

1 简介

本文档介绍如何从 MKW36Z512xxx4 迁移到 MKW35Z512xxx4 微控制器，重点介绍了连接软件。在本文中，将 MKW36Z512xxx4 和 MKW35Z512xxx4 分别称为 MKW36 和 MKW35。本文档面向软件工程师、软件测试人员、软件集成商和自己设计硬件的客户。

2 硬件注意事项

MKW35 无线 MCU 与 MKW36 引脚对引脚兼容，并且两种设备中的几乎所有外设都一致。

MKW36 和 MKW35 微控制器之间的主要区别与片上闪存的大小和寻址有关。表 1 为两者的不同之处。

表 1. MKW36 和 MKW35 的不同

设备	512 KB P-Flash	256 KB P-Flash + 256 KB FlexNVM	能否仿真 8 KB EEPROM	LPUART 的第 2 个实例	FlexCAN 模块
MKW36		X	X	X	X
MKW35	X				

2.1 外设实例化

MKW36/35 设备对两种可用封装（40 针 QFN 和 48 针 LQFN）都是 pin to pin 管脚兼容的。MKW35 不支持 LPUART1 和 FlexCAN，因此，最新的信号在此设备中不可用于多路复用。替代方案只包括 MKW36 微控制器。

表 2. MKW35/36 实例比较

引脚名字	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_PCS 0	LPUART1_ CTS_b		TPM2_CH1				
PTB0		PTB0/ LLWU_P8/ RF_		I2C0_SCL	CMPO_OUT	TPM0_CH1		CLKOUT	CAN0_TX	

下页继续...

目录

1	简介.....	1
2	硬件注意事项.....	1
2.1	外设实例化.....	1
2.2	MCU 片上存储器.....	3
3	软件开发包下载安装.....	4
4	IAR 嵌入式工作台软件迁移.....	6
4.1	项目选项和设置中所需的更改.....	6
5	MCUXpresso IDE 中的软件迁移... ..	14
5.1	项目属性所需的更改.....	14
6	构建并运行低功耗蓝牙连接堆栈示例.....	25



表 2. MKW35/36 实例比较 (续上页)

引脚名字	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9
		RFOSC_EN/ RF_RESET								
PTB1	ADC0_SE1 / CMP0_IN5	PTB1/ RF_PRIORITY	DTM_RX	I2C0_SDA	LPTMR0_A LT1	TPM0_CH2		CMT_IRO	CAN0_RX	
PTB3	ADC0_SE2 / CMP0_IN4	PTB3	LPUART1_ RTS_b		CLKOUT	TPM1_CH1		RTC_CLKO UT		
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_SWITC H	I2C1_SDA	LPUART0_ TX	TPM0_CH1		DTM_TX	SPI1_SIN	CAN0_TX
PTC4		PTC4/ LLWU_P12/ BLE_RF_A CTIVE		EXTRG_IN	LPUART0_ CTS_b	TPM1_CH0		BSM_DATA	SPI1_PCS 0	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_O SC_EN	SPI0_SOU T	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_PCS 0	I2C0_SCL	LPUART0_ CTS_b	BSM_CLK			LPUART1_ CTS_b	

注意

表 2 不是 MKW35/36 引脚的完整描述。

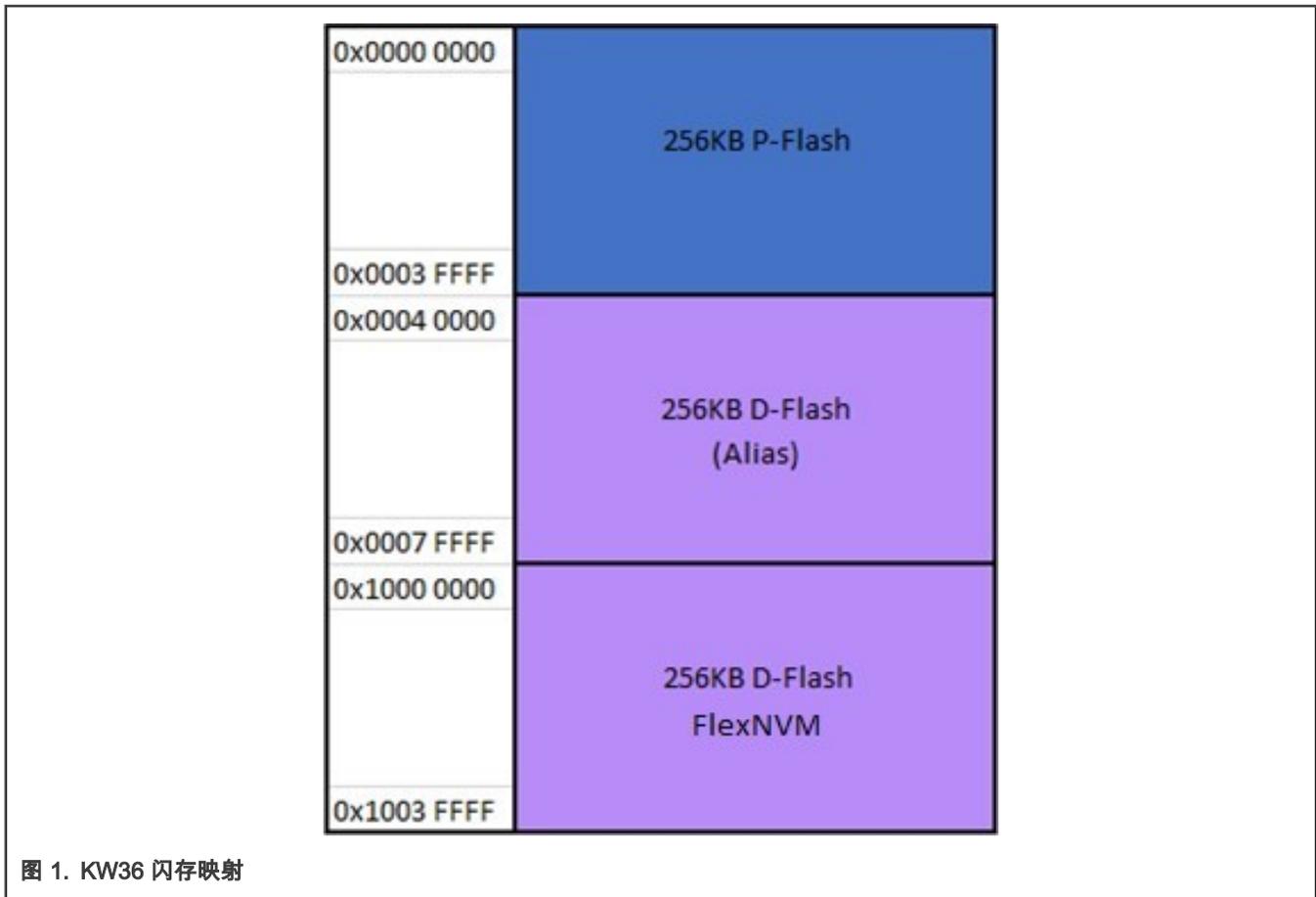
由于引脚数不同，可用的数字引脚数在 40 引脚和 48 引脚之间变化。例如，在 48 引脚封装中，总共有 25 个数字引脚，而在 40 引脚封装中，则有 18 个数字引脚。

2.2 MCU 片上存储器

2.2.1 MKW36 闪存映射

MKW36 闪存被划分为：

- 一个 256 KB 程序闪存 (P-Flash) 阵列，每个扇区 2 KB。闪存地址范围从 0x0000_0000 到 0x0003_FFFF。
- 一个 256 KB FlexNVM 阵列，被大小为 2 KB 扇区的划分为等分，闪存地址范围为 0x1000_0000 到 0x1003_FFFF ，并具有 Alias 存储器，地址范围从 0x0004_0000 到 0x0007_FFFF。FlexNVM 存储器可用作程序闪存 (P-Flash) ，数据闪存 (D-Flash) 和 EEPROM 闪存。



2.2.2 MKW35 闪存映射

MKW35 闪存不支持 FlexNVM 或 Alias 内存，因此，Program Flash 阵列增加到：

- 2 x 256 KB 程序闪存 (P-Flash) 阵列，从 0x0000_0000 到 0x0007_FFFF 地址开始，按 2 KB 扇区划分。

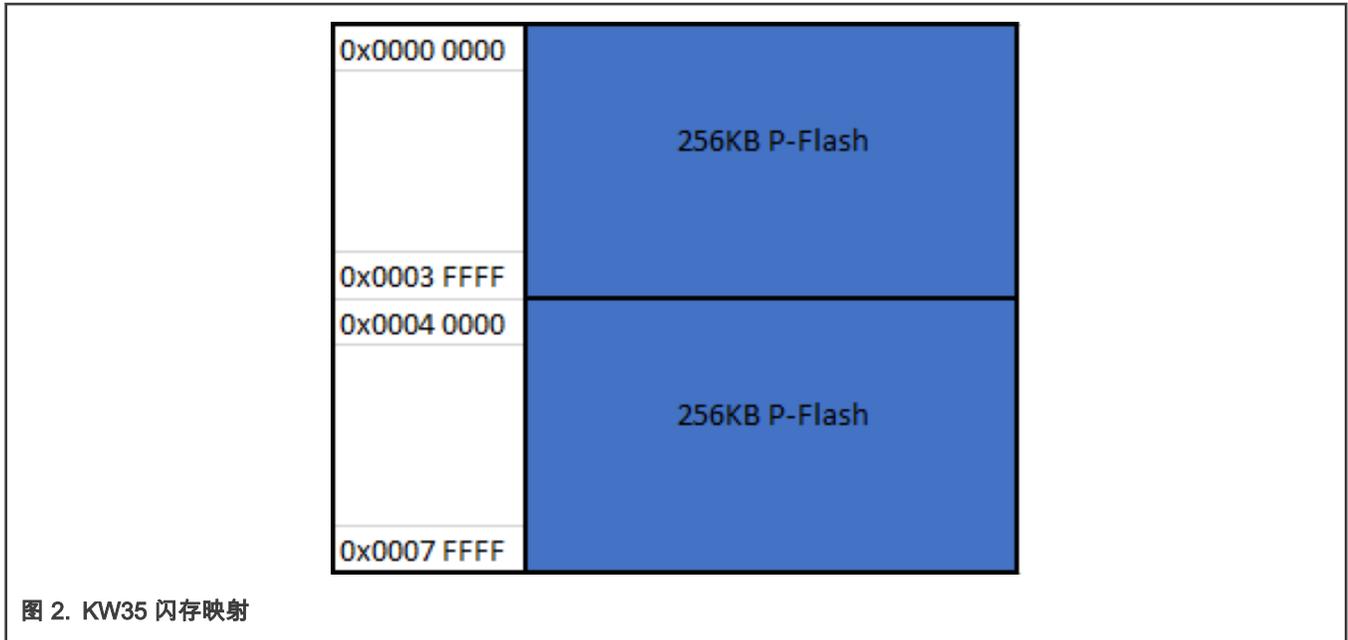


图 2. KW35 闪存映射

3 软件开发包下载安装

本章提供下载 MKW35Z512xxx4 的 SDK (软件开发工具包) 所需的所有步骤，因为它是将 MKW36 代码迁移到 MKW35 MCU 的起点。以下是下载 SDK 软件包的步骤：

1. 转到 [MCUXpresso SDK Builder](#) 网页。
2. 使用您的注册账户登录。
3. 搜索“KW35Z”设备。然后单击建议的处理器并单击“Build MCUXpresso SDK”。

