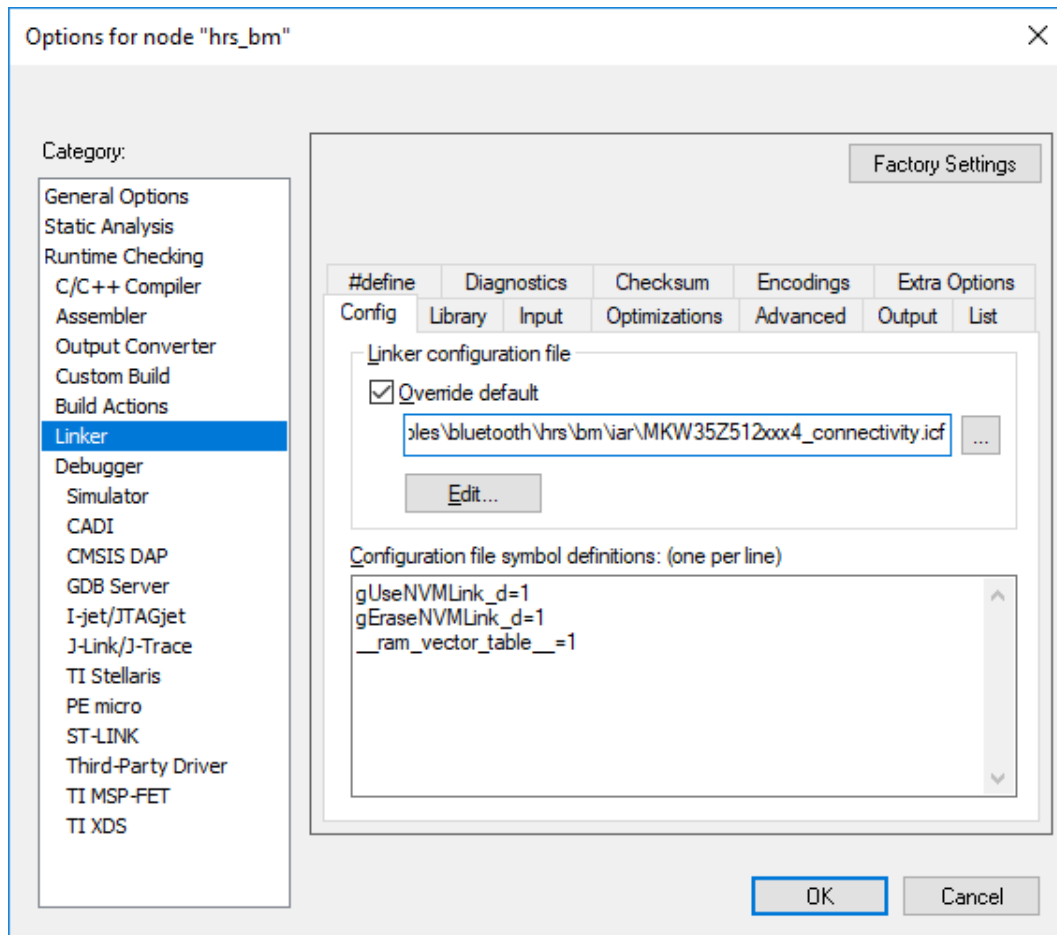


Figure 18. Updating the linker file

15. Click the OK button in the project options window to save the new configuration.
16. Select the project in the workspace, click the right mouse button and press clean.
17. Rebuild the project. At this point, the project is already migrated and you can customize the application for your own purpose using MKW35 device.

5. Software Migration in MCUXpresso IDE

This chapter provides all the steps needed to migrate a MKW36 example code to MKW35 MCU in MCUXpresso IDE. The Heart Rate Sensor project is used as base because it has easy to understand example and involves the Bluetooth LE connectivity software stack (included in SDK).

