AN14802

Migration Guide from MCX W71 to MCX W72

Rev. 2.0 — 17 July 2026 Application note

Document information

Information	Content
Keywords	AN14802, MCX W71, MCX W72, MCU, NBU, ISP, SPSDK, SDK
Abstract	This document describes how to migrate from MCX W71 to MCX W72 MCUs with emphasis on the connectivity software.

1 Introduction

This document describes how to migrate from MCX W71 to MCX W72 MCUs with emphasis on the connectivity software.

This document is intended for software engineers, software testers, software integrators, and customers designing their own hardware.

2 Hardware considerations

The MCX W71 wireless MCUs in 48-pin HVQFN packages are pin-to-pin compatible with MCX W72, and almost all peripherals are the same on both the devices.

The MCX W71 has 40-pin and 48-pin HVQFN packages.

Table 1 shows some of the similarities and differences between the two wireless MCUs.

Table 1. MCX W71/MCX W72 comparison

Features	MCX W71	MCX W72
Application core	Arm Cortex-M33 core of up to 96 MHz	Arm Cortex-M33 core of up to 96 MHz
Bluetooth Low Energy (Bluetooth LE) radio core	Dedicated Cortex-M3 core running at up to 64 MHz	Dedicated **Cortex-M33 core** running at up to 64 MHz
EdgeLock Secure Enclave core	Isolated Arm Cortex-M0+ core	Isolated Arm Cortex-M0+ core
Software	FreeRTOS, Zephyr Bluetooth LE 5.3, Thread, Zigbee	FreeRTOS, Zephyr **Bluetooth LE 6**, Thread, Zigbee **Bluetooth Channel Sounding**
Package	6x6 40 QFN, 22 GPIO 7x7 48 QFN, 29 GPIO with ‘wettable’ flanks (pin compatible with MCX W72)	7x7 48 QFN, 29 GPIO with ‘wettable’ flanks (pin compatible with MCX W71 48 QFN package)
Application memory (flash/SRAM)	8-kB code cache 1-MB flash memory 128-kB SRAM	**16-kB code cache** **2-MB flash memory** **256-kB SRAM**
Radio subsystem memory (flash/RAM)	256-kB radio flash 88-kB radio RAM	**512-kB radio flash** **170-kB radio RAM**
Bluetooth LE, radio core	Bluetooth LE 5.3 radio core, up to 24 simultaneous connections (125 kbit/s, 500 kbit/s, 1 Mbit/s, and 2 Mbit/s) in any central/peripheral combination Modulation types: GFSK, GMSK, FSK, MSK, OQPSK IEEE 802.15.4–2015 compliant radio	**Bluetooth LE 6.0 radio core**, up to 24 simultaneous connections (125 kbit/s, 500 kbit/s, 1 Mbit/s, and 2 Mbit/s) in any central/peripheral combination. Modulation types: GFSK, GMSK, FSK, MSK, OQPSK **Channel sounding** IEEE 802.15.4–2015 compliant radio
Radio TX power	Programmable transmit output power of up to +10 dBm	Programmable transmit output power of up to +10 dBm
Radio power (RX/TX @ 0 dBm)	Typical TX at 0 dBm: 4.6 mA	Typical TX at 0 dBm: 4.6 mA
Radio sensitivity	Bluetooth LE radio core: • 106-dBm 125-kbit/s long-range receive sensitivity • 102-dBm 500-kbit/s long-range receive sensitivity	Bluetooth LE radio core: • 106-dBm 125-kbit/s long-range receive sensitivity • 102-dBm 500-kbit/s long-range receive sensitivity

Table 1. MCX W71/MCX W72 comparison*...continued*

Features	MCX W71	MCX W72
	• 97.5-dBm 1-Mbit/s receive sensitivity • 95-dBm 2-Mbit/s receiver sensitivity IEEE 802.15.4 radio • 103-dBm 250-kbit/s receive sensitivity	• 97.5-dBm 1-Mbit/s receive sensitivity • 95-dBm 2-Mbit/s receiver sensitivity IEEE 802.15.4 radio • 103-dBm 250-kbit/s receive sensitivity
Communication interfaces	1x FlexCAN w/CAN FD support 2x LPUART 2x LPSPI 1x I3C 2x LPI2C supporting SMBus 1x FlexIO	**2x FlexCAN w/CAN FD support** 2x LPUART 2x LPI2C 2x LPSPI 1x I3C 1x FlexIO **1x LCE**
Clocks	32-MHz RF OSC 32.768-kHz OSC 192-MHz FRO 6-MHz FRO 32-kHz FRO	32-MHz RF OSC 32.768-kHz OSC 192-MHz FRO 32-kHz FRO
Security	EdgeLock Secure Enclave, core profile AES-128, AES-192, AES-255, supporting ECB, CBC, CTR AEAD (AES GCM, AES CCM) SHA-224, SHA-256, SHA-384, SHA-512 CMAC-AES, HMAC ECDSA P-224, P-256, P-384, P-521, X25519 TRNG with DRBG-CTR, key generation via DRBG Key derivation: LTK and session key, ECDH(E) (P-224, P-256, P-384, P-521, Ed25519) SESIP L2/PSA L2/RED, Tamper protection, Trust Provisioning	EdgeLock Secure Enclave, core profile AES-128, AES-192, AES-256 – supporting ECB, CBC, CTR, AEAD (AES GCM, AES CCM) **SHA-1**, SHA-224, SHA-256, SHA-384, SHA-512, **SHA-3** **Multipart hashing, 4 operations in parallel, SHA2xx, SHA3 context import/export, and context clone** CMAC-AES, HMAC, **Multipart MAC, four operations in parallel** ECDSA P-224, P-256, P-384, P-521, **Brainpool**, X25519 TRNG with DRBG-CTR, Key generation via DRBG Key derivation: LTK and Session Key, **CKDF (CTR), HKDF, SPAKE2+ (Matter)**, ECDH(E) (P-224, P256, P-384, P-521, Ed25519, **Brainpool**), **NBU Advance Key Protection**, SESIP L2/PSA L2/RED, Tamper Protection, Trust Provisioning, **ISO 21434**.

2.1 Peripherals instantiation

The MCX W71 devices in 48-pin HVQFN packages are pin-to-pin compatible with the MCX W72 devices. The MCX W71 devices are also available in 40-pin ‘wettable’ HVQFN packages.

The main difference between the MCX W71 and MCX W72 is the RF, UART, and CAN support pins on the MCX W72.