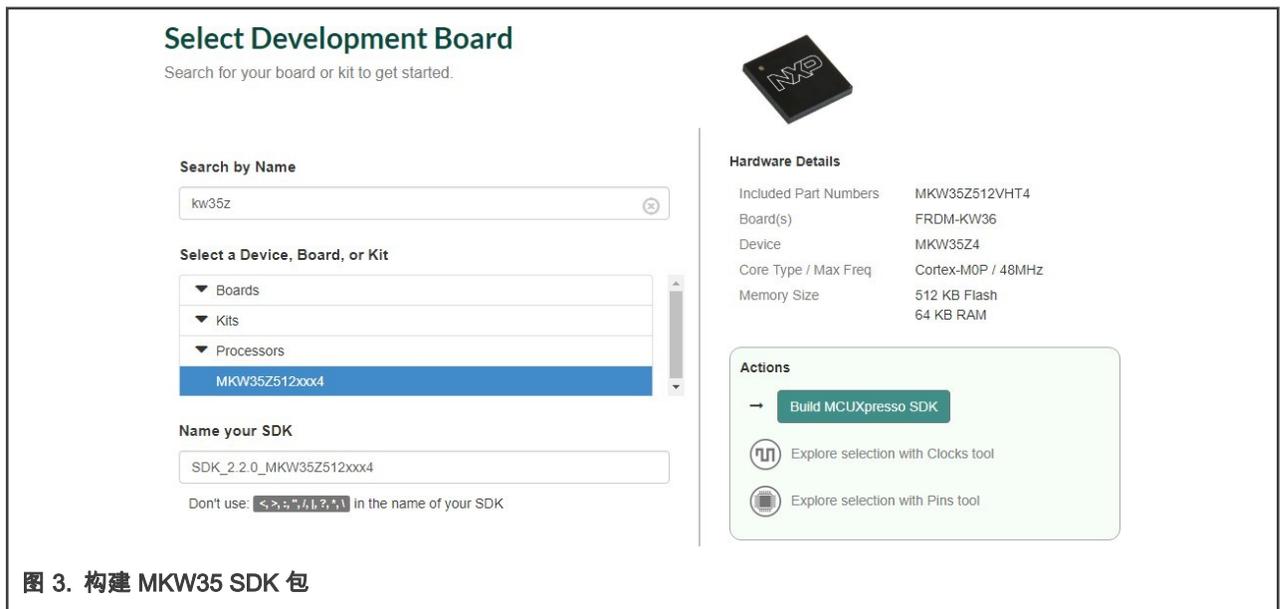


图 3. 构建 MKW35 SDK 包

4. 显示下一页。在“Toolchain/IDE”框中选择“All toolchains”，并提供名称以标识软件包。

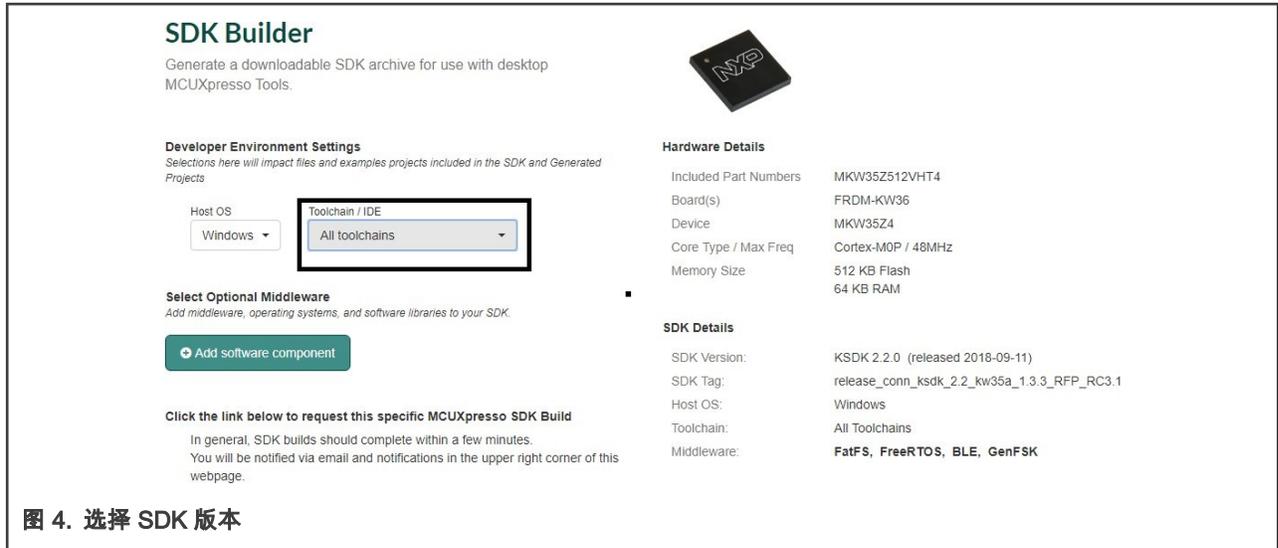


图 4. 选择 SDK 版本

5. 点击“Request Build”。这将启动所需 SDK 的构建过程。几分钟后，系统会软件包放入 MCUXpresso 网页的配置文件中。
6. SDK 准备下载后，将显示在 MCUXpresso 仪表板中，如 图 5 所示。

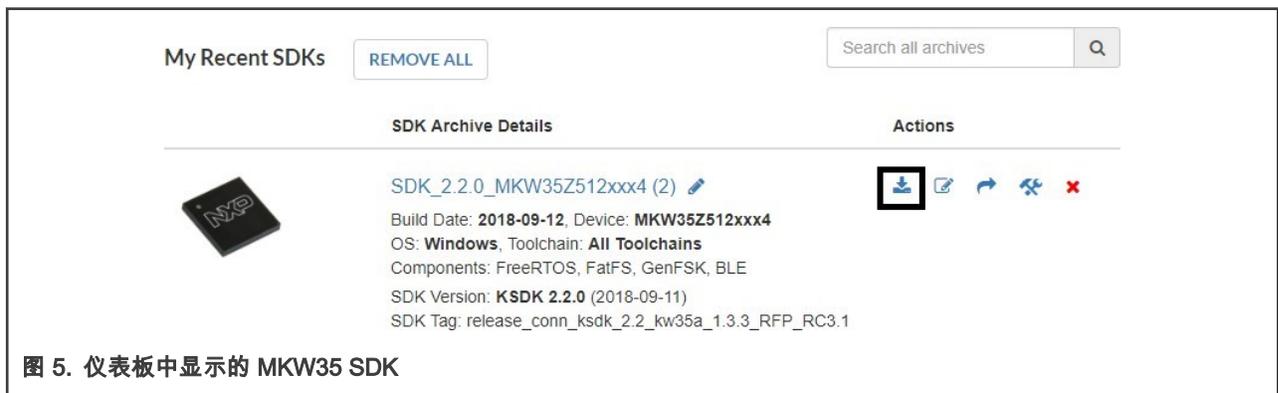


图 5. 仪表板中显示的 MKW35 SDK

7. 单击下载图标，如 图 5 所示。出现一个新窗口来下载软件包。

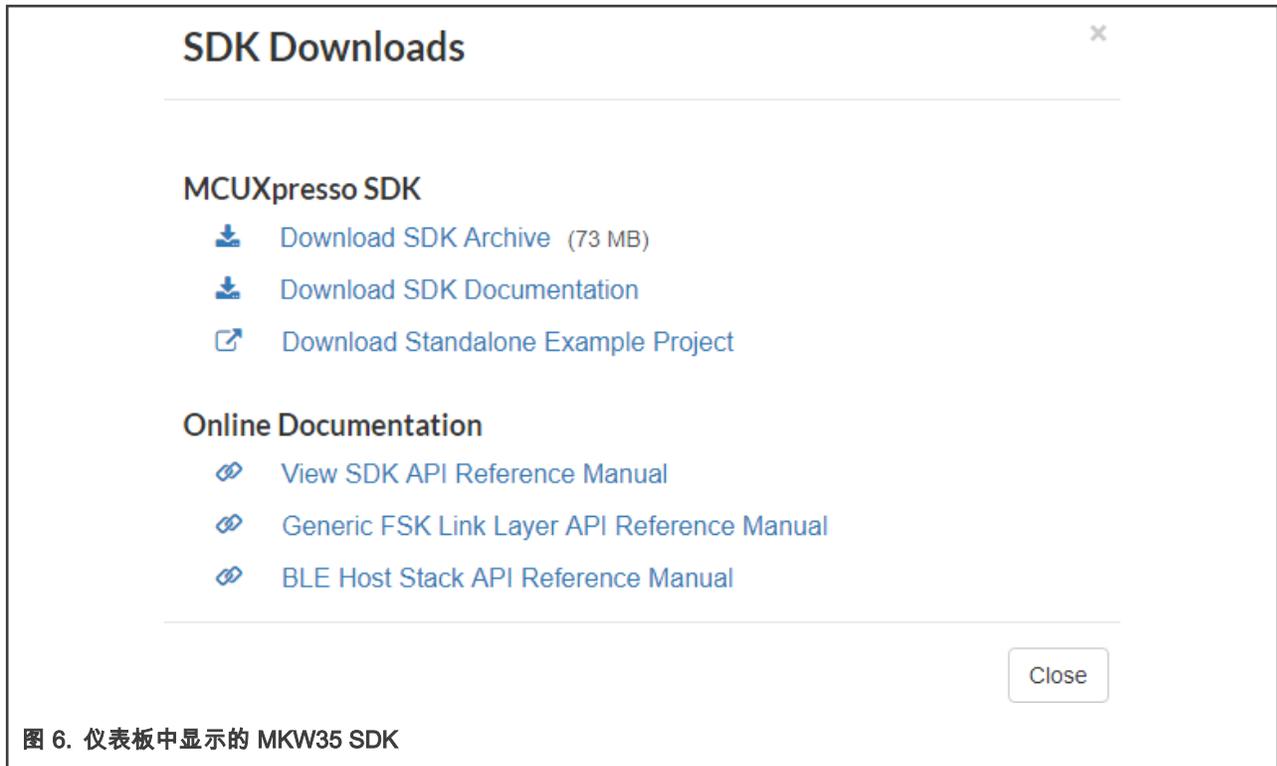


图 6. 仪表板中显示的 MKW35 SDK

8. 接受许可协议，然后将 SDK 保存在所需的位置。
 9. 如果使用了 MCUXpresso IDE，则将 MKW35 SDK.zip 文件夹拖放到“Installed SDK's”透视图中以安装该软件包。
- 至此，您已经下载并安装了 MKW35Z 设备的 SDK 包。

4 IAR 嵌入式工作台中的软件迁移

本章提供将 MKW36 示例代码迁移到 IAR Embedded Workbench IDE 中的 MKW35 MCU 所需的所有步骤。本文档以心率传感器项目为基础，因为它具有易于理解的示例，并且涉及低功耗蓝牙连接软件堆栈（包含在 SDK 中）。

4.1 项目选项和设置中所需的更改

注意

在本节中，使用了项目的“裸机”版本，但是，相同的步骤也适用于“FreeRTOS”项目。如果使用“FreeRTOS”版本，与“bm”项目相关的某些路径可能会有所不同。

1. 打开位于以下路径的心率传感器项目：

```
<SDK_root>/boards/frdmkw36/wireless_examples/bluetooth/hrs/bm/iar/hrs_bm.eww
```

2. 在工作区中选择项目，然后按 Alt + F7 打开项目选项。

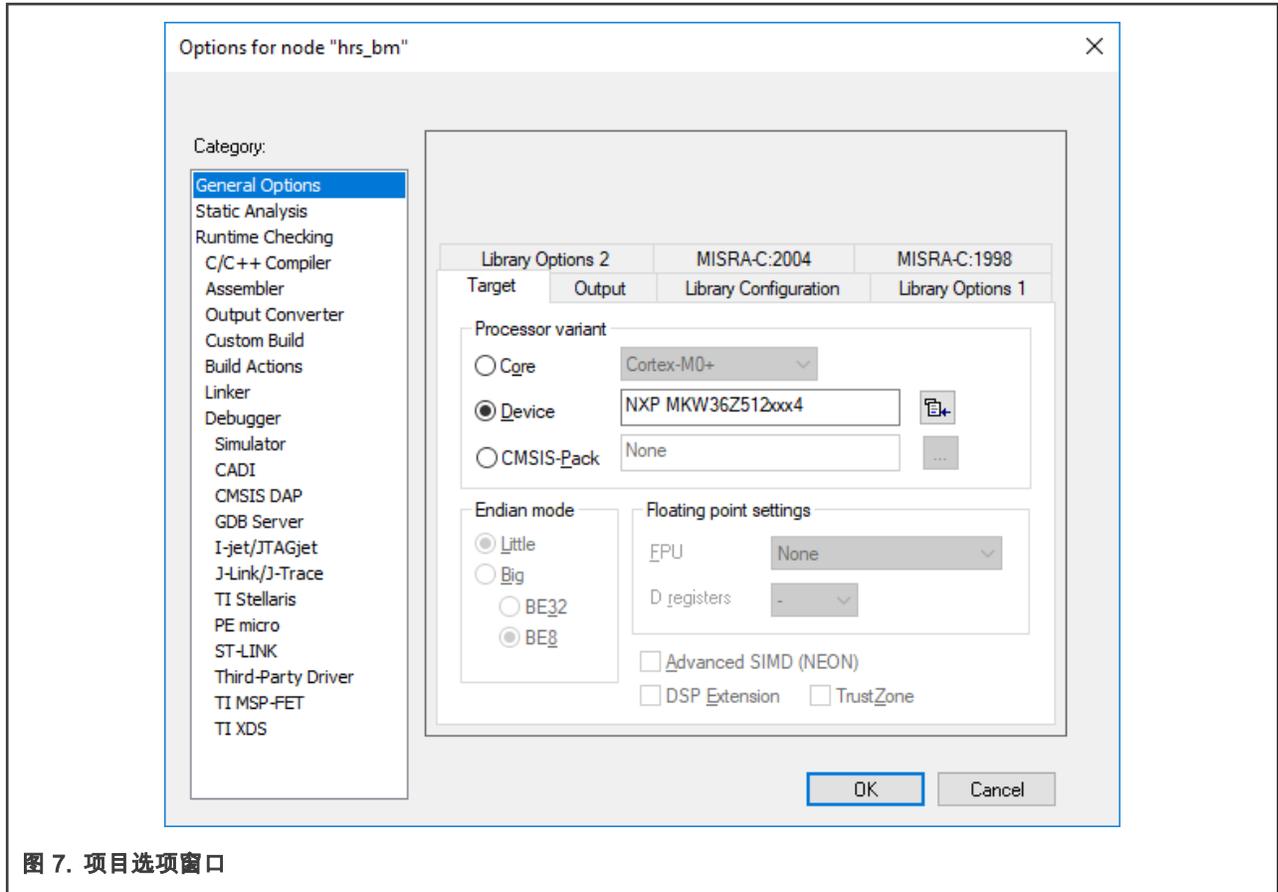


图 7. 项目选项窗口

3. 在“常规选项/目标”窗口中，单击设备名称旁边的图标，选择合适的设备 NXP/KinetisKW/KW3x/NXP MKW35Z512xxx4，然后单击“确定”按钮。