5.1. Changes Required in Project Properties

1. Find the **Quickstart Panel** in the lower left-hand corner.

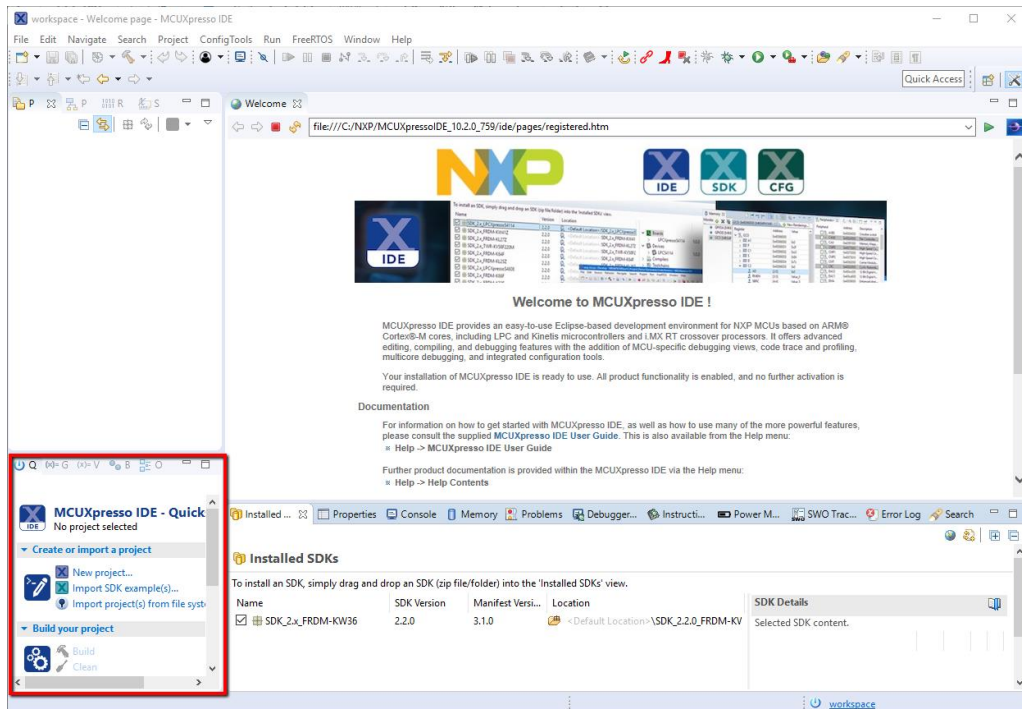


Figure 19. Quickstart Panel perspective in MCUXpresso IDE

2. Then click on “Import SDK example(s)...” option.

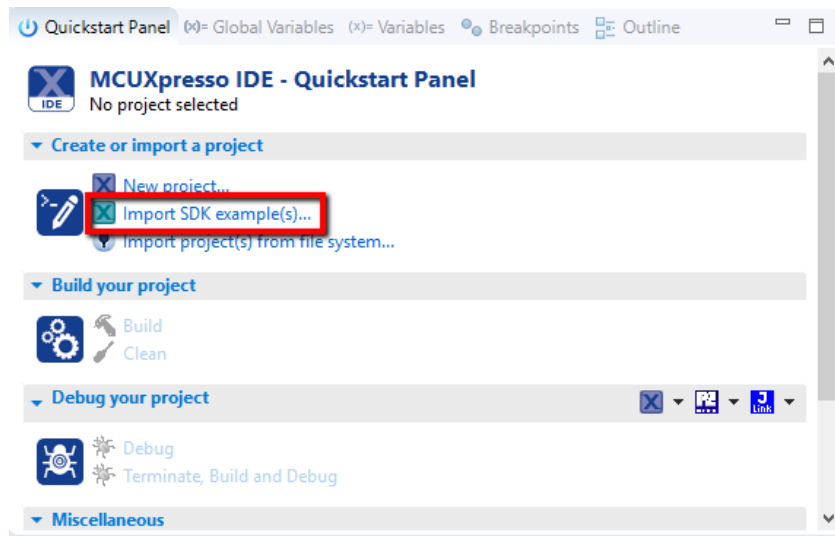


Figure 20. Importing an SDK example

- Click on the **frdmkw36** board and then click on Next.