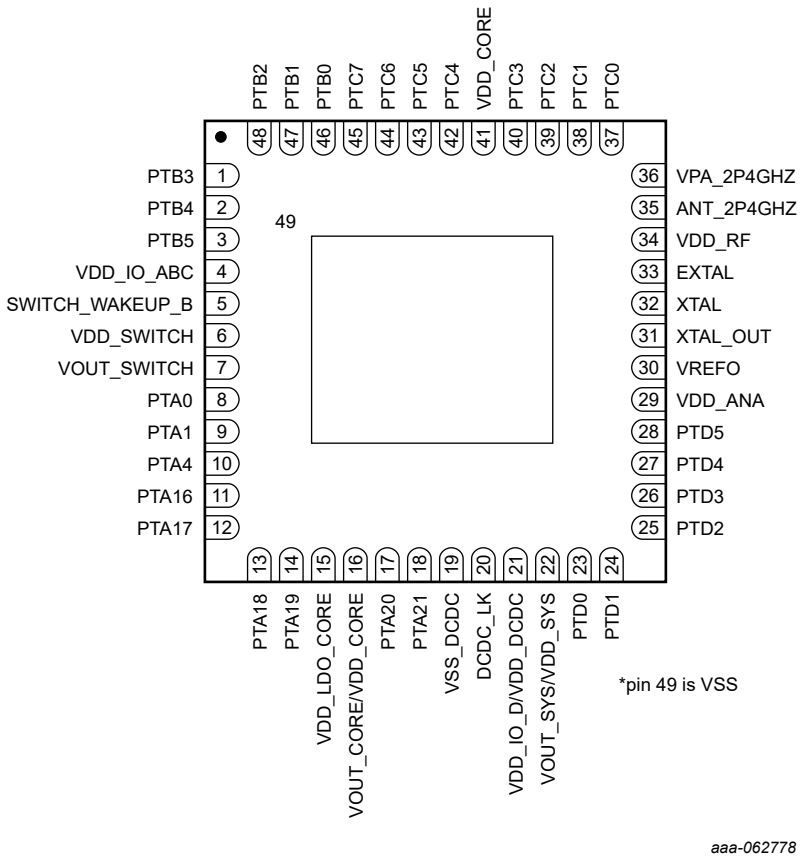


Figure 1. MCX W71 48-pin HVQFN package pinout diagram with radio modules

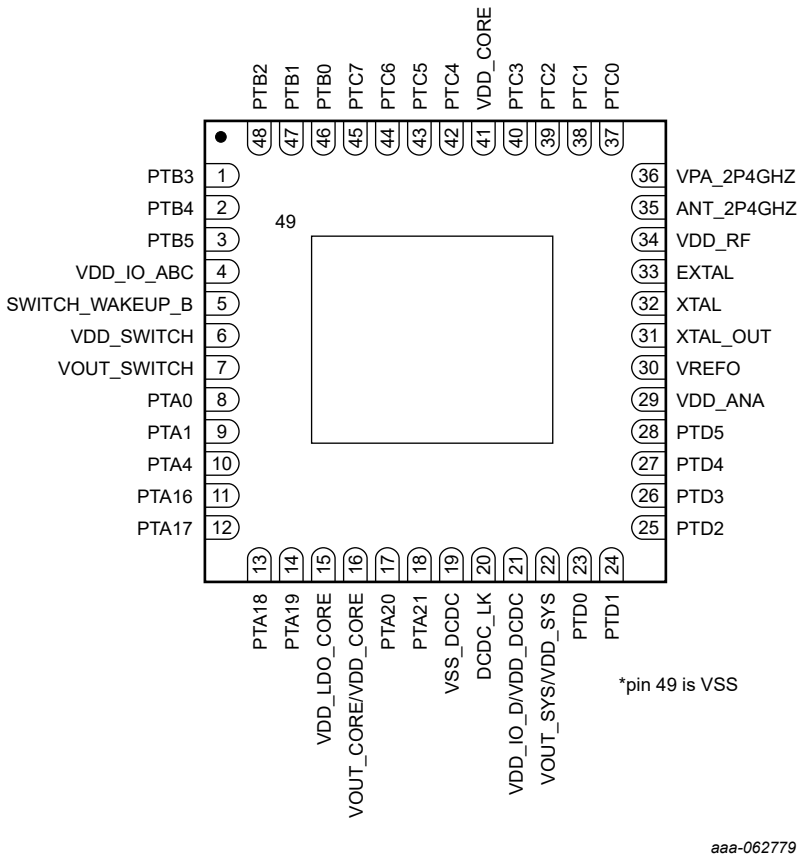


Figure 2. MCX W72 48-pin HVQFN package pinout diagram with radio modules

Table 2 specifies the changes implemented on the package pinout table on MCX W71 and MCX W72.

Note: *The **bold** alternatives are available only for the MCX W72.*

Table 2. MCX W71/MCX W72 instance comparative

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
40	1	PTB3	ALT0 -ADC0_B13 ALT1 - PTB3 ALT2 - LPSPI1_SOUT ALT3 - LPUART1_RX ALT5 - TPM1_CH3 ALT9 - FLEXIO0_D29 VDD SYS - WUU0_P14
1	2	PTB4	ALT1 - PTB4 ALT2 - LPSPI1_PCS3 ALT3 - LPUART1_CTS_b ALT4 - LPI2C1_SDA ALT5 - I3C0_SDA ALT6 - TRGMUX0_IN0 ALT9 - FLEXIO0_D30 VDD SYS - WUU0_P15
2	3	PTB5	ALT1 - PTB5

Table 2. MCX W71/MCX W72 instance comparative*...continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
			ALT2 - LPSPI1_PCS2 ALT3 - LPUART1_RTS_b ALT4 - LPI2C1_SCL ALT5 - I3C0_SCL ALT6 - TRGMUX0_OUT0 ALT9 - FLEXIO0_D31
3	4	VDD_IO_ABC	ALT0 - VDD_IO_ABC
4	5	SWITCH_WAKEUP_B	ALT0 - SWITCH_WAKEUP_B
5	6	VDD_SWITCH	ALT0 - VDD_SWITCH
6	7	VOUT_SWITCH	ALT0 - VOUT_SWITCH
7	8	PTA0	ALT1 - PTA0 ALT2 - CMP0_OUT ALT3 - LPUART0_CTS_b ALT4 - RF_GPO_11 ALT5 - TPM0_CH4 ALT6 - FLEXIO0_D0 ALT7 - SWD_DIO VDD SYS - WUU0_P0
8	9	PTA1	ALT1 - PTA1 ALT2 - CMP1_OUT ALT3 - LPUART0_RTS_b ALT4 - RF_GPO_10 ALT5 - TPM0_CH5 ALT6 - FLEXIO0_D1 ALT7 - SWD_CLK
9	10	PTA4	ALT0 - ADC0_A10/CMP0_IN0 ALT1 - PTA4 ALT3 - RF_GPO_9 ALT4 - TPM0_CLKIN ALT5 - TRACE_SWO ALT6 - FLEXIO0_D4 ALT7 - BOOT_CONFIG VDD SYS - WUU0_P2/ RF_XTAL_OUT_ENABLE
-	11	PTA16	ALT0 - ADC0_A12 ALT1 - PTA16 ALT2 - LPSPI0_PCS0 ALT3 - EWM0_OUT_b ALT4 - LPI2C0_SCLS ALT5 - TPM0_CH4 ALT6 - LPUART0_RX ALT7 - RF_GPO_8 ALT9 - FLEXIO0_D5 VDD SYS - WUU0_P19/ RF_NOT_ALLOWED

Table 2. MCX W71/MCX W72 instance comparative*...continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
10	12	PTA17	ALT0 - ADC0_A13 ALT1 - PTA17 ALT2 - LPSPI0_SIN ALT3 - EWM0_IN ALT4 - LPI2C0_SDAS ALT5 - TPM0_CH5 ALT6 - LPUART0_TX ALT7 - RF_GPO_7 ALT8 - RF_GPO_8 ALT9 - FLEXIO0_D6 ALT11 - RF_EXT_XTAL_REQUES T/RF_GPO_7 VDD SYS - WUU0_P3/ RF_NOT_ALLOWED
11	13	PTA18	ALT0 - CMP1_IN1 ALT1 - PTA18 ALT2 - LPSPI0_SOUT ALT3 - LPUART0_CTS_b ALT4 - LPI2C0_SDA ALT5 - TPM0_CH3 ALT6 - RF_GPO_0 ALT10 - LPUART0_RX ALT11 - SPC0_LPREQ **VDD SYS - WUU0_P20**
12	14	PTA19	ALT0 - CMP1_IN0 ALT1 - PTA19 ALT2 - LPSPI0_SCK ALT3 - LPUART0_RTS_b ALT4 - LPI2C0_SCL ALT5 - TPM0_CH2 ALT6 - RF_GPO_1 VDD SYS - WUU0_P4
13	15	VDD_LDO_CORE	ALT0 - VDD_LDO_CORE
14	16	VDD_CORE	ALT0 - VOUT_CORE
15	17	PTA20	ALT0 - ADC0_A14/CMP0_IN3 ALT1 - PTA20 ALT2 - LPSPI0_PCS2 ALT3 - LPUART0_TX ALT4 - EWM0_IN ALT5 - TPM0_CH1 ALT6 - RF_GPO_2 ALT8 - FLEXIO0_D7 **ALT11 - LPUART0_RTS_b**
16	18	PTA21	ALT0 - ADC0_A15/CMP0_IN2

Table 2. MCX W71/MCX W72 instance comparative*...continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
			ALT1 - PTA21 ALT2 - LPSPI0_PCS3 ALT3 - LPUART0_RX ALT4 - EWM0_OUT_b ALT5 - TPM0_CH0 ALT6 - RF_GPO_3 ALT7 - RF_GPO_7 ALT8 - FLEXIO0_D8 ALT9 - RF_GPO_10 **ALT11 - LPUART0_CTS_b** VDD SYS - WUU0_P5
17	19	VSS_DCDC	ALT0 - VSS_DCDC
18	20	DCDC_LX	ALT0 - DCDC_LX
19	21	VDD_IO_D	ALT0 - VDD_IO_D
20	22	VDD_SYS	ALT0 - VOUT_SYS/VDD_SYS
21	23	PTD0	ALT0 - ADC0_A5 ALT1 - PTD0 ALT3 - RESET_b
-	24	PTD1	ALT0 - ADC0_B5 ALT1 - PTD1 ALT2 - SPC0_LPREQ ALT3 - NMI_b ALT4 - RF_GPO_4 **ALT5 - RF_GPO_7** **ALT7 - LPTMR2_TRIG_OUT_b**
-	25	PTD2	ALT0 - ADC0_A6 ALT1 - PTD2 ALT2 - LPTMR0_ALT3 ALT3 - TAMPER0 ALT4 - RF_GPO_5 **ALT5 - TPM2_CH0** **ALT7 - LPTMR1_TRIG_OUT_b**
-	26	PTD3	ALT0 - ADC0_B6 ALT1 - PTD3 ALT2 - LPTMR1_ALT3 ALT3 - TAMPER1 ALT4 - RF_GPO_6 **ALT5 - TPM2_CH1** ALT6 - TRGMUX0_IN2 **ALT7 - LPTMR0_TRIG_OUT_b**
22	27	PTD4	ALT0 - XTAL32K ALT1 - PTD4 ALT2 - LPTMR0_ALT2 ALT3 - TAMPER2