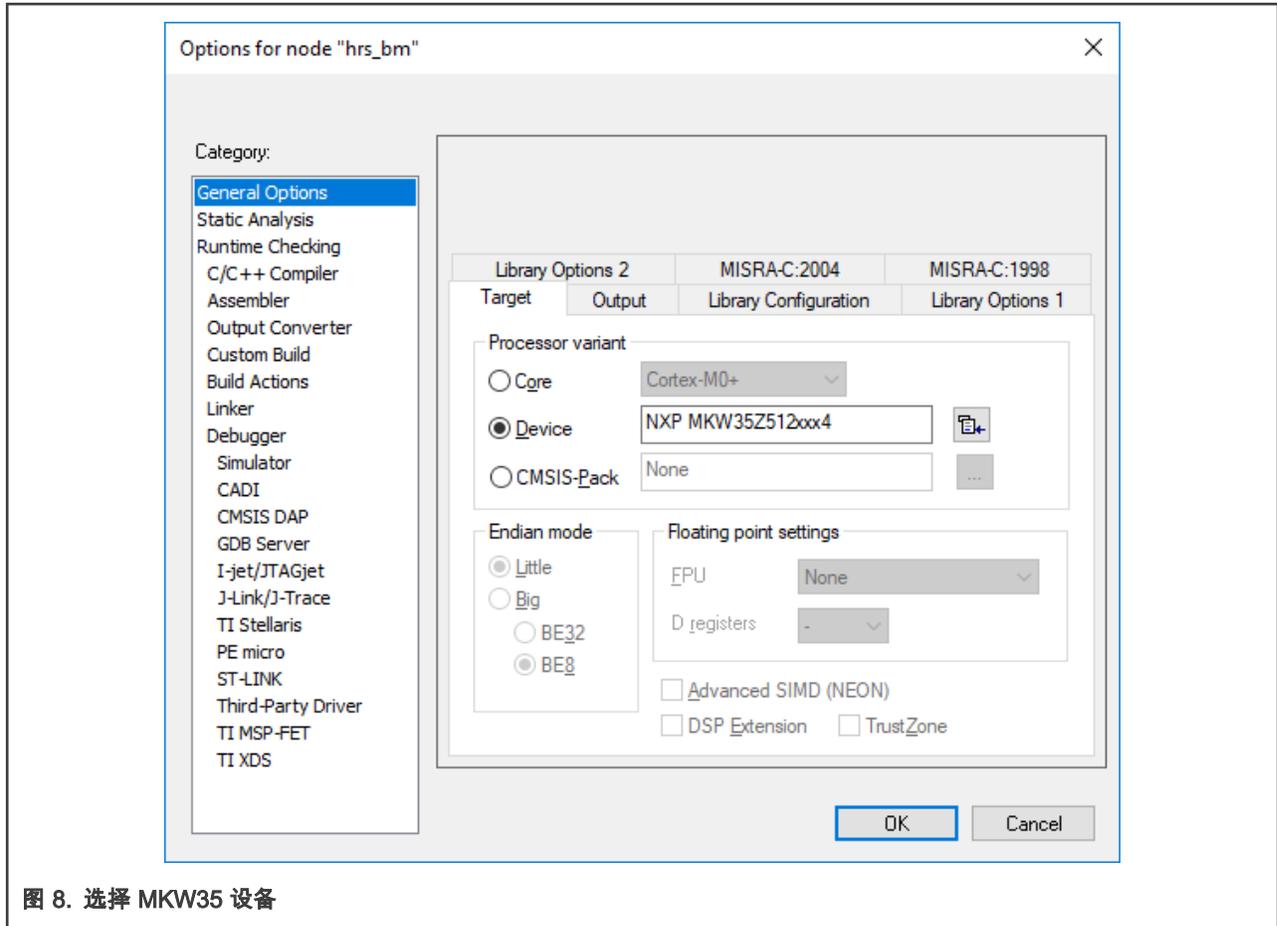


图 8. 选择 MKW35 设备

- 在以下路径中创建一个名为 MKW35Z 的新文件夹：

```
<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
```

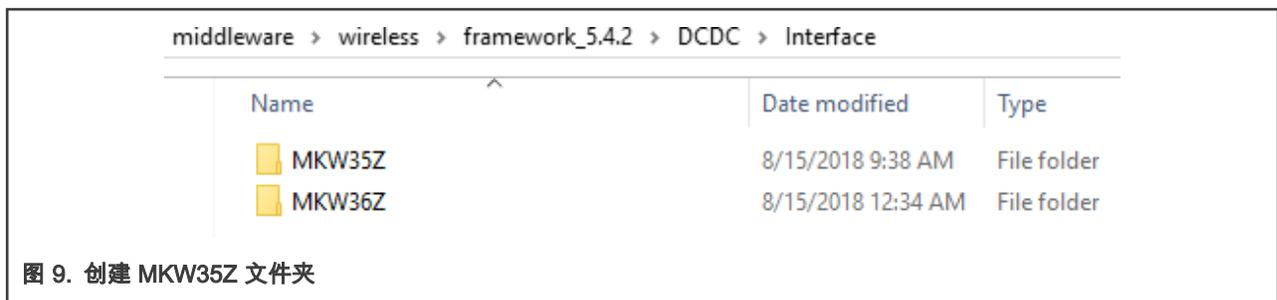


图 9. 创建 MKW35Z 文件夹

- 复制位于上述路径的 MKW36Z 文件夹内的所有文件，然后粘贴到 MKW35Z 文件夹中。

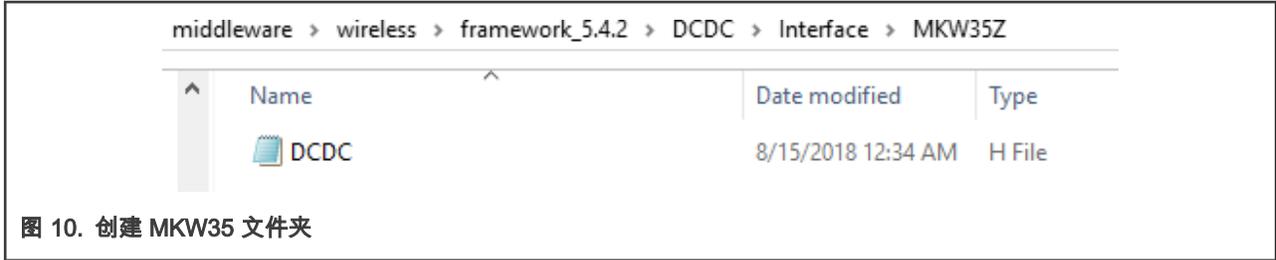


图 10. 创建 MKW35 文件夹

- 在工作区中选择心率传感器，然后按 Alt + F7 打开项目选项窗口。在“C/C++编译器/预处理器”窗口中，将与 MKW36Z 文件夹相关的路径重命名为 MKW35Z 文件夹，从而在附加包含目录文本框中将数字改为 5，而不是 6。结果必须类似于如图 11 所示：

```
$PROJ_DIR$/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
```

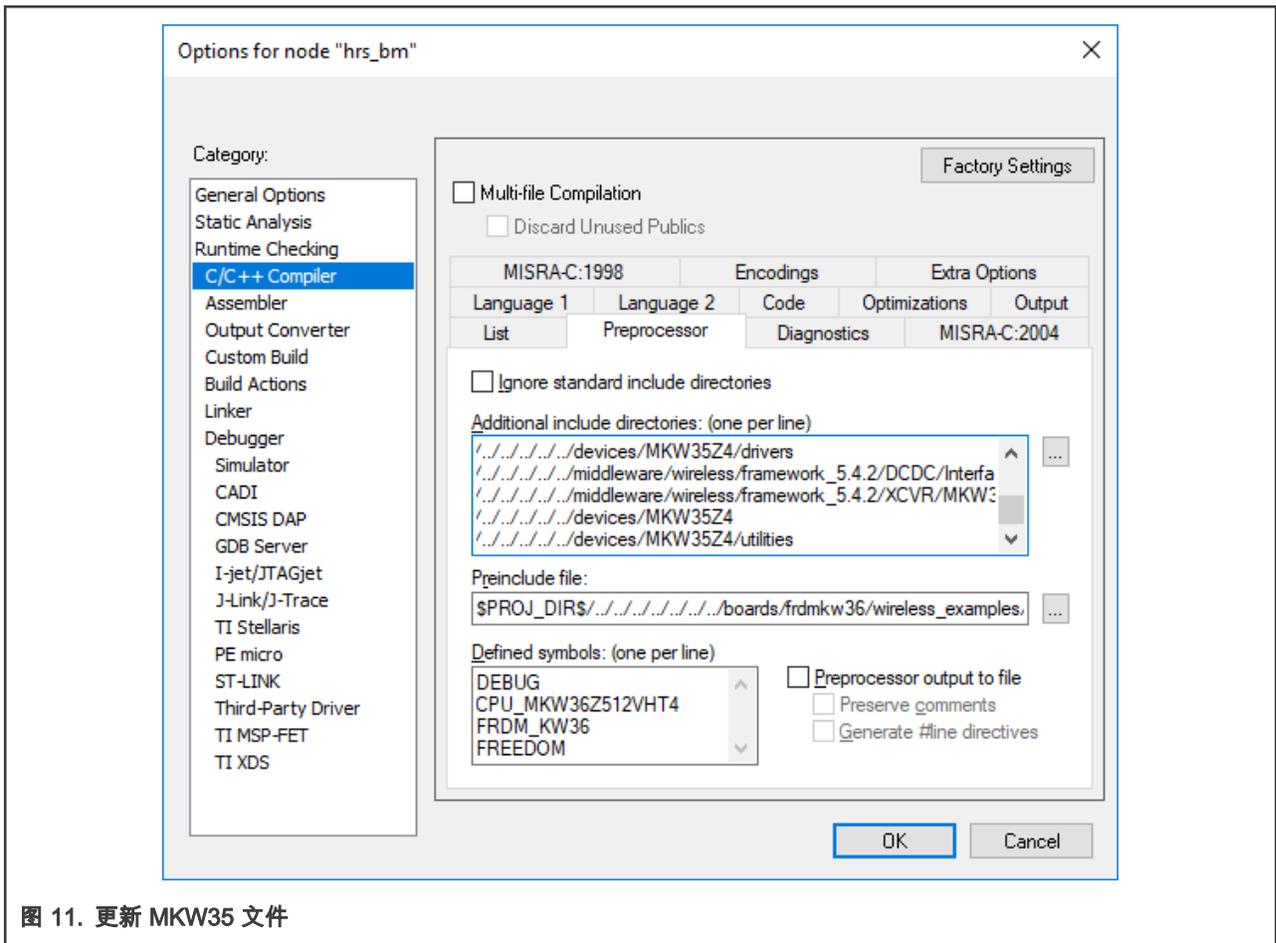


图 11. 更新 MKW35 文件

- 将 CPU_MKW36Z512VHT4 宏重命名为 CPU_MKW35Z512VHT4，然后在已定义符号文本框中删除 FRDM_KW36 和 FREEDOM 宏。单击“确定”按钮。

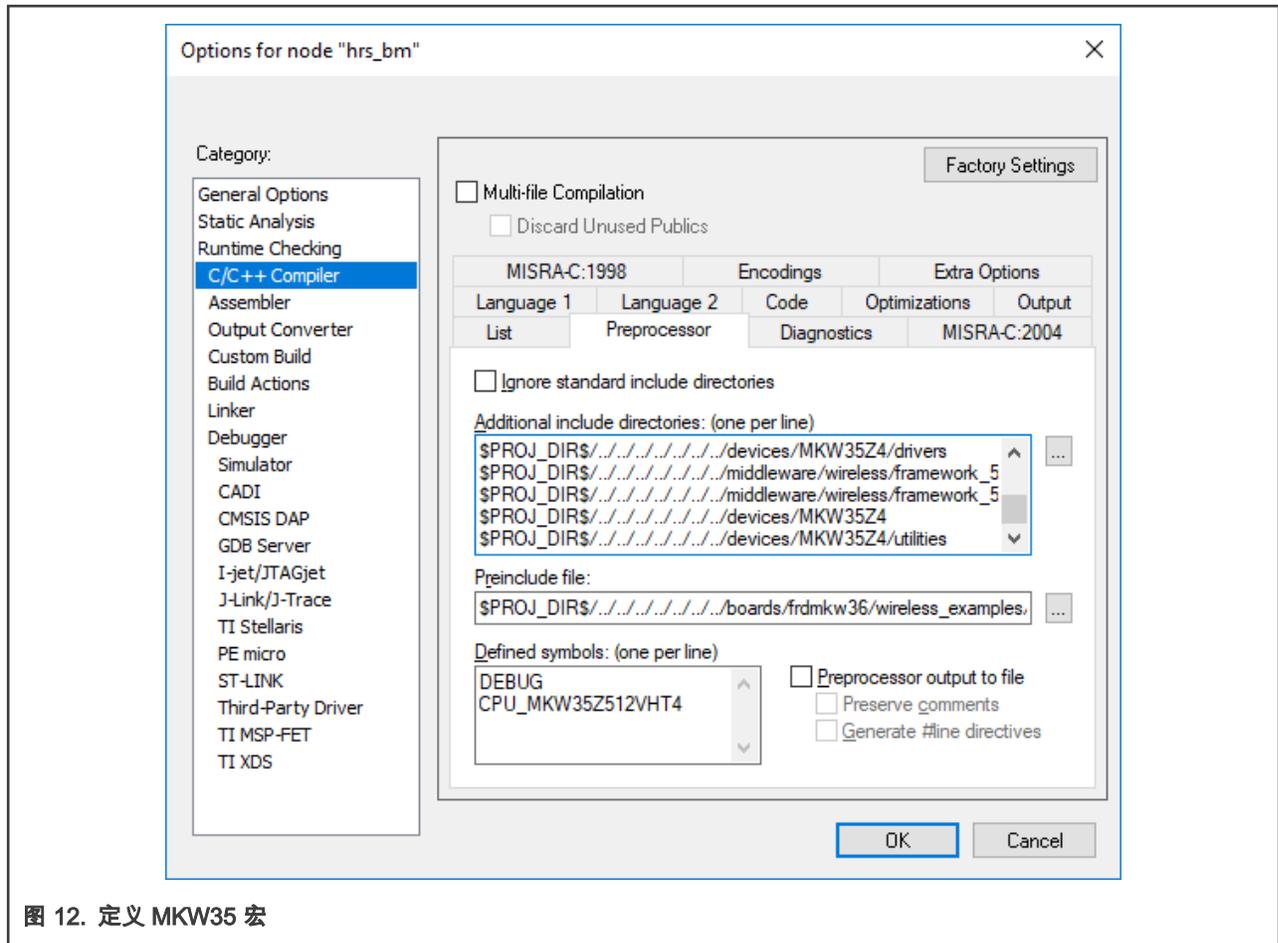


图 12. 定义 MKW35 宏

- 展开启动文件夹，选择所有文件，单击鼠标右键，然后选择“删除”选项。在文件夹上单击鼠标右键，然后选择“添加/添加文件”。添加位于以下路径的 startup_MKW35Z4.s：