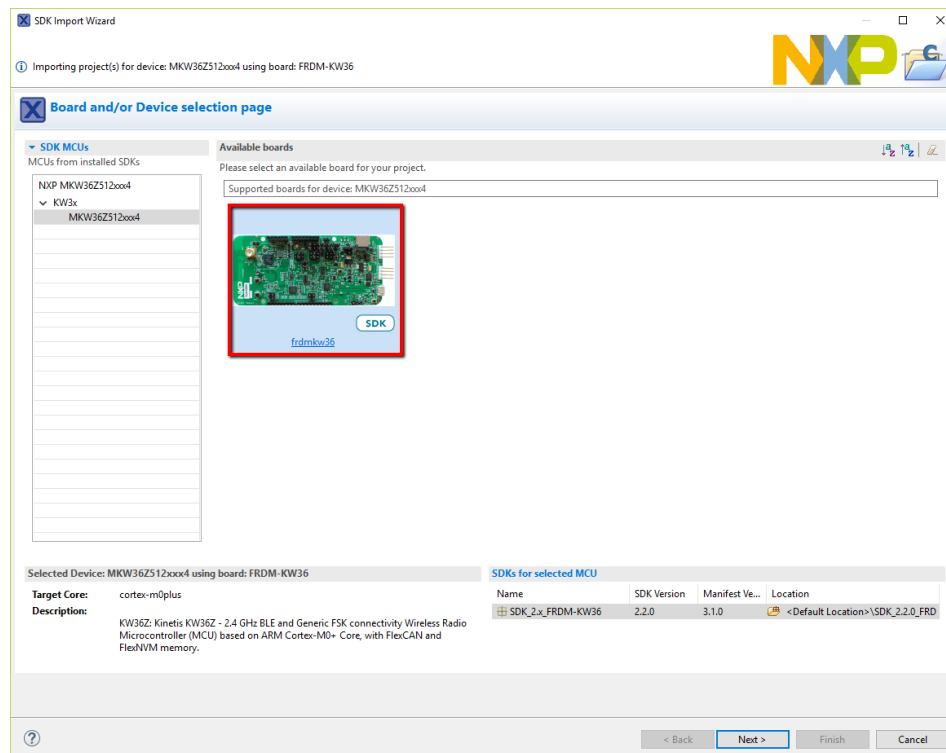


Figure 21. Select FRDM-KW36 board

- In the search text box, type “hrs” to filter the example projects. Use the arrow button to expand the list and locate the “hrs” project (wireless_examples -> bluetooth -> hrs), then, select the **bm** version of the project and click “Finish”.

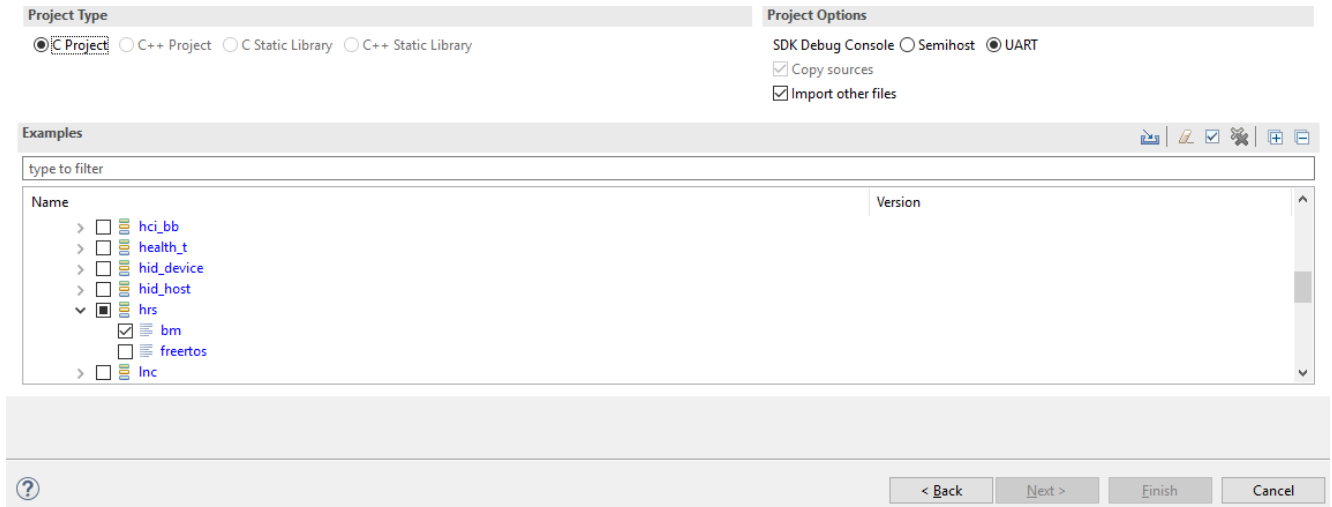


Figure 22. Opening heart rate sensor project

- Go to *Project/Properties*. Expand *C/C++ Build/MCU settings* and select MKW35Z512xxx4 MCU. Click Apply and Close button to save the configuration.