Table 2. MCX W71/MCX W72 instance comparative*...continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
23	28	PTD5	ALT0 - EXTAL32K ALT1 - PTD5 ALT2 - LPTMR1_ALT2
24	29	VDD_ANA	ALT0 - VDD_ANA
25	30	VREF_OUT	ALT0 - VREFO
-	31	XTAL_OUT	ALT0 - XTAL_OUT
26	32	XTAL	ALT0 – XTAL
27	33	EXTAL	ALT0 - EXTAL
28	34	VDD_RF	ALT0 - VDD_RF
29	35	ANT_2P4GHZ	ALT0 - ANT_2P4GHZ
30	36	VPA_2P4GHZ	ALT0 - VPA_2P4GHZ
-	37	PTC0	ALT1 - PTC0 ALT2 - LPSPI1_PCS2 ALT3 - CAN0_TX [1] ALT4 - I3C0_SDA ALT5 - TPM1_CH0 ALT7 - LPI2C1_SCL ALT9 - FLEXIO0_D16 VDD SYS - WUU0_P7
-	38	PTC1	ALT1 - PTC1 ALT2 - LPSPI1_PCS3 ALT3 - CAN0_RX ALT4 - I3C0_SCL ALT5 - TPM1_CH1 ALT7 - LPI2C1_SDA ALT9 - FLEXIO0_D17 VDD SYS - WUU0_P8
31	39	PTC2	ALT1 - PTC2 ALT2 - LPSPI1_SOUT ALT3 - LPUART1_RX ALT4 - LPI2C1_SCLS ALT5 - TPM1_CH2 ALT7 - I3C0_PUR ALT9 - FLEXIO0_D18 **ALT11 - CAN1_RX** VDD SYS - WUU0_P9
32	40	PTC3	ALT1 - PTC3 ALT2 - LPSPI1_SCK ALT3 - LPUART1_TX ALT4 - LPI2C1_SDAS ALT5 - TPM1_CH3 ALT9 - FLEXIO0_D19 **ALT11 - CAN1_TX**

Table 2. MCX W71/MCX W72 instance comparative*...continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
33	41	VDD_CORE	ALT0 - VDD_CORE
34	42	PTC4	ALT1 - PTC4 ALT2 - LPSPI1_SIN ALT3 - CAN0_TX [1] ALT4 - LPI2C1_SCL ALT6 - TPM2_CH0 ALT9 - FLEXIO0_D20 VDD SYS - WUU0_P10
35	43	PTC5	ALT1 - PTC5 ALT2 - LPSPI1_PCS0 ALT3 - CAN0_RX [1] ALT4 - LPI2C1_SDA ALT5 - TPM1_CH4 ALT6 - TPM2_CH1 ALT9 - FLEXIO0_D21 VDD SYS - WUU0_P22
-	44	PTC6	ALT0 - ADC0_A8 ALT1 - PTC6 ALT2 - LPSPI1_PCS1 ALT5 - TPM1_CH5 ALT9 - FLEXIO0_D22 **ALT11 - CAN1_RX** VDD SYS - WUU0_P11
36	45	PTC7	ALT0 - DISABLED ALT1 - PTC7 ALT2 - TRGMUX0_IN3 ALT3 - TRGMUX0_OUT3 ALT4 - SFA0_CLK ALT5 - TPM1_CLKIN ALT6 - TPM2_CLKIN ALT7 - CLKOUT ALT9 - FLEXIO0_D23 **ALT10 - NMI_b** **ALT11 - CAN1_TX** VDD SYS - WUU0_P12/ NMI_b/ RF_NOT_ALLOWED
37	46	PTB0	ALT0 - ADC0_B10 ALT1 - PTB0 ALT2 - LPSPI1_PCS0 **ALT3 - CAN1_RX** ALT5 - TPM1_CH0 ALT9 - FLEXIO0_D26 VDD SYS - WUU0_P13
38	47	PTB1	ALT0 - ADC0_B11

Table 2. MCX W71/MCX W72 instance comparative...*continued*

MCX W71 40HVQFN	MCX W71/MCX W72 48HVQFN	Pin name	Pin mux assignment
			ALT1 - PTB1 ALT2 - LPSPI1_SIN **ALT3 - CAN1_TX** ALT5 - TPM1_CH1 ALT9 - FLEXIO0_D27
39	48	PTB2	ALT0 - ADC0_B12 ALT1 - PTB2 ALT2 - LPSPI1_SCK ALT3 - LPUART1_TX ALT5 - TPM1_CH2 ALT9 - FLEXIO0_D28
41	49	VSS	ALT0 - VSS

Table 2 is not a full description of the MCX W71/72 pinout. For more details, see the MCX W71 and MCX W72 data sheets.

3 Memory

This section describes the memory comparison between the MCX W71 and MCX W72 devices.

3.1 System memory map

Table 3 describes the system memory map for the MCX W71 and MCX W72 devices, including memory regions, start addresses, sizes, security attributes, and cache configurations.

Table 3. System memory map for MCX W71 and MCX W72

Description	MCX W71		MCX W72		Default	Cached
	Start address	Size	Start address	Size		
Program flash	0000_0000	**1024 kB**	0000_0000	**2 MB**	Nonsecure	Code cache
Reserved	**0010_0000**	—	**0020_0000**	—		—
Flash Logical Window	0100_0000	**1024 kB**	0100_0000	**2 MB**		Code cache
Reserved	**0110_0000**	—	**0120_0000**	—		—
Tightly Coupled Memory - Code	0400_0000	**16 kB**	0400_0000	**32 kB**		No
Reserved	**0400_4000**	—	**0400_8000**	—		—
Program flash	1000_0000	**1024 kB**	1000_0000	**2 MB**	Secure	Code cache
Reserved	**1010_0000**	—	**1020_0000**	—		—
Flash logical window	1100_0000	**1024 kB**	1100_0000	**2 MB**		Code cache
Reserved	**1110_0000**	—	**1120_0000**	—		—
Tightly coupled memory- code	1400_0000	**16 kB**	1400_0000	**32 kB**		Code cache
Reserved	**1400_4000**	—	**1400_8000**	—		—

Table 3. System memory map for MCX W71 and MCX W72*...continued*

Description	MCX W71		MCX W72		Default	Cached
	Start address	Size	Start address	Size		
Tightly coupled memory- system	2000_0000	**112 kB**	2000_0000	**232 kB**	Nonsecure	Code cache
Reserved	**2001_C000**	—	**2003_A000**	—		—
Tightly coupled memory- system	3000_0000	**112 kB**	3000_0000	**232 kB**	Secure	Code cache
Reserved	**3001_C000**	—	**3003_A000**	—		—

Note: *Table 3 does not contain the entire memory map. For more details, refer to the MCX W71 and MCX W72 reference manuals.*

3.2 ROM

The internal ROM size of the MCX W71 and MCX W72 devices is 96 kB. It supports automatic booting from the internal flash and enables image download through serial interfaces (LPUART, LPI2C, and LPSPI).

These devices support the following ROM bootloader operating frequencies:

- Normal boot: MCX W71 (32 MHz), MCX W72 (48 MHz)
- Fast boot: MCX W71 (96 MHz), MCX W72 (96 MHz)

The ROM behavior varies across different lifecycle states. The lifecycle fuse must be programmed correctly; otherwise, the ROM bootloader does not boot.

3.2.1 ROM bootloader memory usage comparative

The Figure 3 and Figure 4 compare the ROM bootloader memory usage of MCX W71 and MCX W72.

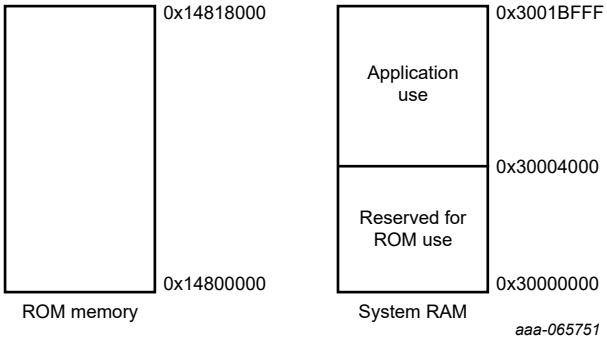


Figure 3. MCXW71 ROM bootloader memory usage

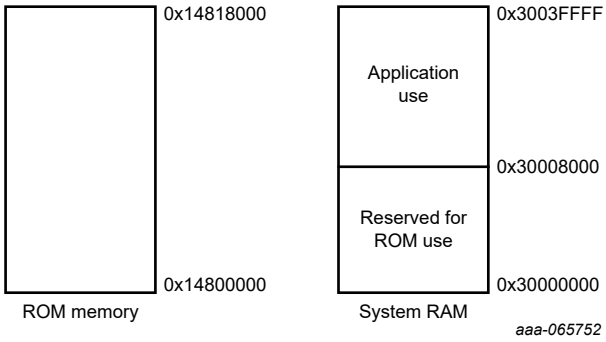


Figure 4. MCXW72 ROM bootloader memory usage

3.2.2 Lifecycle and fuses

Table 4 summarizes the lifecycle states and corresponding fuse configurations for the MCX W71 and MCX W72 devices, including debug authentication requirements.