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

另外，将 system_MKW35Z4.c 和 system_MKW35Z4.h 添加到启动文件夹中。这两个文件都位于下一个路径：

```
<SDK_root>/devices/MKW35Z4
```

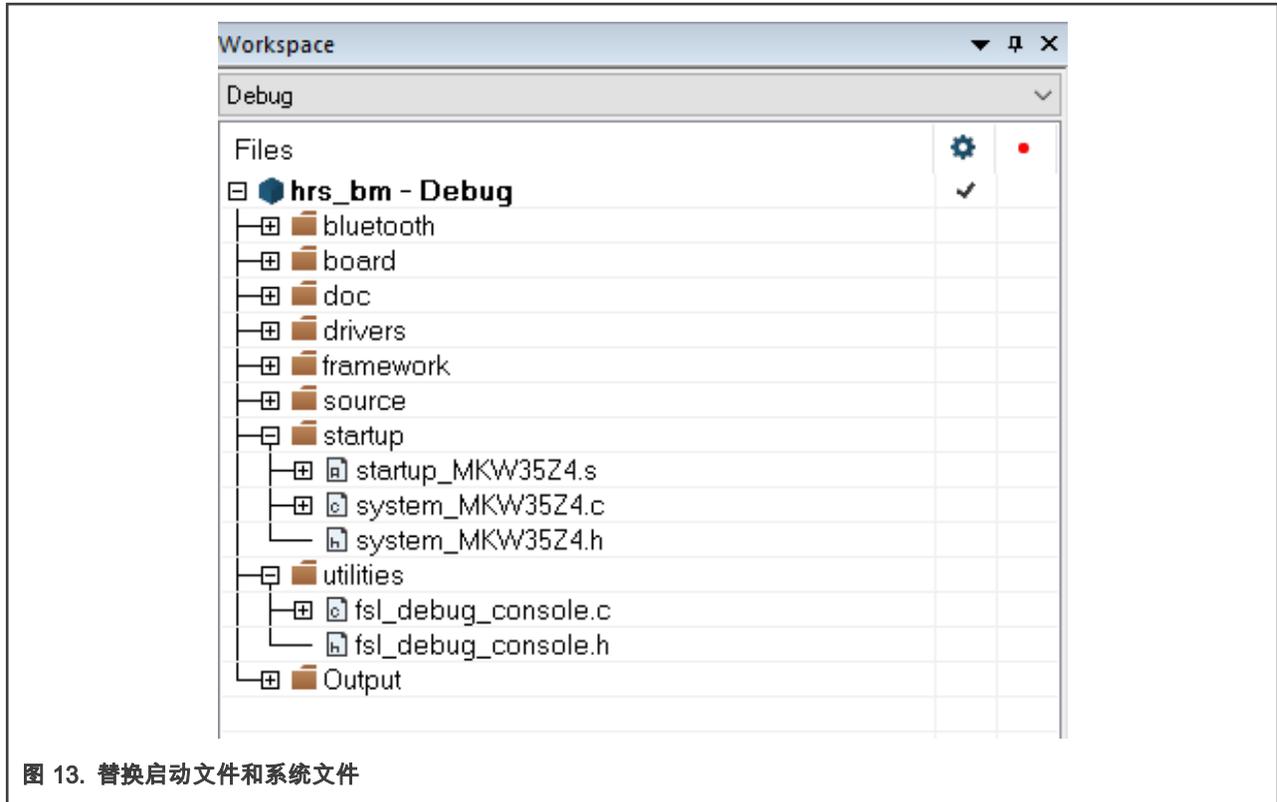


图 13. 替换启动文件和系统文件

- 有些文件需要更新。转到 PWRLib.c 文件（在 workspace framework/LowPower/Source 中）。在 PWRLib\u StopUpdateWakeupReason 函数中将 LPUART0_LPUART1_IRQn 替换为 LPUART0_IRQn。



图 14. 更新 PWRLib.c 文件

- 打开 board 文件夹中的 clock_config.c 文件。删除对 BOARD_BootClockRUN 函数中 CLOCK_SetLpuart1Clock 的调用。

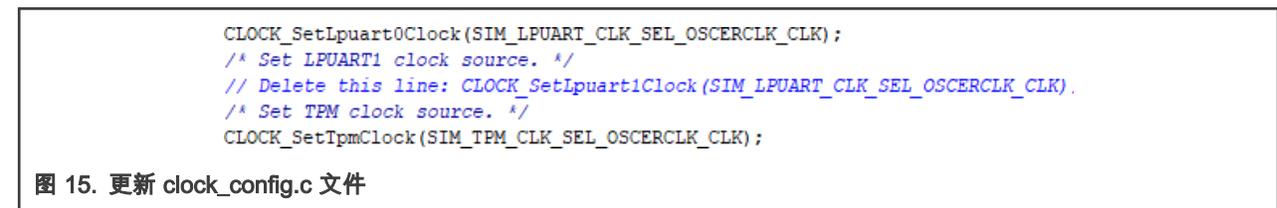


图 15. 更新 clock_config.c 文件

- 如有必要，打开工作区中源目录下的 app_preinclude.h 文件。此文件包含有关电路板的重要信息，如开关和 LED 的数量、计时器、功耗设置等。请检查并更新此文件以适应自己的自定义电路板。

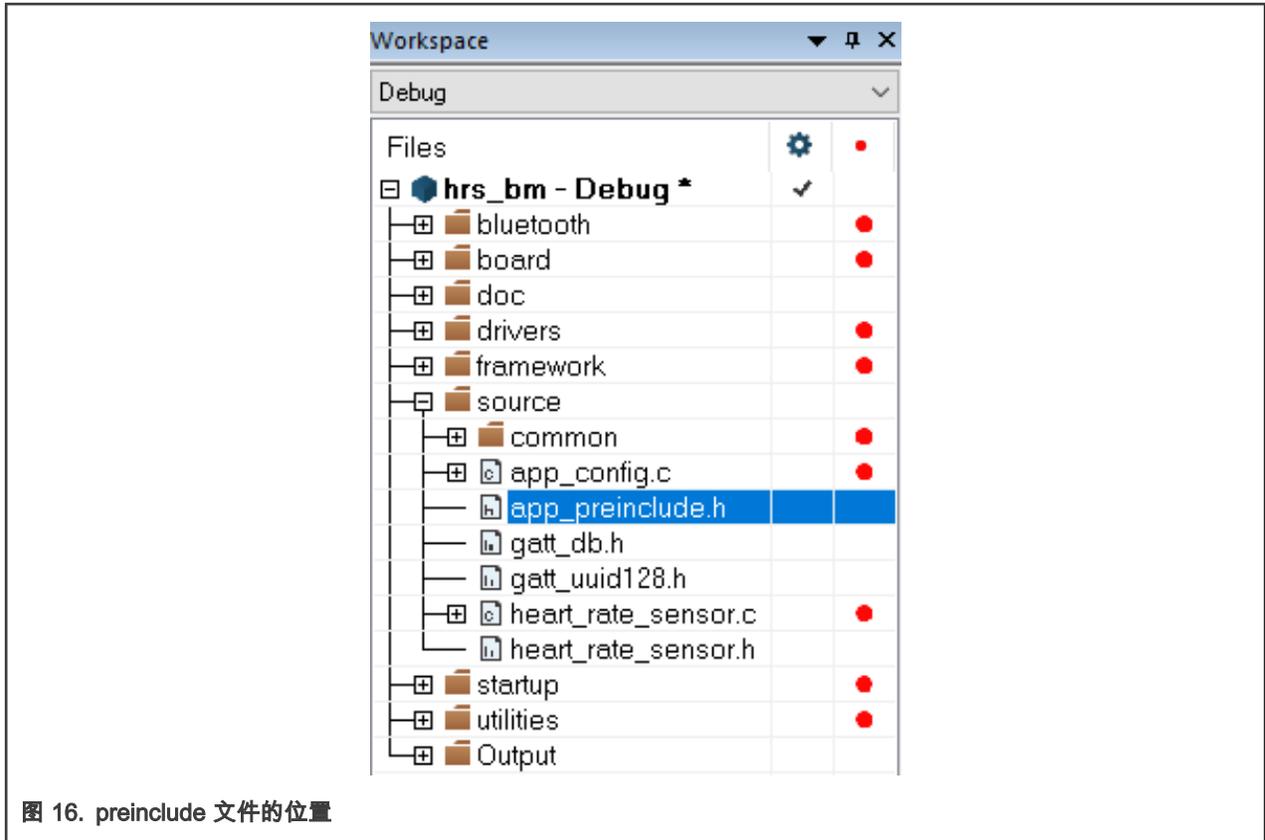


图 16. preinclude 文件的位置

12. 如有必要，在工作区的 board 目录下打开 pin mux 文件 (pin_mux.c 和 pin_mux.h) 和 gpio 文件 (gpio_pins.c 和 gpio_pins.h)。这些文件包含引脚的备选方案、选项和多路复用信息。检查和更新以适应自己的定制板。

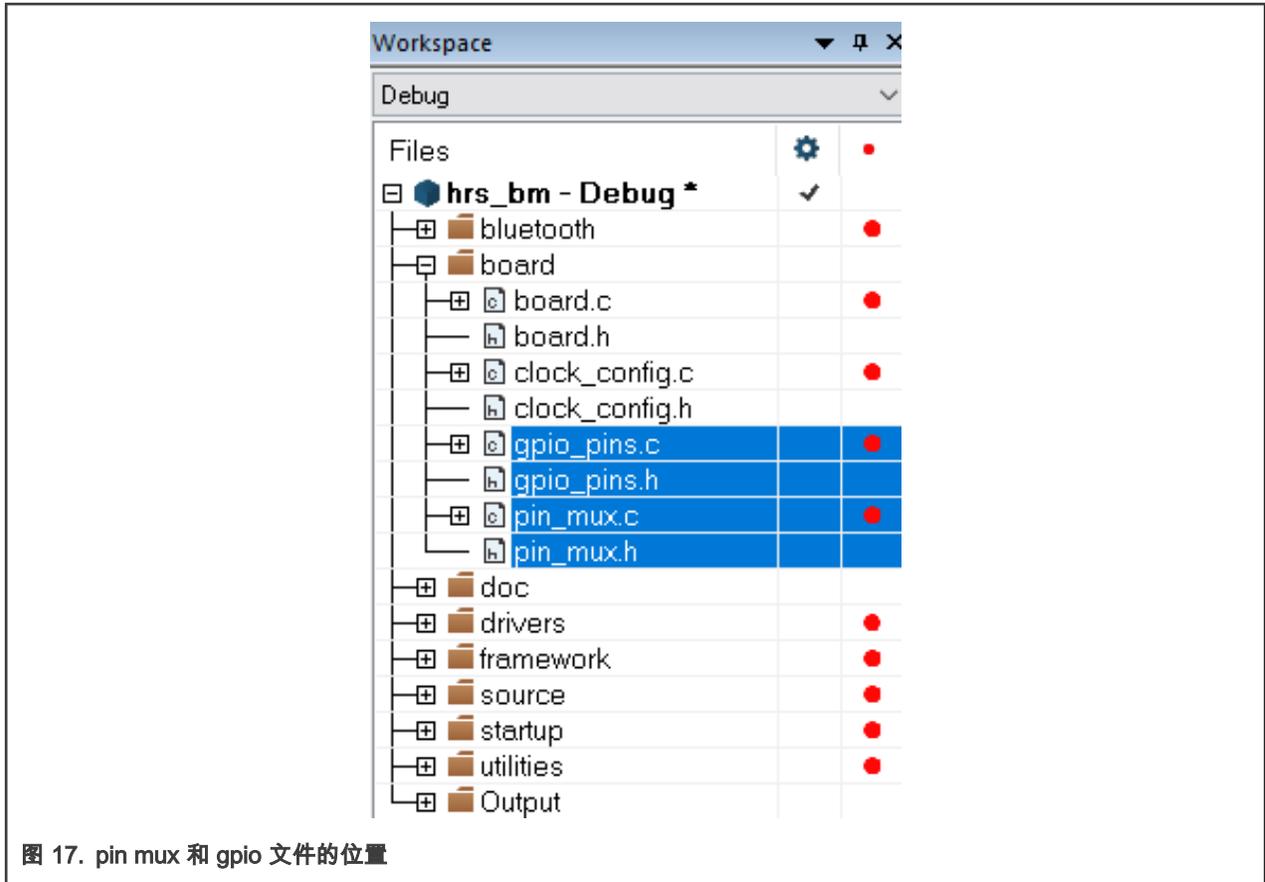


图 17. pin mux 和 gpio 文件的位置

13. 在以下位置保存 MKW35Z512xxx4_connectivity.icf 文件:

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

进入位于以下路径的心率传感器项目文件夹：

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

14. 打开“项目选项”窗口 (Alt+F7)。在 Linker/Config 窗口中，单击 Linker path 旁边的图标并选择链接器配置文件 MKW35Z512xxx4_connectivity.icf。