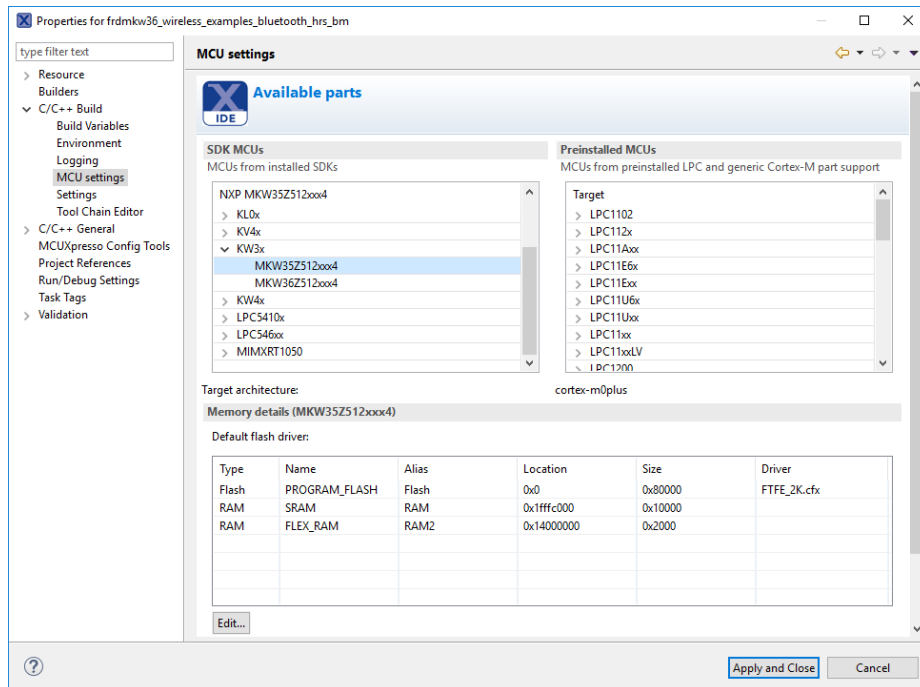


Figure 23. Selecting MKW35 MCU

- Rename MKW36 folders as MKW35 at the following paths (Default MCUXpresso root in Documents/MCUXpressoIDE<version>/workspace):

<MCUXpresso_hrs_project_root>/framework/DCDC/Interface

<MCUXpresso_hrs_project_root>/framework/DCDC/Source

<MCUXpresso_hrs_project_root>/framework/LowPower/Interface

<MCUXpresso_hrs_project_root>/framework/LowPower/Source

<MCUXpresso_hrs_project_root>/framework/XCVR

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

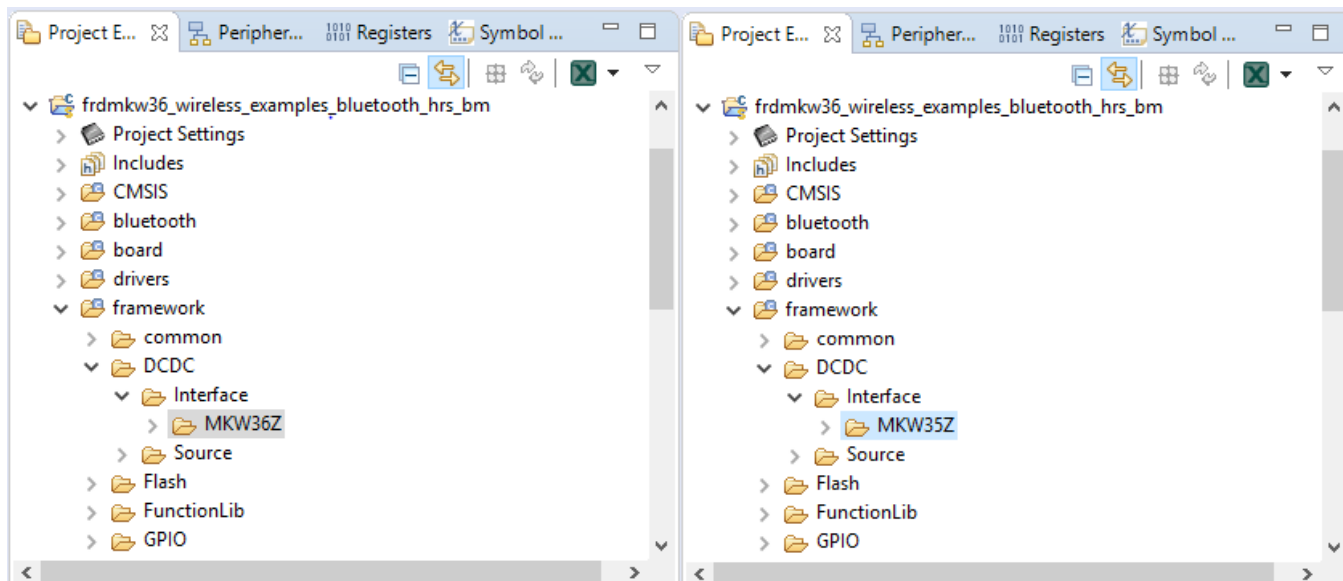


Figure 24. Updating project folders in the workspace

- Open the *Project/Properties* window in MCUXpresso IDE. Go to *C/C++ Build/Settings* and select *MCU C Compiler/Includes* folder in *Tool Settings* window. Edit all paths related to MKW36 MCU, in according to MKW35 folders before created. The results must look similar as shown below:

../framework/LowPower/Interface/MKW35Z

../framework/LowPower/Source/MKW35Z

../framework/DCDC/Interface/MKW35Z

../framework/XCVR/MKW35Z4

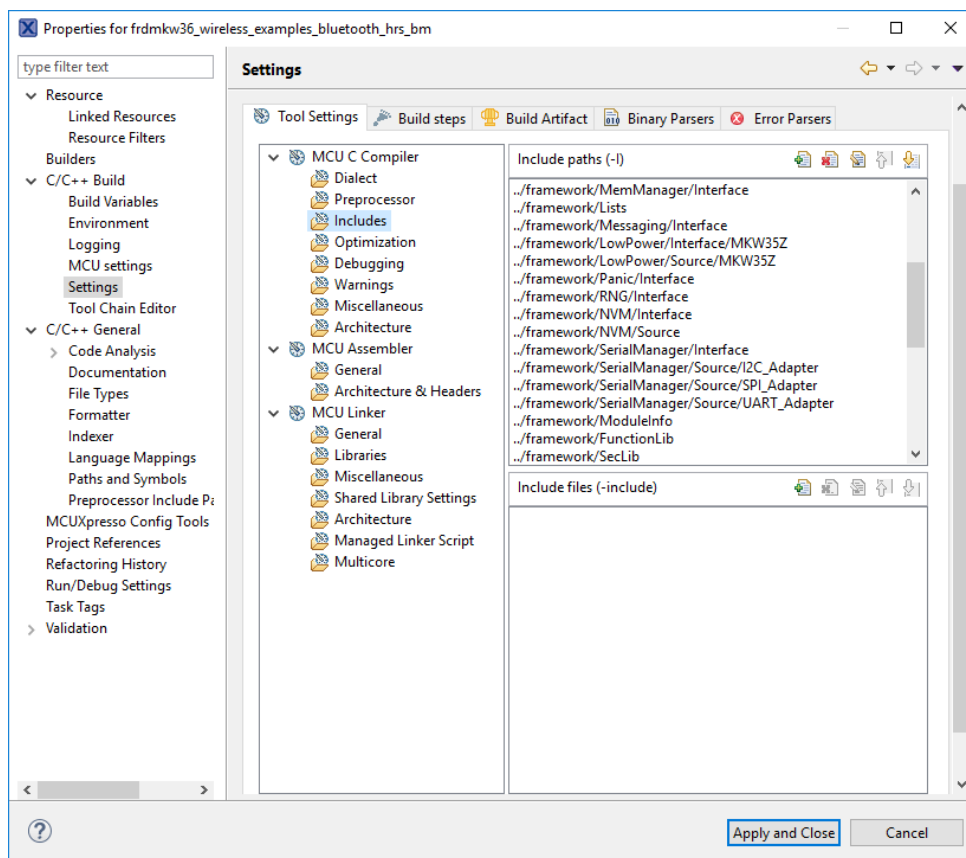


Figure 25. Updating include paths

9. Select *MCU Assembler/General* folder in *Tool Settings*. Edit the paths related to MKW36 MCU. The results must look similar as shown below:

```

../framework/LowPower/Interface/MKW35Z
../framework/LowPower/Source/MKW35Z
../framework/DCDC/Interface/MKW35Z
../framework/XCVR/MKW35Z4
    
```