Table 4. Lifecycle and fuses

Lifecycle	Lifecycle state [0:7]	Debug authentication required to debug CM33		Debug authentication required to debug the radio core	
		MCX W71	MCX W72	MCX W71	MCX W72
0000_0000	Blank	N/A	N/A	N/A	N/A
0000_0111	OEM Open	No	No	Yes	No
0000_1111	OEM Secure World Closed	Yes	Yes	Yes	Yes
0001_1111	OEM Closed	Yes	Yes	Yes	Yes
1001_1111	OEM Locked	N/A	N/A	N/A	N/A
0011_1111	OEM Return	No	No	Yes	N/A
11xx_xxxx	Brick	N/A	N/A	N/A	N/A

Figure 5 illustrates the lifecycle state transitions supported by the devices.

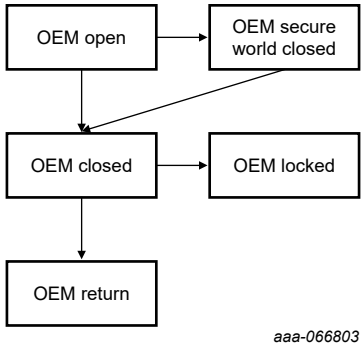


Figure 5. Lifecycle transitions

In addition to the lifecycles fuses, there are a handful of other fuses that will be used by the ROM. Table 5 highlights the differences in fuse configurations between the MCX W71 and MCX W72 devices.

Table 5. Differences in fuses between MCX W71 and MCX W72

Name	Fuse index		Description
	MCX W71	MCX W72	
SECURE_PHANTOM_CONFIG	-	0x18	Token configuration.
OEM_Enablement_Token	-	0x21	If a valid token is used by programming the SECURE_PHANTOM_CONFIG fuse, the 256-bit RoTKTH (root key hash) used for radio firmware authentication.
GD_EN	0x30	-	0b - the core VDD glitch detection does not generate a hardware reset. 1b - the boot ROM configures/enables the core VDD glitch detection based a reset.
HVD_EN	0x2F	-	0b - the core VDD high-voltage event does not generate a hardware reset. 1b - the boot ROM configures/enables the core VDD high-voltage detection based on a reset.
NBU_SEC_BOOT_EN	-	0x33	NBU secure boot enable fuse.
OEM_SEC_BOOT_EN	-	0x35	Main core secure boot enable fuse.

Table 5 is not a full description of the MCX W71 and MCX W72 fuses. For the full fuse list, refer to the 'Lifecycle and fuses' section in the MCX W71 and MCX W72 reference manuals.

3.2.3 ROM bootloader configuration

Sector 0 of the user IFR is the ROM and ISP configurations. These fields provide the configuration for the ROM bootloader, including the boot configuration with an offset value of 0x0050. Both devices support alternate boot speeds through the ROM bootloader feature.

MCX W72 implements the boot configuration to modify the DC-DC power for a low-power boot mode to save energy consumption:

- Bit 0 - Enable ISP:
  - 0: BOOT_CFG pin is disabled
  - 1: BOOT_CFG pin is enabled (default)
- Bits [2:1] - boot speed:
  - 00: Normal boot (32 MHz for MCX W71 and **48 MHz for MCX W72**)
  - 10: Fast boot (96 MHz)
  - 01, 11: Normal boot (default)
- Bit 3 - DC-DC boot power mode:
  - **0: Low-power boot mode**
  - 1: Normal-power mode

The bit 0 of the boot configuration field enables the ISP path for the ROM bootloader. If this bit is set, the BOOT_CFG pin is enabled. The pull-down on the BOOT_CFG pin ensures that the device does not enter the ISP by default. When the BOOT_CFG pin is enabled and not pulled down, the ROM bootloader enters the ISP path.

To determine how to configure the power and clock, the ROM bootloader loads the boot speed. The bit 1:2 of the boot configuration indicates the boot speed.

An additional DC-DC feature is available on MCX W72, as highlighted in bold in Table 6.

Table 6. Boot speed

Boot speed value in IFR0	Clock source	CPU_CLK/BUS_CLK/SLOW_CLK	DC-DC and CORE_LDO voltage
11/00/01: normal boot (default)	FRO_192M	48/48/24 MHz	CORE_LDO: 1.0 V **DC-DC: 1.8 V or 1.25 V**
10: fast boot	FRO_192M	96/96/24 MHz	CORE_LDO: 1.1 V **DC-DC: 1.8 V or 1.35 V**

MCX W72 offers new configuration fields for ISP modes with an offset of 0x140.

Go ISP mode options

When entering the ISP mode, if this bit and the 'go ISP mode fail alert pin' are enabled, the log is recorded on the corresponding pin:

- Bit 0: Reserved
- Bit 1: Secure boot option
  - 0: Enable
  - 1: 1 disable (default)
- Bit 7:2 – reserved

Go ISP mode fail alert pin

Set the ISP log pin:

- Bit [4:0] - Pin selection
  - 0x0 - 0x1F: Pin0~Pin31
- Bit [7:5] - Port selection
  - 0x0 - 0x4: PortA to PortE
  - 0x5 - 7: Reserved

For more details, refer to the *'Table 66. ROM configuration fields'* and *'Table 67. Boot speed'* in the MCX W72 Reference Manual.

3.2.4 ISP assignment

The ROM bootloader provides an in-system programming utility that operates over a serial connection on the MCUs. It enables quick and easy programming of MCUs through the entire product life cycle, including application development, final product manufacturing, and beyond.

In MCX W71 and MCX W72, the ROM bootloader supports automated booting from internal flash and downloading image from serial interface (LPUART, LPI2C, LPSPI, and CAN). The UART interface is up to 4 Mbit/s.

3.2.4.1 ISP path

To make the ROM bootloader enter the ISP path, press the BOOT_CONFIG pin PTA4. The BOOT_CONFIG pin is enabled by default in the Information Flash Region (IFR0).

For the FRDM-MCXW72 board, the users can place the ROM bootloader into ISP mode by pressing the SW3 button.

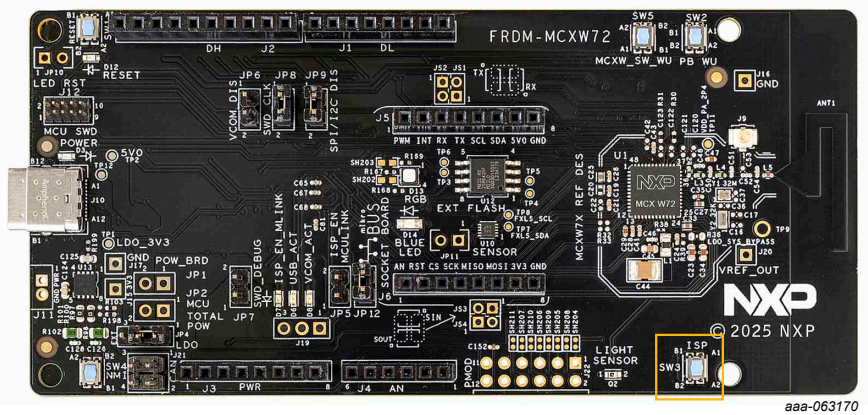


Figure 6. ISP assignment

Table 7 shows the peripheral instances and pin assignments that the ISP uses.

Table 7. Peripheral instances and pin assignments used by ISP

Peripheral instances	Pin name	Pin assignment	ALT
LPUART1	LPUART1_TX	PTC3	3
	LPUART1_RX	PTC2	3
LPSPI1	LPSPI1_PCS0	PTB0	2
	LPSPI1_SIN	PTB1	2
	LPSPI1_SCK	PTB2	2
	LPSPI1_SOUT	PTB3	2
CAN0	CAN0_TX	PTC4	3
	CAN0_RX	PTC5	3
LPI2C1	LPI2C1_SCL	PTB5	4
	LPI2C1_SDA	PTB4	4
Boot pin	BOOT_CONFIG	PTA4	Default

For more details on configuring the BOOT_CONFIG pin, refer to the Section 3.2.3.

The ROM bootloader enters the ISP path when no boot image is found, PC and SP are not valid, and some other boot failure conditions exist.

For more details on the ISP path, commands, and usage, refer to the '15.2.7. ISP path' section in the *MCX W72 Reference Manual* (MCXW72RM).

4 Hardware considerations (development boards)

This section summarizes the main differences and considerations of using NXP development boards with MCX W71 and MCX W72.