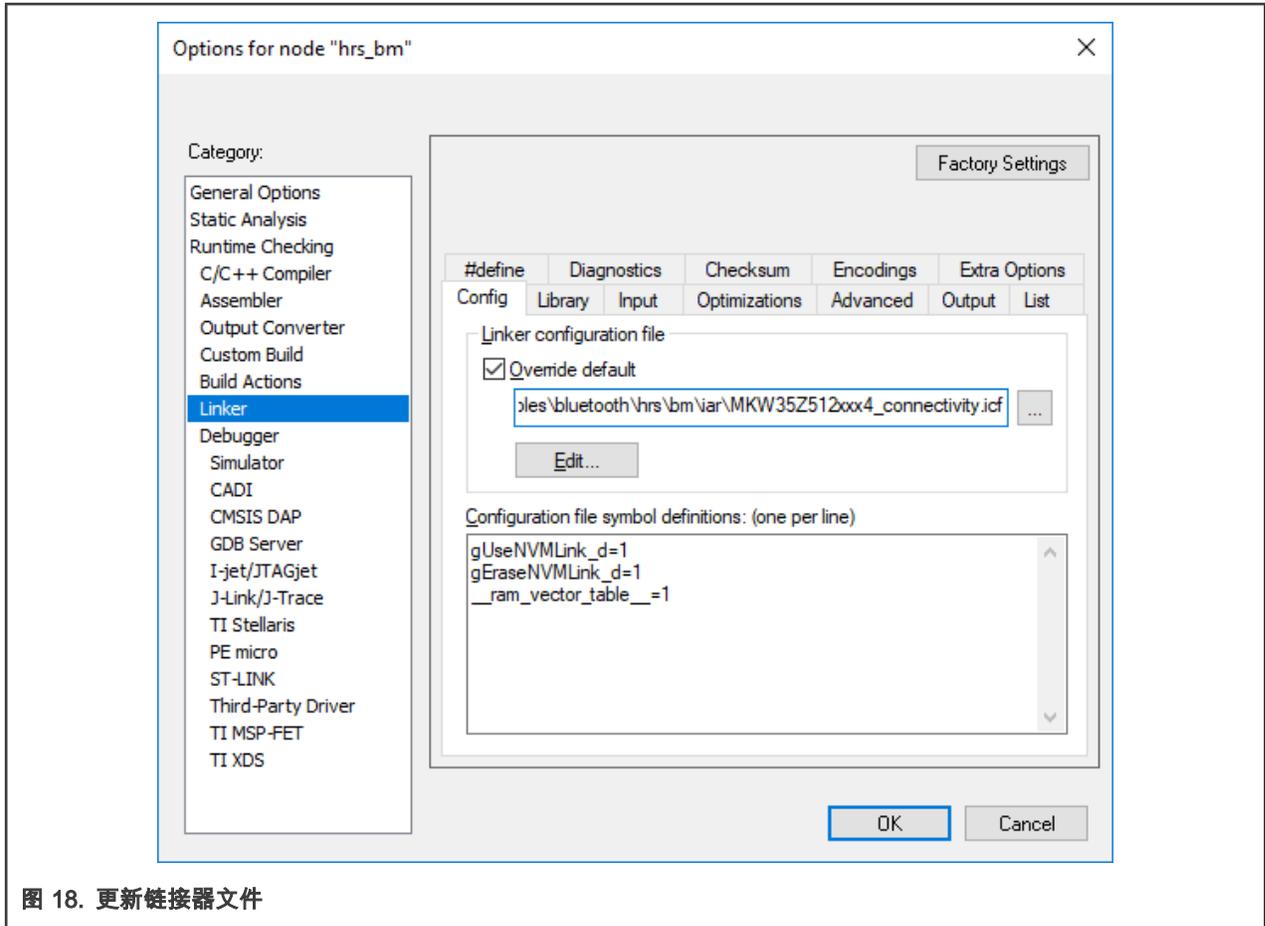


图 18. 更新链接器文件

15. 单击“项目选项”窗口中的“确定”按钮保存新配置。
16. 在工作区中选择项目，单击鼠标右键，然后按 clean。
17. 重建项目。至此，项目已经迁移完毕，您可以使用 MKW35 设备自定义应用程序。

5 MCUXpresso IDE 中的软件迁移

本章提供将 MKW36 示例代码迁移到 MCUXpresso IDE 中的 MKW35 MCU 所需的所有步骤。使用心率传感器项目作为基础，因为它有易于理解的示例，并且涉及低功耗蓝牙连接软件堆栈（包含在 SDK 中）。

5.1 项目属性所需的更改

1. 在左下角找到**快速启动**面板。

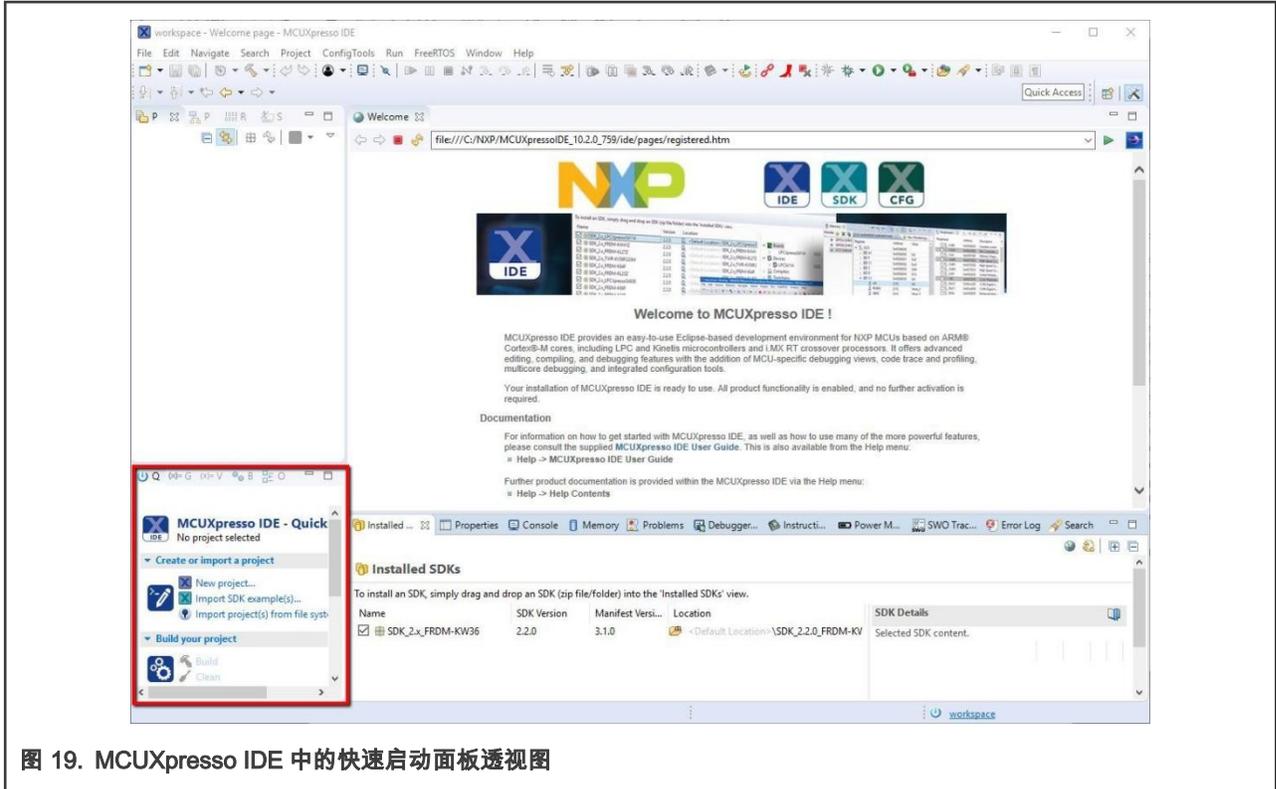


图 19. MCUXpresso IDE 中的快速启动面板透视图

2. 然后单击“导入 SDK 示例...”选项。

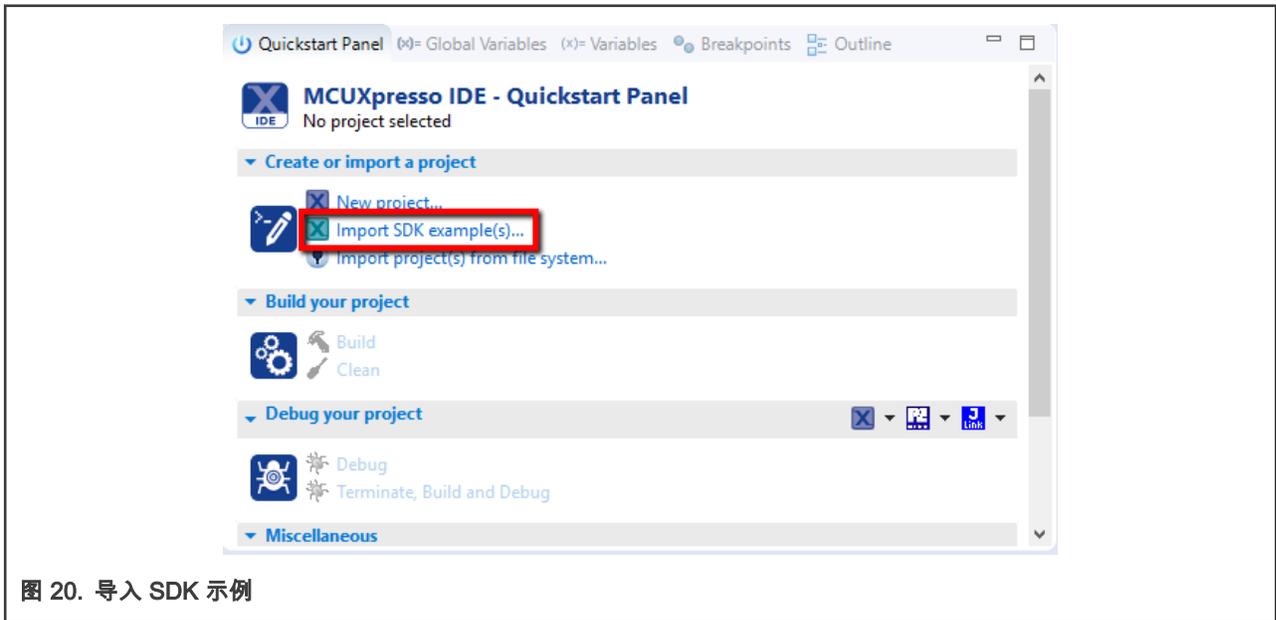


图 20. 导入 SDK 示例

3. 单击 frdmkw36 板，然后点击下一步。

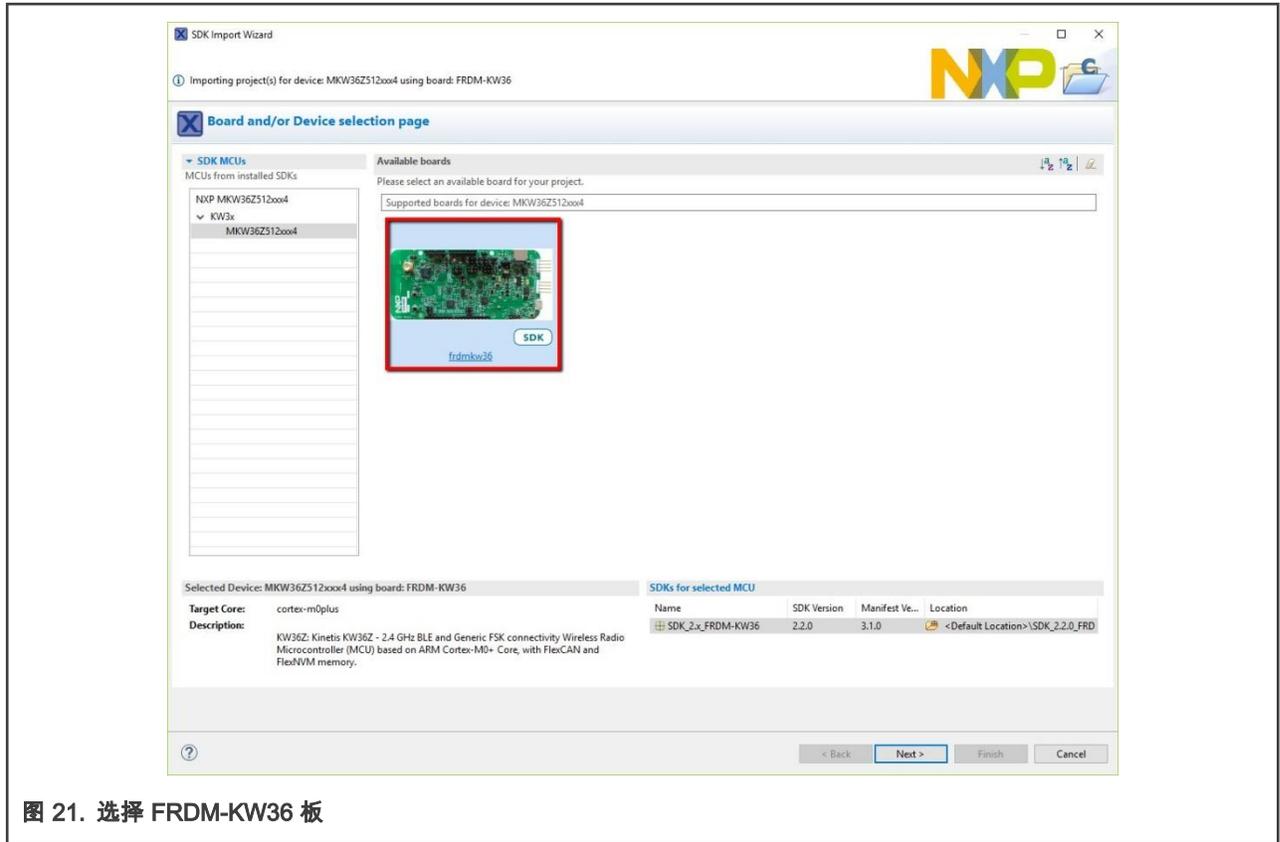


图 21. 选择 FRDM-KW36 板

- 在搜索文本框中，键入“hrs”以过滤示例项目。使用箭头按钮展开列表并找到“hrs”项目（wireless_examples -> 蓝牙 -> hrs），然后选择该项目的 **bm** 版本并单击“完成”。