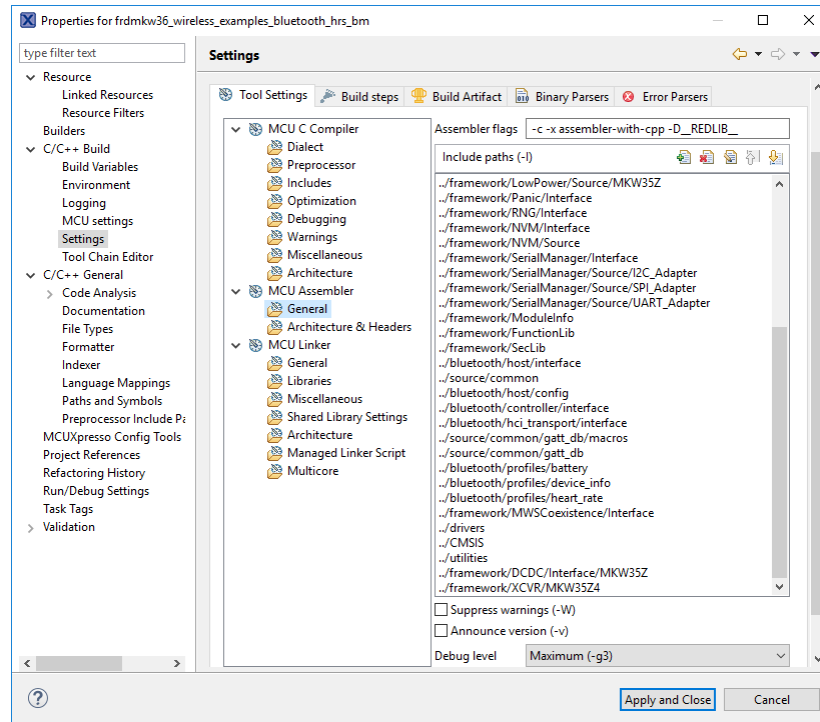


Figure 26. Updating assembler paths

- Open *MCU C Compiler/Preprocessor* window, edit the CPU_MKW36Z512VHT4 and CPU_MKW36Z512VHT4_cm0plus macros by CPU_MKW35Z512VHT4_cm0plus and CPU_MKW35Z512VHT4_cm0plus respectively. Delete FRDM_KW36 and FREEDOM macros. Then click Apply and Close button.

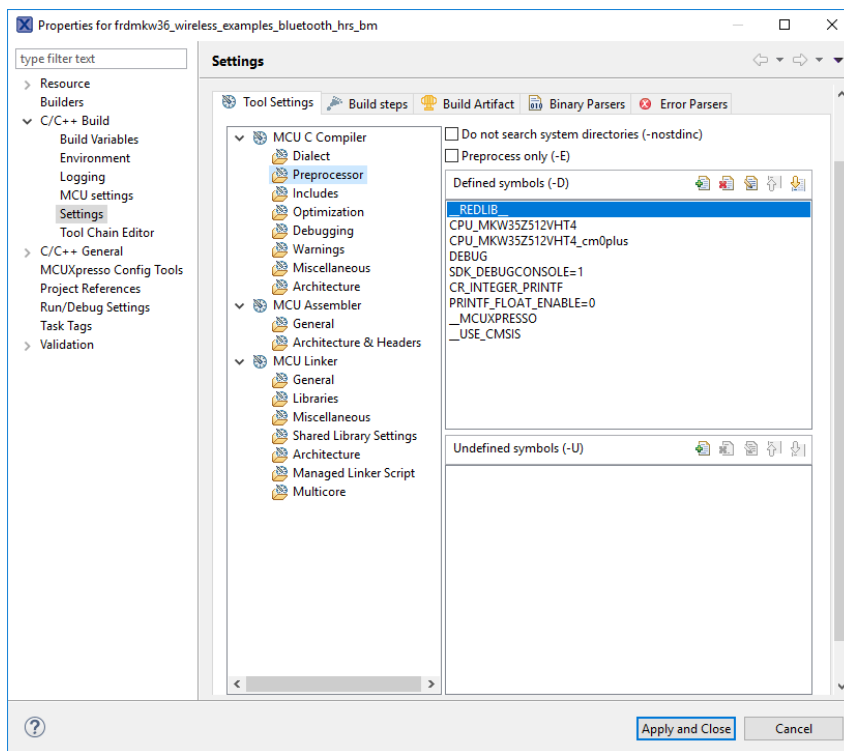


Figure 27. Updating preprocessor macros

11. Delete “fsl_device_registers, MKW36Z4, MKW36Z4_features, system_MKW36Z4.h and system_MKW36Z4.c” files located at CMSIS folder in the workspace.

12. Unzip the MKW35Z SDK package and search for “fsl_device_registers, MKW35Z4, MKW35Z4_features, system_MKW35Z4.h and system_MKW35Z4.c” files into this folder at the following paths:

```

<SDK_folder_root>/devices/MKW35Z4/fsl_device_registers.h
<SDK_folder_root>/devices/MKW35Z4/MKW35Z4.h
<SDK_folder_root>/devices/MKW35Z4/MKW35Z4_features.h
<SDK_folder_root>/devices/MKW35Z4/system_MKW35Z4.h
<SDK_folder_root>/devices/MKW35Z4/system_MKW35Z4.c
    
```

Drag and drop this files into the CMSIS folder in the workspace.