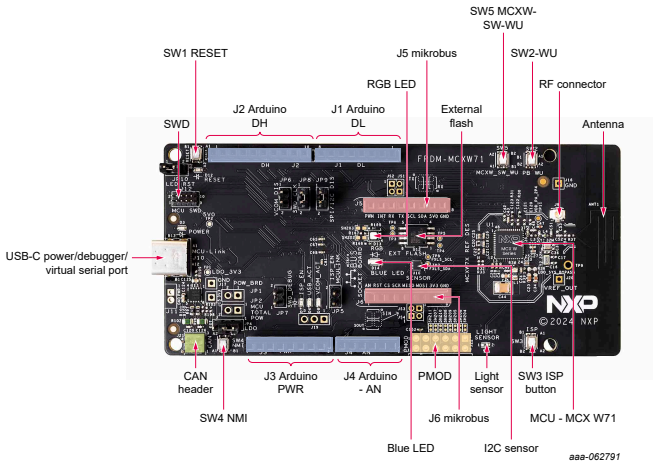


Figure 7. MCX W71 connectors, push buttons, and LEDs

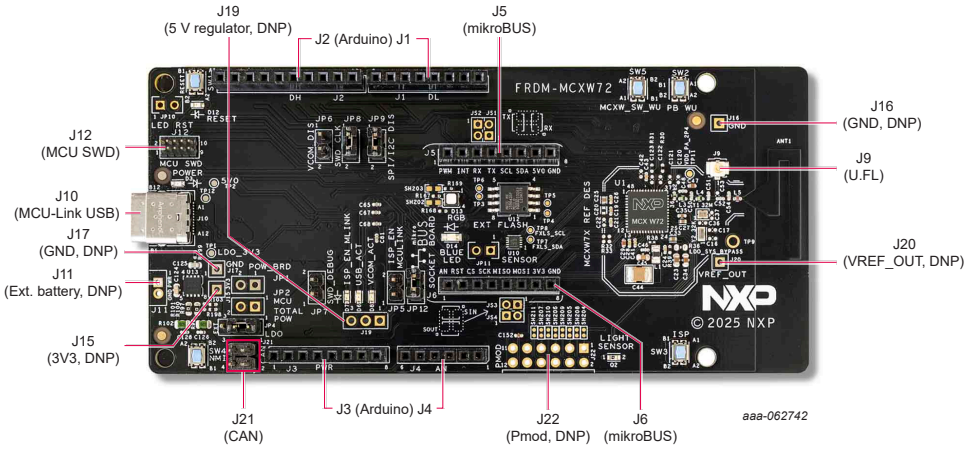


Figure 8. FRDM-MCXW72 board connectors

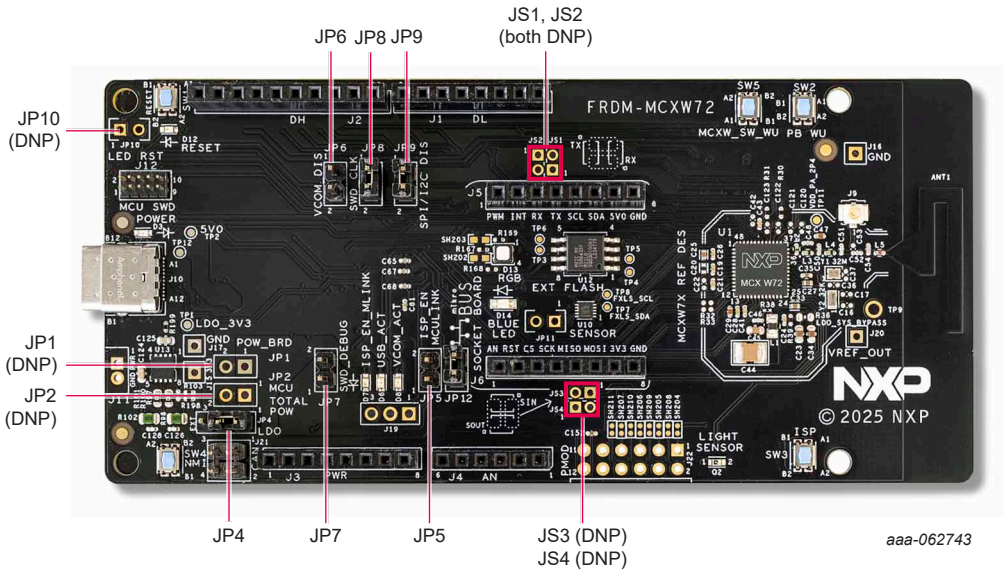


Figure 9. FRDM-MCXW72 board jumpers

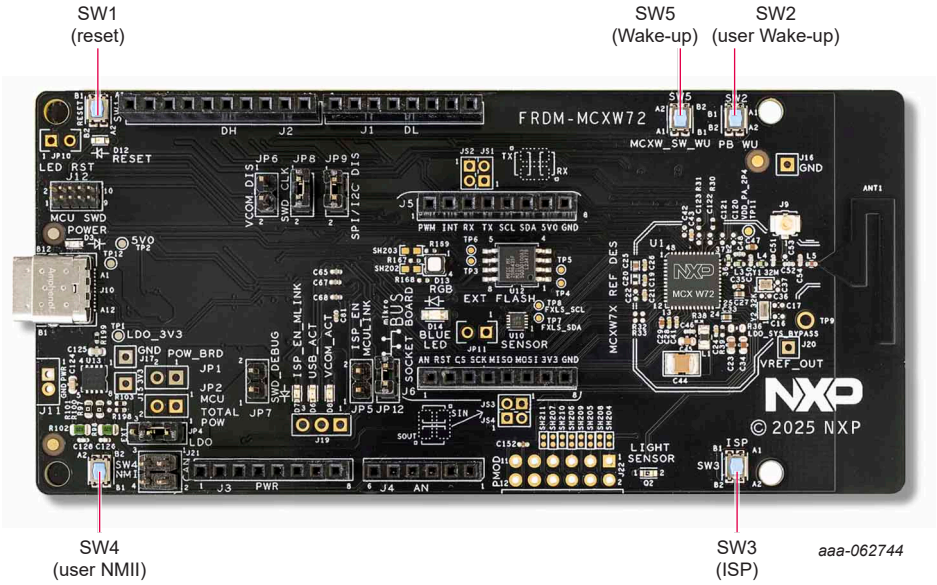


Figure 10. FRDM-MCXW72 board pushbuttons

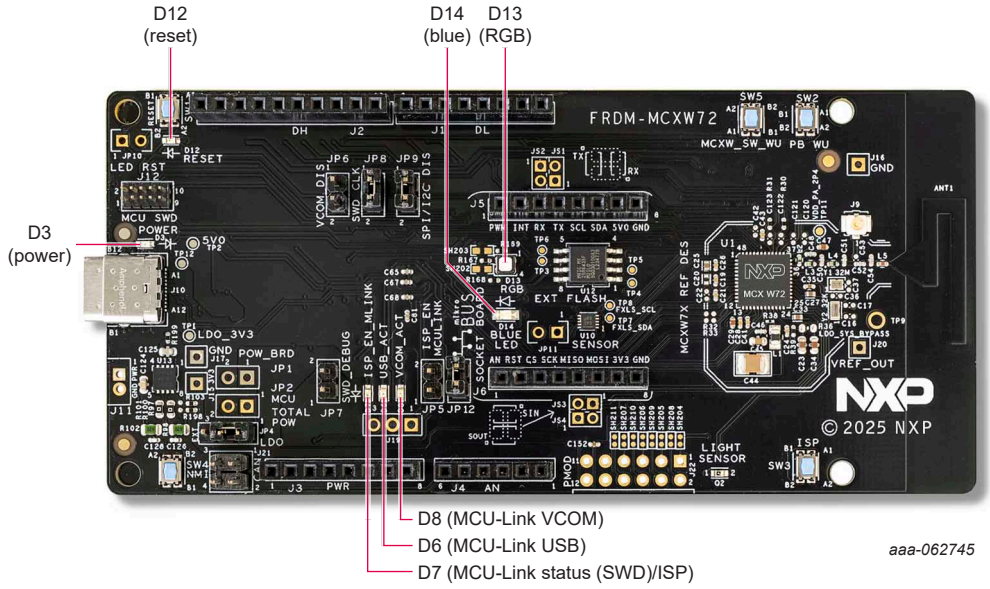


Figure 11. FRDM-MCXW72 board LEDs

4.1 Layout guidelines

To have a successful wireless hardware development, the proper device footprint, RF layout, circuit matching, antenna design, and RF measurement capability are essential. The RF circuit design, layout, and antenna design are specialties that require investment in tools and experience.

For FRDM-MCXW71, the design and board layout considerations are described in the *MCXW71 Hardware Design Guide* (UG10146) and The best way to build a PCB first time right with KW45 (Automotive) or K32W1/MCXW71 (IoT/Industrial).

For FRDM-MCXW72, the design and board layout considerations are described in the MCX W72 Hardware Design Guide (UG10273) and the best way to build a PCB first time right with KW47 (Automotive) or MCX W72 (IoT/Industrial).