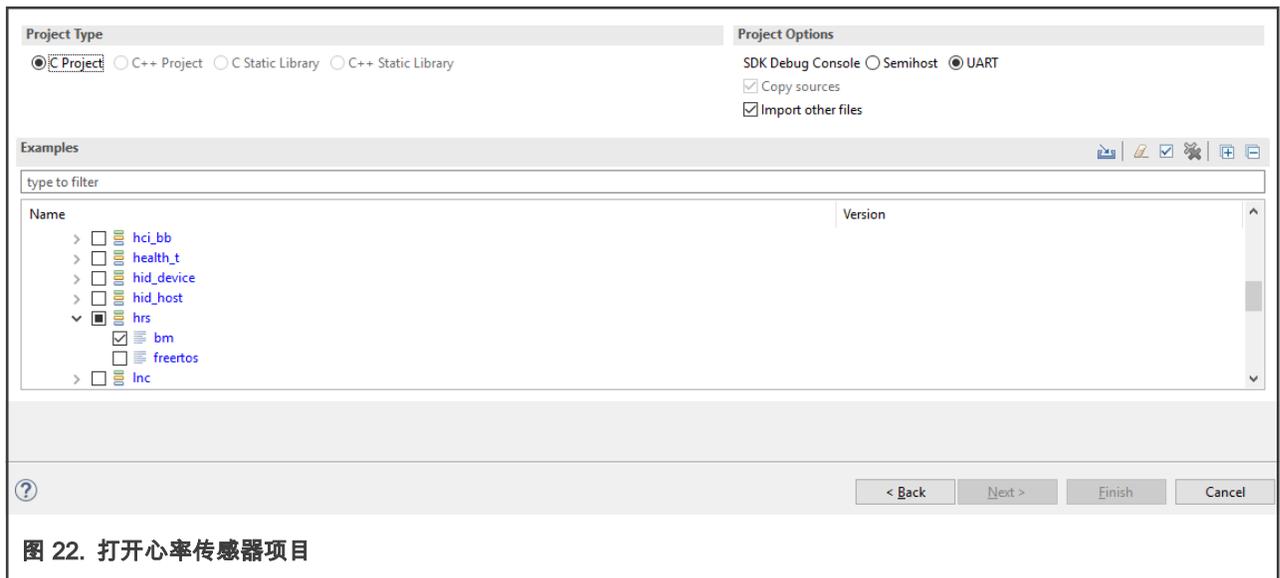


图 22. 打开心率传感器项目

- 转到 *Project/Properties*。展开 *C/C++ Build/MCU settings*，然后选择 MKW35Z512xxx4 MCU。单击“Apply and Close”按钮以保存配置。

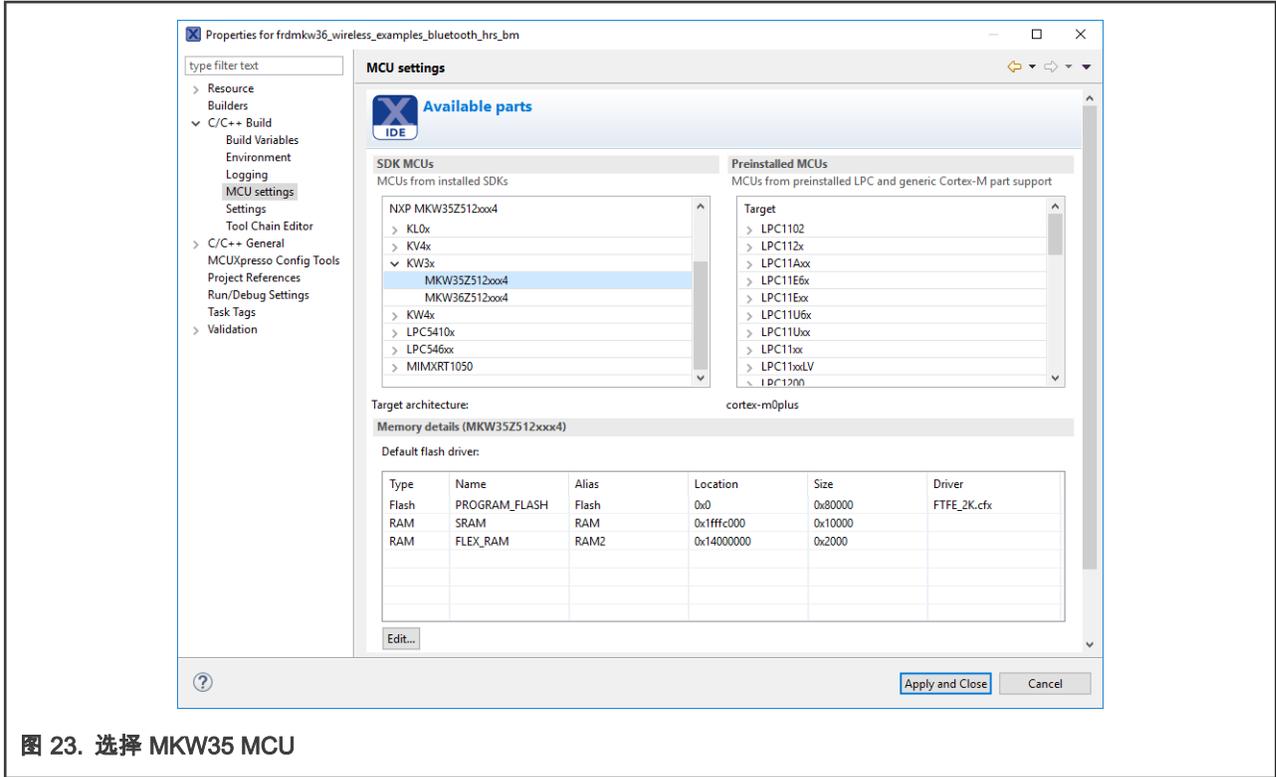


图 23. 选择 MKW35 MCU

- 在以下路径将 MKW36 文件夹重命名为 MKW35 (Documents/MCUXpressoIDE<version>/workspace 中的默认 MCUXpresso 根目录) :

```
<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
```

- 在工作区中选择心率传感器项目，然后按 F5 刷新修改的文件夹。

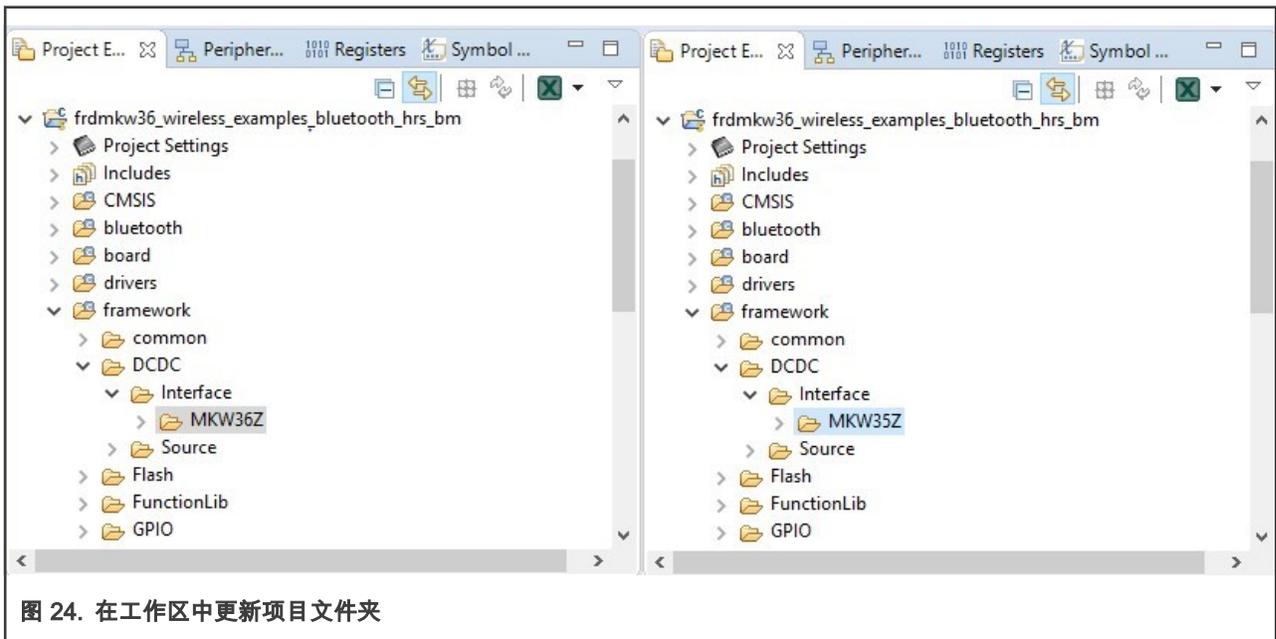


图 24. 在工作区中更新项目文件夹

8. 在 MCUXpresso IDE 中打开 “*Project/Properties*” 窗口。转到 *C/C++ Build/Settings*，然后在 “*Tool Settings*” 窗口中选择 “*MCU C Compiler/Includes*” 文件夹。在创建之前，根据 MKW35 文件夹编辑与 MKW36 MCU 相关的所有路径。结果必须类似如下所示：

```

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

```

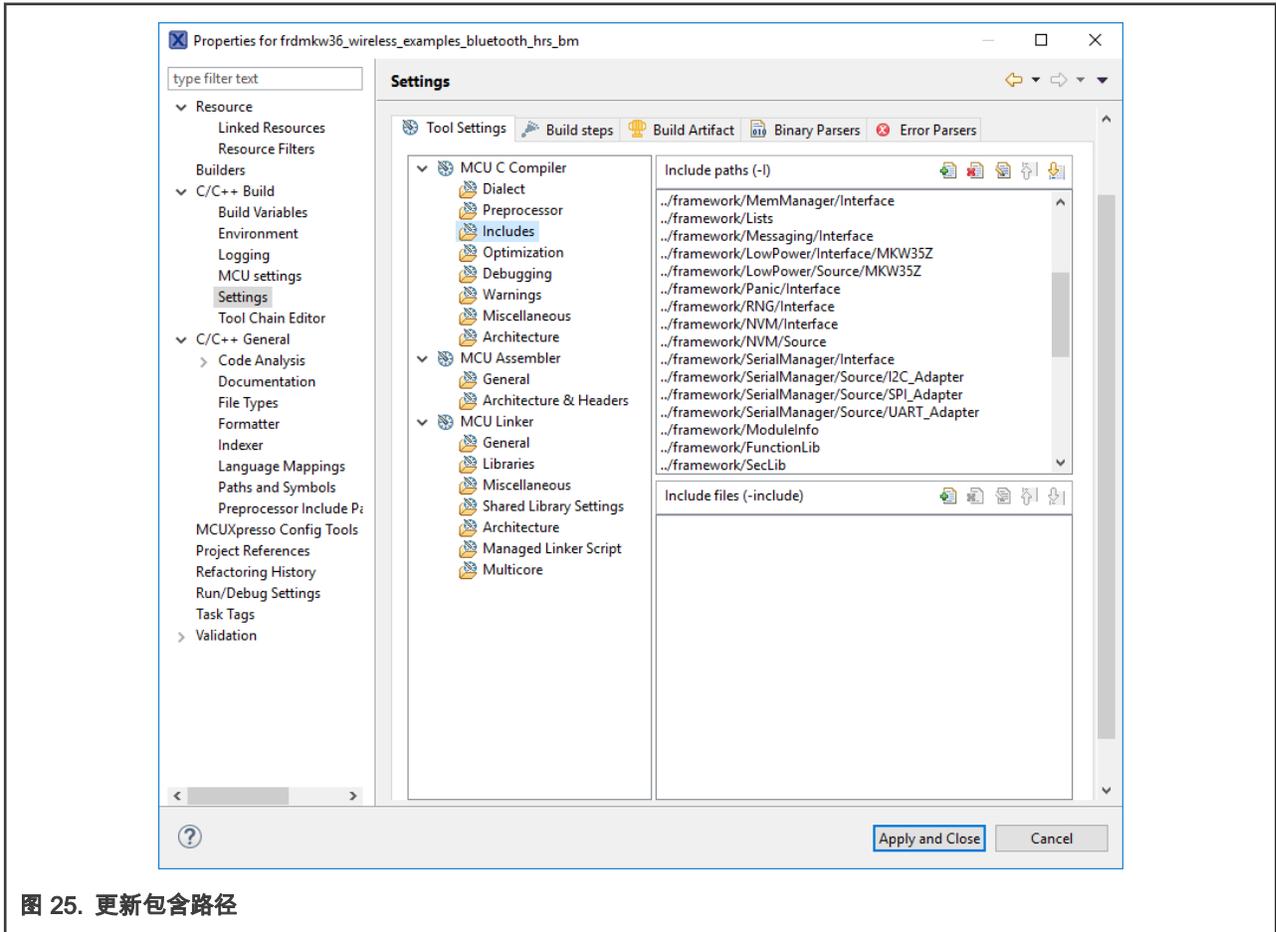


图 25. 更新包含路径

9. 在 “*Tool Settings (工具设置)*” 中选择 “*MCU Assembler/General*” 文件夹。编辑与 MKW36 MCU 相关的路径。结果必须类似如下所示：

```

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