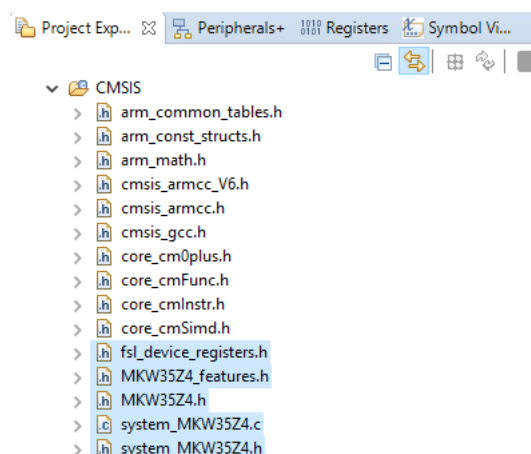


Figure 28. Updating CMSIS files

- Replace the “startup_MKW36Z.S” file located at startup folder in the workspace by the “startup_MKW35Z4.S”. This file is located, as the same as in the last step, in the MKW35Z SDK at following path:

<SDK_folder_root>/devices/MKW35Z4/gcc/startup_MKW35Z4.S

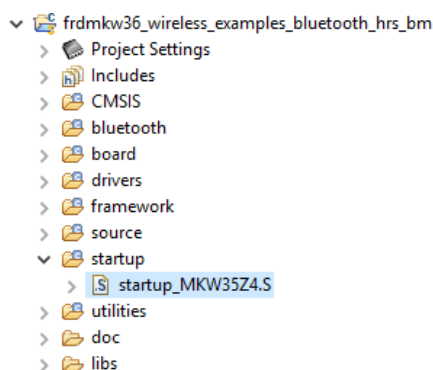


Figure 29. Updating the startup file

- Some files need to be updated. Go to PWRLib.c file (in workspace framework/LowPower/Source/MKW35Z). Replace LPUART0_LPUART1_IRQn by LPUART0_IRQn in the PWRLib_StopUpdateWakeupReason function.

```

void PWRLib_StopUpdateWakeupReason(void)
{
    uint32_t iser;
    uint32_t ispr;

    iser = NVIC->ISER[0];
    ispr = NVIC->ISPR[0];

    if(( iser & ispr) & (1 << LPUART0_IRQn)) //Replace LPUART0_LPUART1_IRQn by LPUART0_IRQn
    {
        PWRLib_MCU_WakeupReason.Bits.FromPSwitch_UART = 1;
    }

    if(( iser & ispr) & (1 << LVD_LVW_DCDC_IRQn))
    {
        PWRLib_MCU_WakeupReason.Bits.FromPSwitch_UART = 1;
    }
}

```

Figure 30. Updating PWRLib.c file

- Open clock_config.c file in board folder. Delete CLOCK_SetLpuart1Clock call to 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 31. Updating clock_config.c file

- If it is 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 LED's, timers, power consumption settings, etc. Examine and update this file to fit into own custom board.