4.2 Power-supply configurations

This section outlines the supported power supply configurations for MCX W71 and MCX W72 devices, including power modes, input supply sources, and distribution of secondary supplies to on-board components.

4.2.1 Programming interfaces

To use the MCU-Link for debugging MCX W71 and MCX W72 MCUs, disable the MCU-Link SWD on the board (short jumper JP7, and do not connect it to the ground). Connect the target MCU SWD interface to the MCU-Link. Do not use the target MCU external debugger connector J12 for an external connection.

To use an external debugger to debug MCX W71 and MCX W72 MCUs, short jumper JP7 and connect the external debugger to J12. The target MCU SWD interface is connected to the external debugger.

4.2.2 Power supply modes

Both MCX W71 and MCX W72 support the following power supply configurations:

- DC-DC Buck
- DCDC Bypass (LDO Mode)
- Smart Power Switch

Note: *PMIC supply mode is supported only on MCX W71.*

The FRDM-MCXW71 and FRDM-MCXW72 boards can be powered using one of the following primary power supply options:

- An external 5 V supply through MCU-Link USB-Type C connector J10
- An external 5 V to 9 V supply through Arduino socket connector J3 (pin 8)

The primary power supply produces secondary power supplies on the board. The secondary power supplies provide power to board components, including the MCX W7x MCU, MCU-Link, QSPI flash memory, CAN transceiver, accelerometer, Arduino socket, mikroBUS socket, and Pmod connector.

For more details on different Power modes, refer to the ‘2.1 Power supplies’ section in the *FRDM-MCXW71 Board User Manual* (UM12063) and *FRDM-MCXW72 User Manual* (UM12222).

4.2.3 Current measurement

The FRDM-MCXW71 and FRDM-MCXW72 boards support current measurement using an ammeter.

The power supplies for which the current measurement can be done and the related steps are listed in the table *‘Power supplies with current measurement support’* in the FRDM-MCXW71 Board User Manual (UM12063) and the FRDM-MCXW72 User Manual (UM12222).

5 Software considerations

This section outlines the key software considerations for working with MCX W71 and MCX W72 devices. It covers low-power specifications, SDK download and setup (including integration with VS Code), and programming the NBU firmware for wireless examples on both devices

5.1 Low-Power specifications

Both MCX W71 and MCX W72 devices support Active, Sleep, Deep-Sleep, Power-Down, and Deep-Power-Down modes. Any reset brings the chip back to the Active mode.

There is also another power mode called 'VBAT ON', which can be referred to as a ‘Smart Power Switch’.

For more information, refer to the *'Chapter 31 Smart Power Switch (VBAT)'* of *MCX W72 Reference Manual* (MCXW72RM) and *Features, Usage, and Capabilities of Smart Power Switch on the MCX W72 Microcontroller* (AN14745).

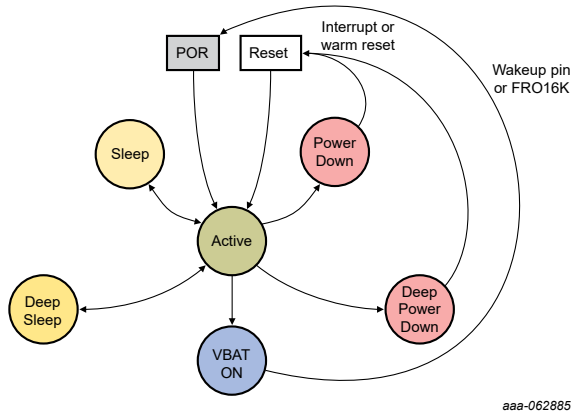


Figure 12. MCX W71 and MCX W72 Power modes

For more information, refer to the *'Power Modes'* section of MCX W71 Reference Manual and MCX W72 Reference Manual.

5.2 Software Development Kit (SDK) download and install

This section shows how to download the SDK for the MCX W71 and MCX W72 devices. The steps to download the SDK package for the MCX W71 devices are listed in the following subsections.

5.2.1 Importing MCUXpresso SDK repository to VS Code

One method to import the NXP SDK into **MCUXpresso for VS Code** is by importing a remote repository. Follow the steps below:

1. In MCUXpresso for VS Code, click **Import Repository** from the *Quick Start Panel*.

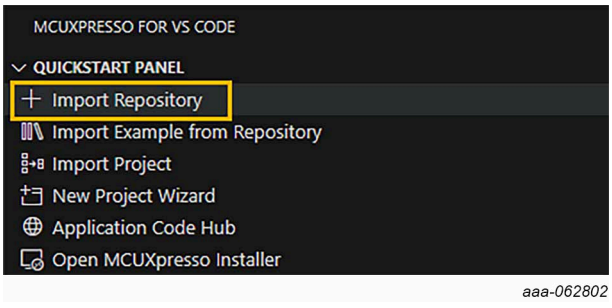


Figure 13. Import repository option in quick start panel

2. Click the **Remote** tab and select the **MCUXpresso SDK Repository**. By default, the **main** revision is selected. If a different revision is required, expand the list and choose accordingly. After making the selection, click **Import**.

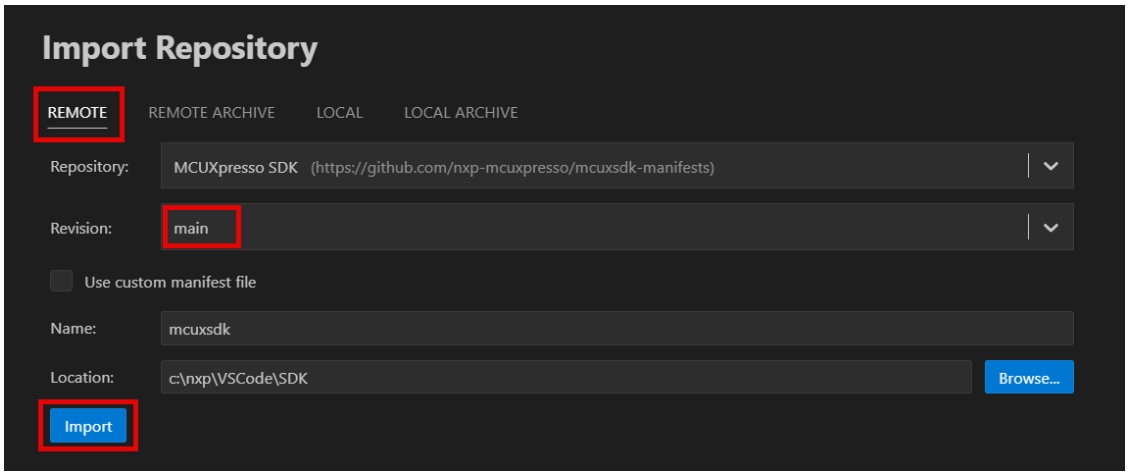


Figure 14. Selecting remote repository and importing SDK

3. After the import process is complete, the **MCUXpresso SDK** repository appear in the **Imported Repositories** window.

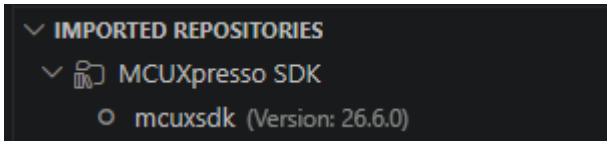


Figure 15. Imported MCUXpresso SDK repository

5.3 Program NBU firmware for wireless examples

MCX W71 and MCX W72 devices run the Radio Link Layer (LL) software from the dedicated Narrow Band Unit (NBU). The NBU is based on a Cortex-M3 in MCX W71, and a Cortex-M33 in MCX W72.

For MCX W71, the NBU firmware is provided in the SDK folder as a Secure Binary (SB3) file, preprogrammed with NXP default keys and a XIP file to generate a secure binary firmware based on custom keys. For MCX W72, the NBU firmware is provided as a standard binary file in the SDK folder.

The primary difference between the NBU on MCX W71 and MCX W72 lies in how each device manages access and security during the boot process:

- MCX W71 restricts access to NBU flash through the boot ROM. As a result, the device requires an SB3 file for authentication and secure boot. This SB3 file is signed using the OEM authentication keys already provisioned on the device, along with the OEM symmetric key for encryption.
- In contrast, MCX W72 supports an open NBU, which provides direct access to the NBU. In this case, during development, an SB3 file is not required when communicating with the boot ROM. Standard binary files (.bin) provided in the SDK can be used to update the NBU firmware using ISP commands, provided the MCU remains in the development lifecycle state. If the lifecycle is advanced to production, an SB3 file is required for updates, similar to MCX W71.