```

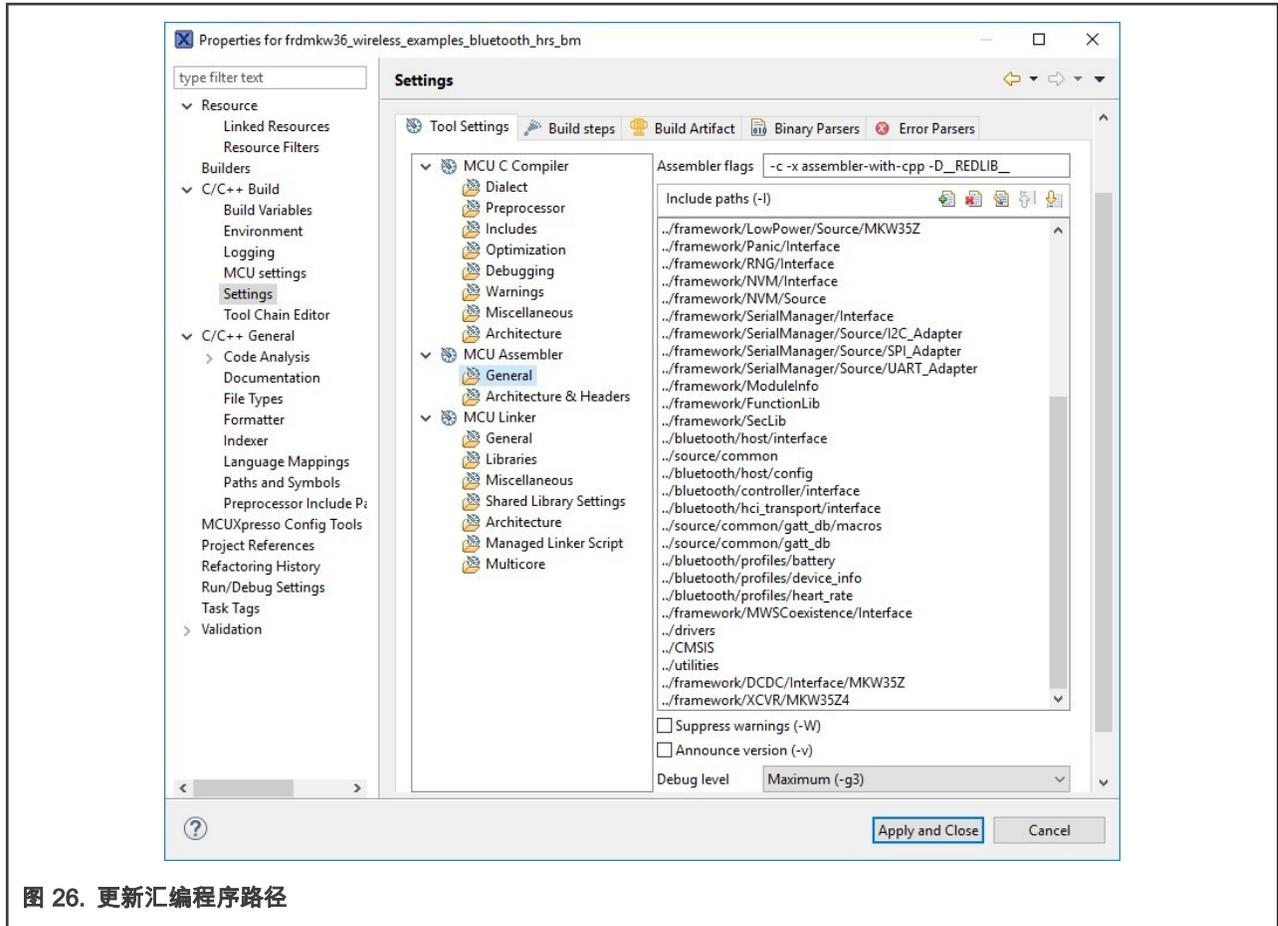


图 26. 更新汇编程序路径

- 打开 *MCU C Compiler/Preprocessor* 窗口，分别通过 CPU_MKW35Z512VHT4_cm0plus 和 CPU_MKW35Z512VHT4_cm0plus 编辑 CPU_MKW36Z512VHT4 和 CPU_MKW36Z512VHT4_cm0plus 宏。删除 FRDM_KW36 和 FREEDOM 宏。然后单击“Apply and Close”按钮。

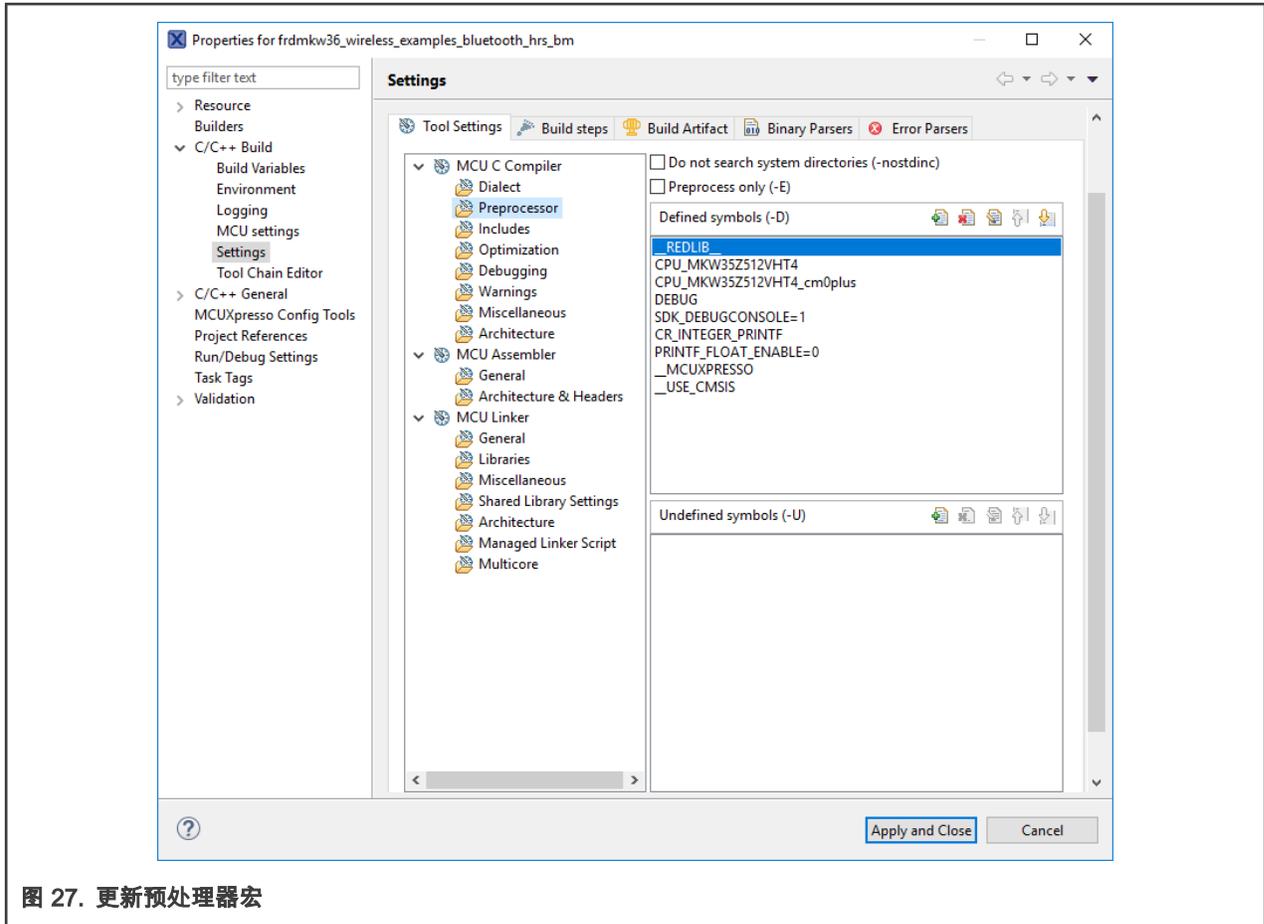


图 27. 更新预处理器宏

11. 删除位于工作区 CMSIS 文件夹中的 fsl_device_registers , MKW36Z4 , MKW36Z4_features , system_MKW36Z4.h 和 system_MKW36Z4.c 文件。
12. 解压缩 MKW35Z SDK 软件包 , 并在以下路径中搜索“ fsl_device_registers , MKW35Z4 , MKW35Z4_features , system_MKW35Z4.h 和 system_MKW35Z4.c”文件。

```
<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
```

将这些文件拖放到工作区中的 CMSIS 文件夹中。

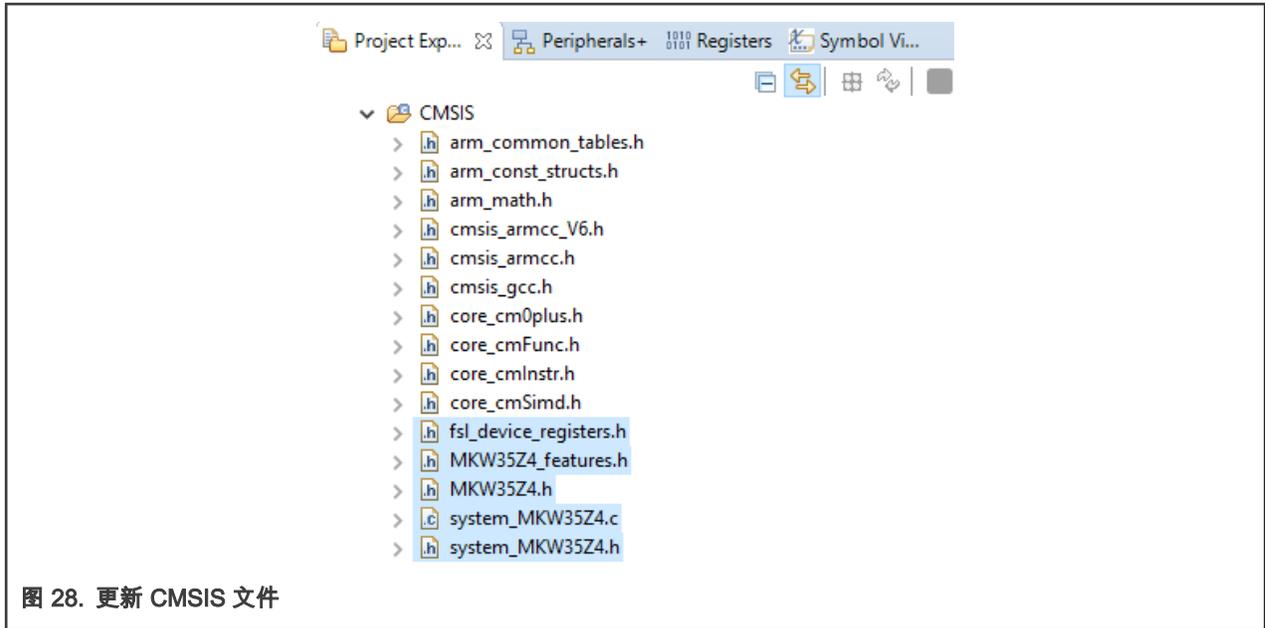


图 28. 更新 CMSIS 文件

- 用“startup_MKW35Z4.S”替换位于工作区中启动文件夹的“startup_MKW36Z.S”文件。与上一步相同，该文件位于 MKW35Z SDK 中的以下路径：

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

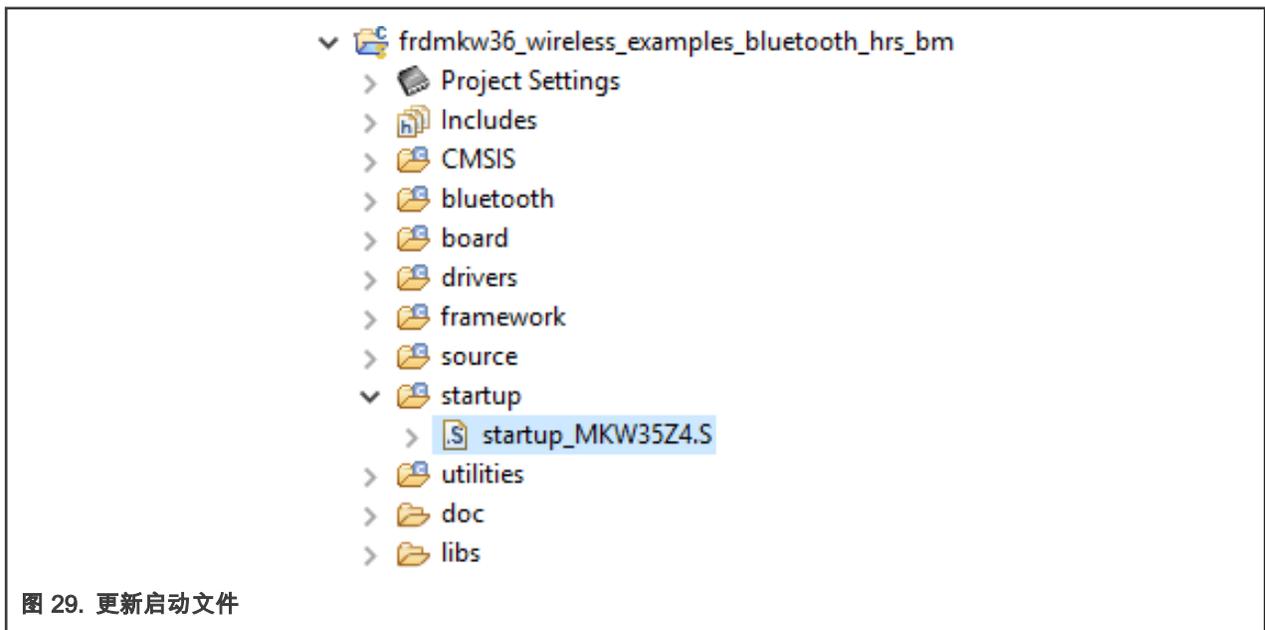


图 29. 更新启动文件

- 有些文件需要更新。转到 PWRLib.c 文件（在 workspace framework/LowPower/Source/MKW35Z 中）。在 PWRLib/lu StopUpdateWakeupReason 函数中将 LPUART0_LPUART1_IRQn 替换为 LPUART0_IRQn。

```

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;
    }
}