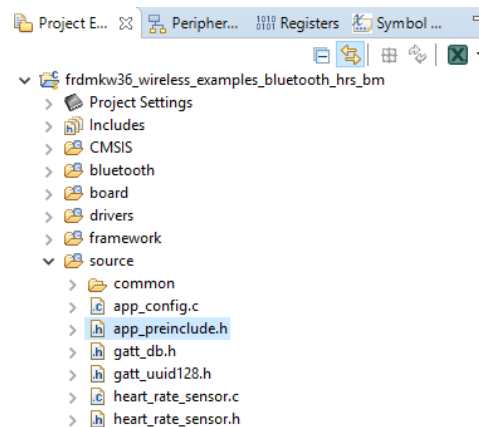


Figure 32. Location of preinclude file

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

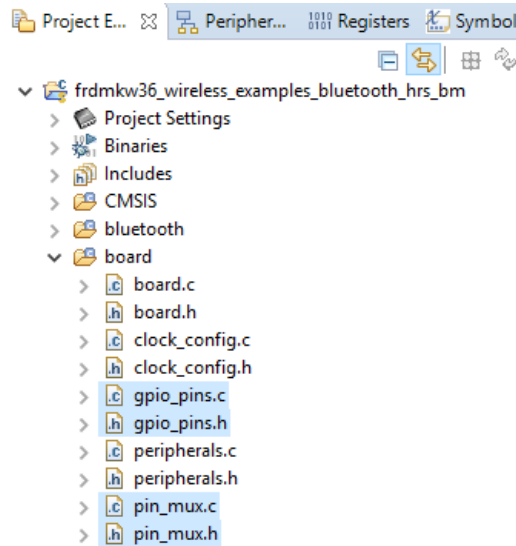


Figure 33. Location of pin mux and gpio files

18. Replace the MKW36Z512xxx4_PD_connectivity.ld file by MKW35Z512xxx4_connectivity.ld located at:

`<SDK_root>/middleware/wireless/framework_5.4.4/Common/devices/MKW35Z4/gcc`

Into the source folder in the workspace.

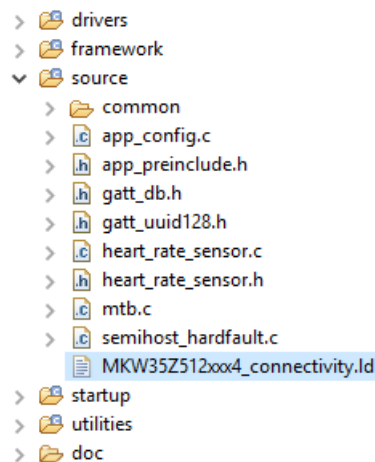


Figure 34. Replacing the linker file

- Open the *Project/Properties* window. Next, go to the *MCU Linker/Managed Linker Script* perspective and edit the Linker Script name according with MKW35 device. Press Apply and Close button.

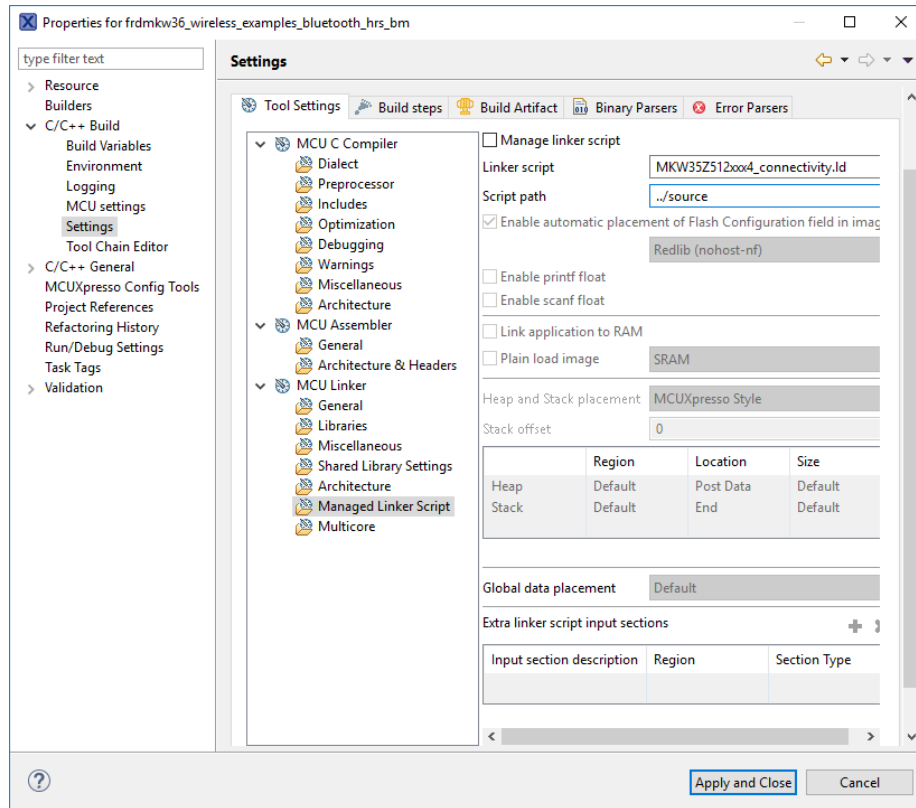


Figure 35. Establishing the linker path

- Build the project. At this point, the project is already migrated and you are ready to play with the demo using MKW35.

6. Build and Run Bluetooth LE Connectivity Stack Examples

All the examples referenced in BLE Demo Applications User’s Guide (BLEDAUG) are compatible with MKW35 device following modifications described in this document. The BLEDAUG is part of the FRDM-KW36 SDK documentation.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018-2019 NXP B.V.

Document Number: AN12252
Rev. 1
06/2019