The NBU binary file is located in the SDK folder. Go to the SDK root folder and open the path: `\main\mcuxsdk\mcuxsdk\middleware\wireless\ble_controller\bin`

For MCX W71, SB3 and XIP files are included for binary files: `mcxw71_nbu_ble_hosted`

For MCX W72, the following binary file is available: `mcxw72_nbu_ble_hosted`

Note: *Ensure that the NBU firmware version matches the SDK version of the application. After downloading the SDK and before running any wireless examples, update the NBU firmware using the binaries provided in the SDK folder.*

5.3.1 Load NBU firmware in MCX W71

The NBU firmware is distributed within the SDK as an *.sb3* file or as a binary file (*.xip*). There are two options to update the NBU firmware.

For the FRDM-MCXW71 board, use a prepared SB file with NBU firmware. The NBU is programmed with a signed image with provisioned NXP keys. NXP provides the default keys in the fuse (SB3KDK and RoTKTH). The *.sb3* file related to these keys is included in each SDK.

The other way is to prepare a custom SB3 file that loads the NBU firmware. The samples not programmed with any keys, for which you can create custom keys, are included in the .xip file.

For the MCX W71 chip received from factory, by default the SBKDK and RoTKTH keys in the fuse are null. Therefore, writing these two keys to fuse is essential.

The following subsections guide you through programming the NBU software for the FRDM-MCXW71 board.

5.3.1.1 Bootloader Host Application (blhost)

The NBU firmware can be uploaded or updated using the NXP Bootloader Host Application (blhost) by setting the board into ISP mode. To program the NBU software for the FRDM-MCXW71 board, follow the steps below:

1. Open a command prompt window on your system.
2. Change the directory to the location of the blhost.exe file: `(BLHost_root_location)\blhost_2.6.7\bin\win`.
3. Place the device in ISP mode. For this example, the UART peripheral is used by connecting a USB cable to J10. On the FRDM board, the ISP mode can be entered by using the following method:
   a. Connect FRDM-MCXW71 board from the MCU-Link J10 connector to the PC using a USB-Type C cable.
   b. While connecting, press and hold *SW3*.
4. Release SW3.
5. To check the MCU-Link COM, open the *Device manager*. In this example, the assigned COM port is 'COM31'.

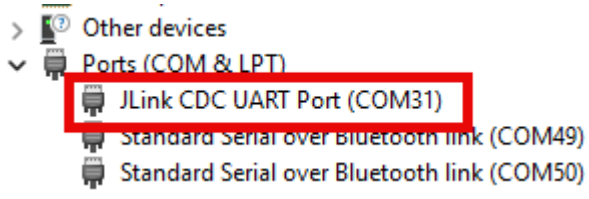


Figure 16. Device manager showing J-Link CDC UART Port

6. In the command prompt, run the following command to verify if communication is working:

```
blhost -p COMX get-property 1
```

Note: *Do not forget to replace with your COM device.*

7. Update the NBU firmware using the following command. Ensure the full path to the binary is provided, or place the binary file in the same folder as blhost.exe:

```
blhost -p COMX,57600 receive-sb-file mcxw71_nbu_ble_hosted.sb3
```

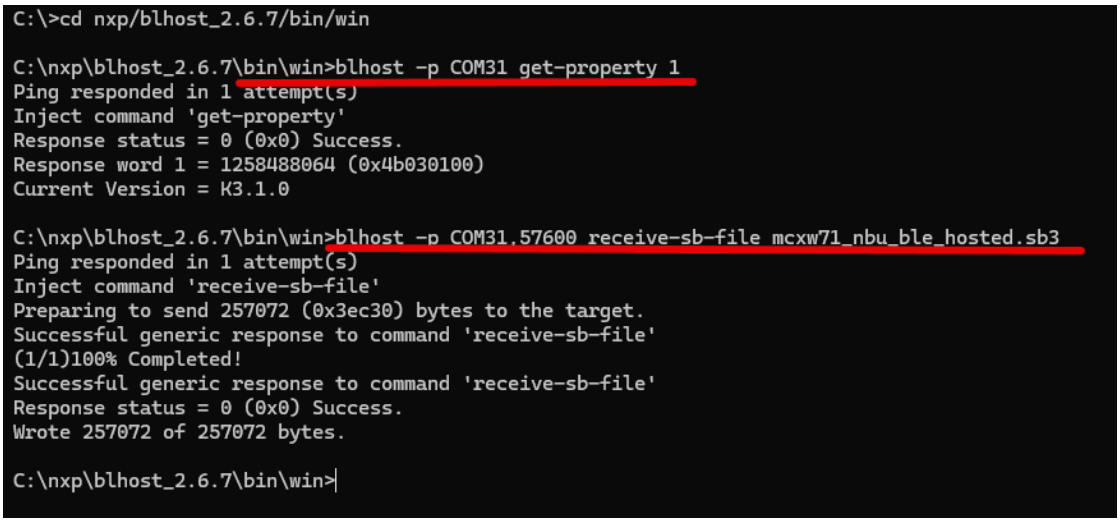


Figure 17. Example command execution and successful response in command prompt

5.3.1.2 Secure Provisioning Tool (SEC Tool)

The NBU firmware can also be uploaded or changed using the NXP Secure Provisioning Tool (SEC Tool). The GUI-based application can be downloaded from: MCUXpresso Secure Provisioning Tool.

To upload the SB3 file with NXP keys on the FRDM-MCXW71 using SEC Tool, follow the steps below:

1. Create or open a workspace for MCX W71 devices.

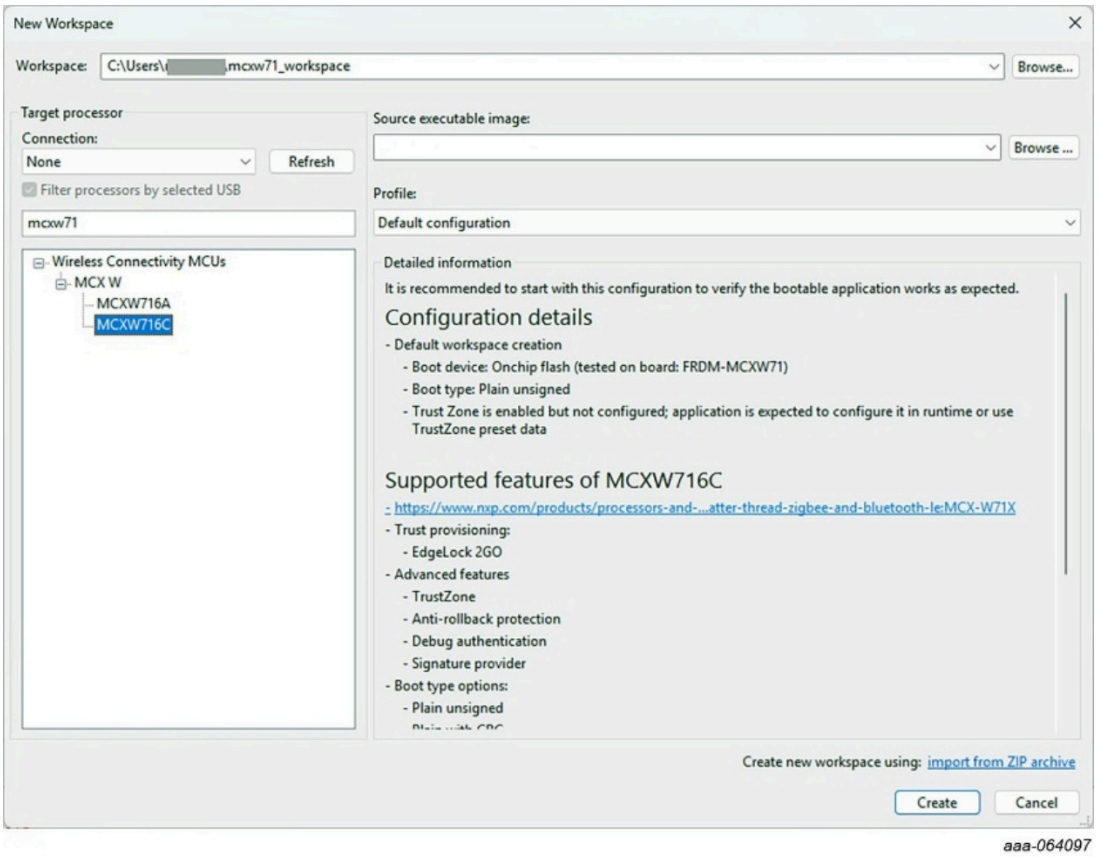


Figure 18. SEC Tool workspace configuration for MCX W71 device

2. In the menu bar, select Tools → Manufacturing Tool.
3. Select the **Apply SB image** operation
4. Provide the SB file. When using the FRDM-MCXW71 board, the SB file is available at the following path folder of the SDK: /`path_to_SDK/middleware/wireless/ble-controller/bin`.
5. Detect the connected device by clicking **Auto detect**, and then click **Test connection** to verify communication.