```

图 30. 更新 PWRLib.c file

15. 打开 board 文件夹中的 clock_config.c 文件。删除对 BOARD_BootClockRUN 函数中的 CLOCK_SetLpuart1Clock 调用。

```

/* 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);

```

图 31. 更新 clock_config.c file

16. 如果需要，请打开工作区中源目录下的 app_preinclude.h 文件。此文件包含有关电路板的重要信息，如开关和 LED 的数量、计时器、功耗设置等。请检查并更新此文件以适合自己的自定义电路板。

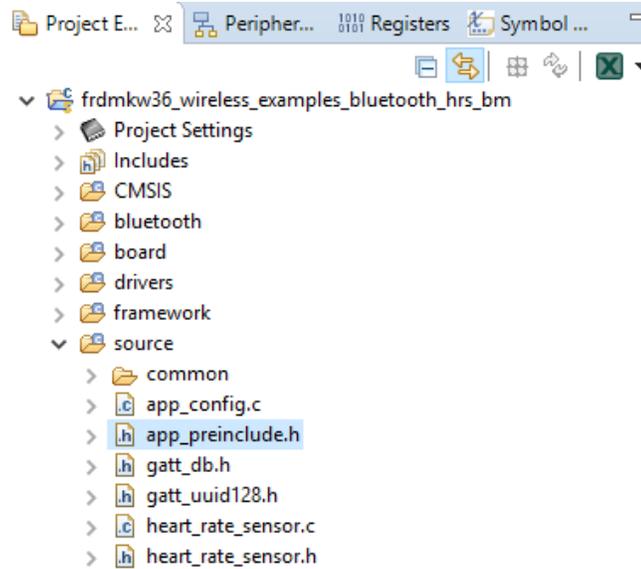


图 32. 预包含文件的位置

17. 如有必要，在工作区的 board 目录下打开 pin mux 文件 (pin_mux.c 和 pin_mux.h) 和 gpio 文件 (gpio_pins.c 和 gpio_pins.h)。这些文件包含引脚的替代品，选项和多路复用信息。检查并更新以适应自己的定制板。

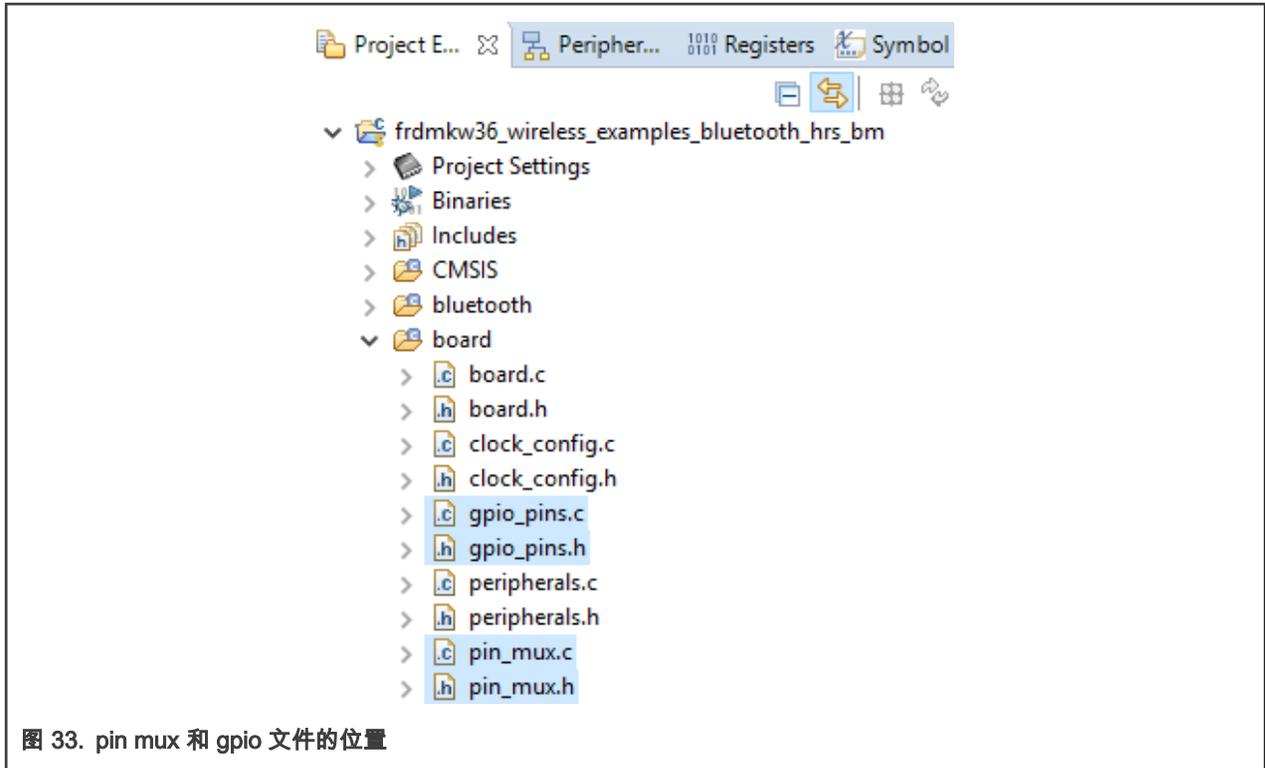


图 33. pin mux 和 gpio 文件的位置

18. 将 MKW36Z512xxx4_PD_connectivity.ld 文件替换为位于以下位置的 MKW35Z512xxx4_connectivity.ld :

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

进入工作区中的源文件夹中。

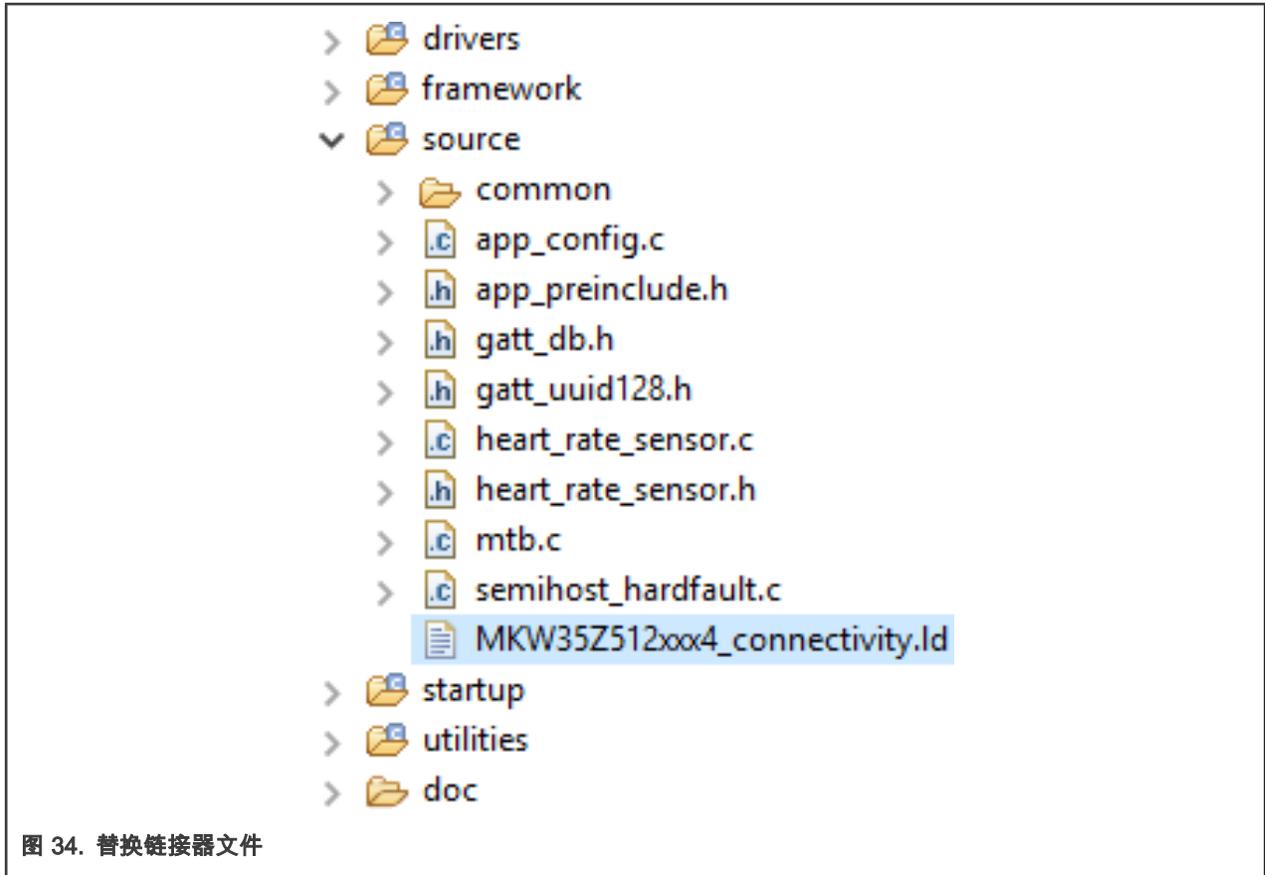


图 34. 替换链接器文件

19. 打开 *Project/Properties* 窗口。接下来，转到 *MCU Linker/Managed Linker Script* 透视图，并根据 MKW35 设备编辑链接器脚本名称。按下 **Apply and Close** 按钮。

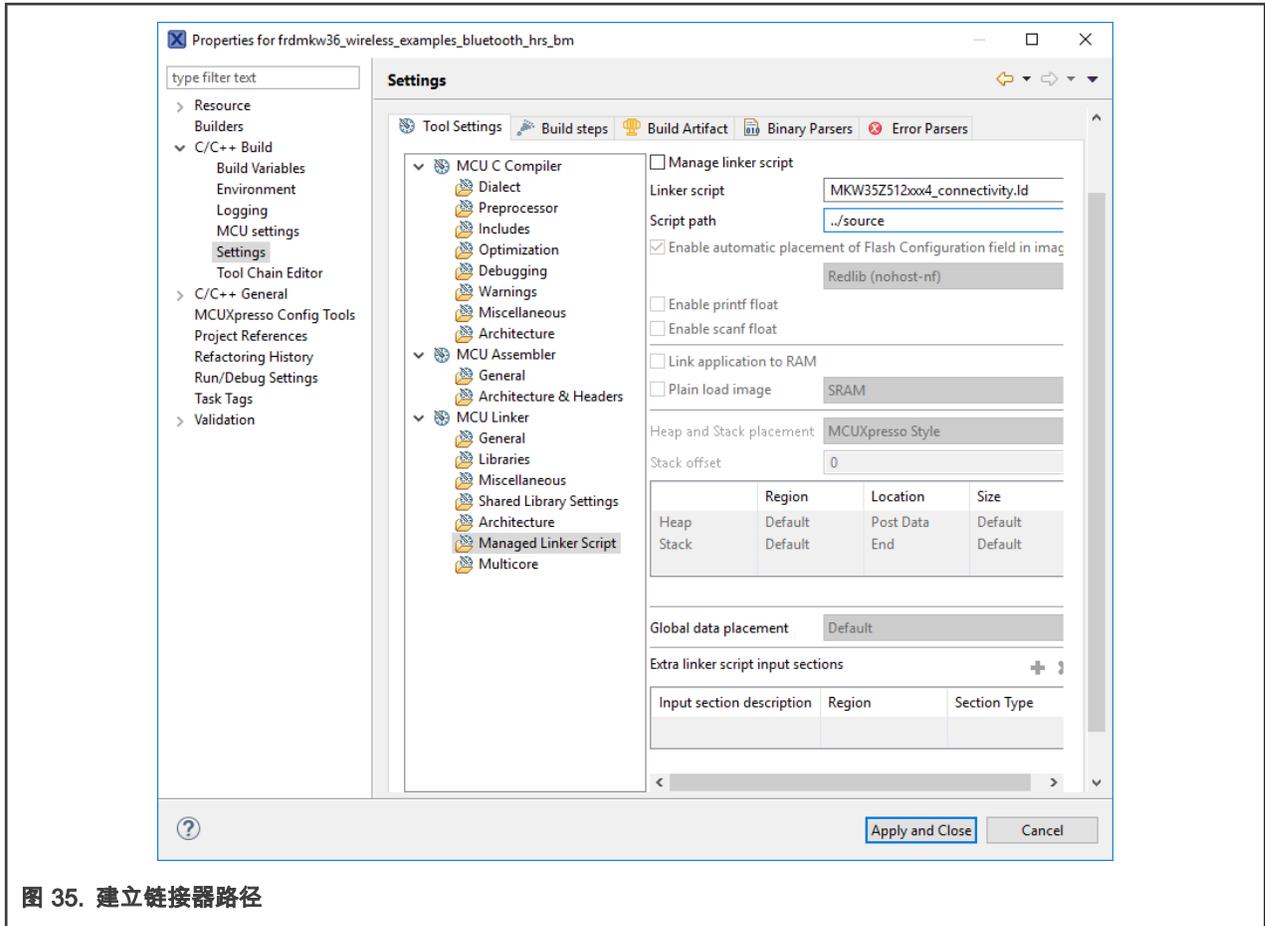


图 35. 建立链接器路径

20. 构建项目。至此，该项目已经迁移完毕，您可以使用 MKW35 进行演示了。

6 构建并运行低功耗蓝牙连接堆栈示例

经过本文档中所述的修改后，《低功耗蓝牙连接演示应用程序用户指南》（BLEDAUG）中引用的所有示例均与 MKW35 器件兼容。BLEDAUG 是 FRDM-KW36 SDK 文档的一部分。

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — 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.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D 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 marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2019-2021.

All rights reserved.

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

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

Date of release: 2019 年 6 月
Document identifier: AN12252