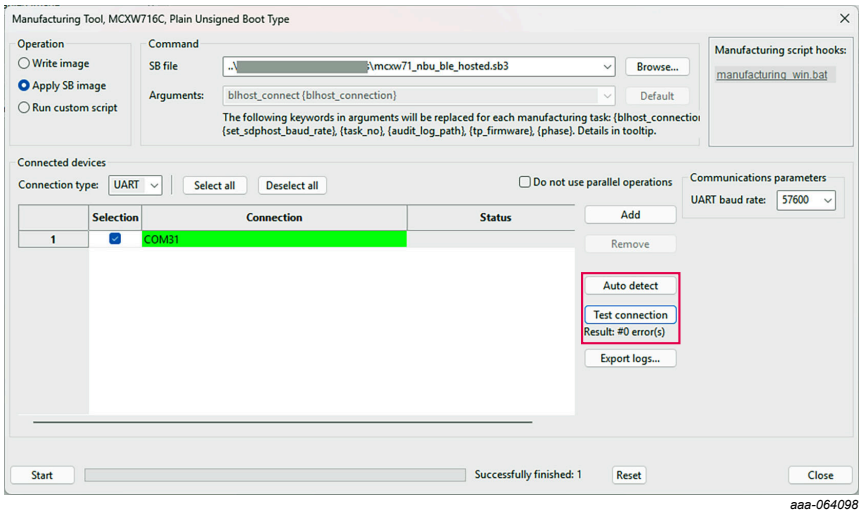
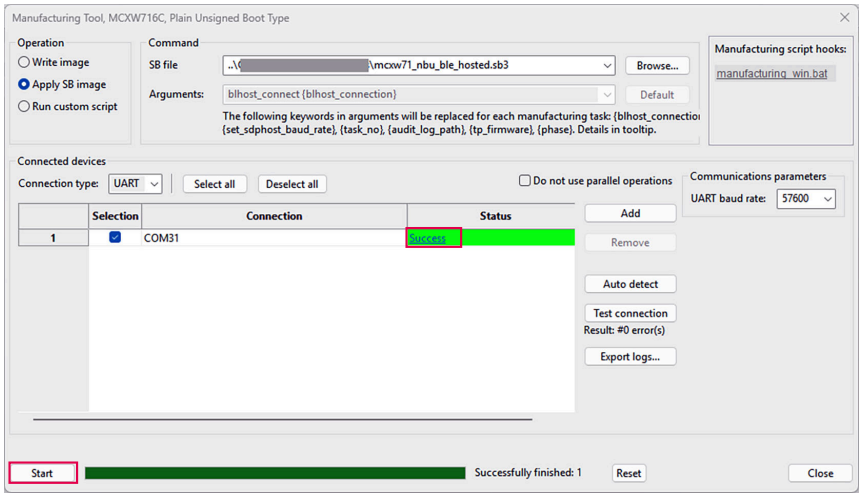


Figure 19. Device detection and connection test in SEC manufacturing tool

6. Load the SB file by clicking the **Start** button. A completion message confirms the successful upload of the SB file.



aaa-064099

Figure 20. Successful SB File upload in SEC Tool

5.3.2 Load NBU firmware into MCX W72

The NBU firmware is included in the SDK folder as binary firmware at the following path:

```
path_to_SDK\mcuxsdk\middleware\wireless\ble_controller\bin
\mcxw72_nbu_ble_hosted.bin
```

5.3.2.1 Bootloader Host Application (blhost)

To program the NBU software for the FRDM-MCXW72 board, follow the steps below:

1. Place the FRDM-MCXW72 board in ISP mode by pressing and holding *SW3*. Then connect the USB cable to the *J10* connector and release **SW3** after the connection is established.
2. Verify the COM port assigned to the FRDM-MCXW72 board by opening **Device Manager** in Windows and searching for **Ports (COM & LPT)**. Then identify and note the corresponding COM port number (for example, **MCU-Link VCOM Port (COM16)**).

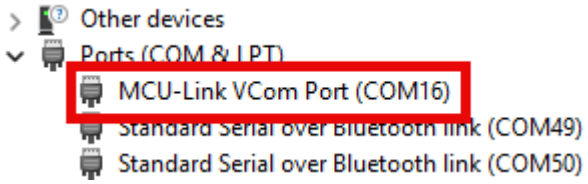


Figure 21. MCU-Link virtual COM Port in Device manager

3. Open a command prompt and change the directory to the location of the blhost.exe file:

```
(BLHost_root_location)\blhost_2.6.7\bin\win
```

4. Then verify communication by running the following command, making sure to replace `COMX` with the COM port assigned to your device:

```
blhost.exe -p COMX get-property 1
```

5. Update the NBU firmware image. First, erase the NBU memory:

```
blhost.exe -p COMX flash-erase-all 2
```

6. Write the new firmware image using the following command. The write-memory command uses 0x48800000 as the start address, which corresponds to the NBU memory base. Provide the full path to the binary file, or alternatively, copy the .bin file into the same directory as blhost.exe to simplify the command:

```
blhost.exe -p COMX write-memory 0x48800000 mcxw72_nbu_ble_all_hosted.bin
```

```
C:\nxp\blhost_2.6.7\bin\win>blhost.exe -p COM16 get-property 1
Ping responded in 1 attempt(s)
Inject command 'get-property'
Response status = 0 (0x0) Success.
Response word 1 = 1258488064 (0x4b030100)
Current Version = K3.1.0

C:\nxp\blhost_2.6.7\bin\win>blhost.exe -p COM16 flash-erase-all 2
Ping responded in 1 attempt(s)
Inject command 'flash-erase-all'
Successful generic response to command 'flash-erase-all'
Response status = 0 (0x0) Success.

C:\nxp\blhost_2.6.7\bin\win>blhost.exe -p COM16 write-memory 0x48800000 mcxw72_nbu_ble_all_hosted.bin
Ping responded in 1 attempt(s)
Inject command 'write-memory'
Preparing to send 268609 (0x41941) bytes to the target.
Successful generic response to command 'write-memory'
(1/1)100% Completed!
Successful generic response to command 'write-memory'
Response status = 0 (0x0) Success.
Wrote 268609 of 268609 bytes.

C:\nxp\blhost_2.6.7\bin\win>
```

Figure 22. Successful command execution for firmware programming using blhost

5.3.2.2 Secure provisioning Tool (SEC Tool)

The NBU firmware can be uploaded or updates using the NXP Secure Provisioning Tool (SEC Tool). The GUI-based application can be downloaded from: MCUXpresso Secure Provisioning Tool.

1. Create workspace for MCX W727C device.

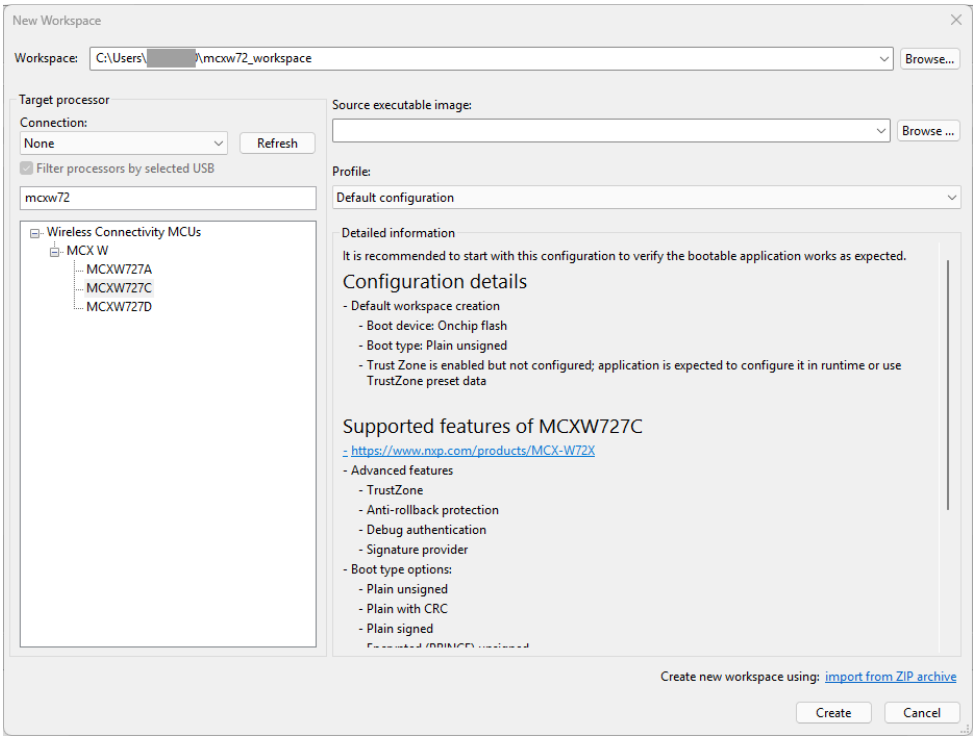


Figure 23. SEC Tool workspace configuration for MCX W72C

2. Place the device in ISP mode by pressing and holding SW3, connecting the USB cable to the J10 connector, and then releasing SW3 after the connection is established.
   To verify communication:
   a. Click the **UART** tab.
   b. Enter the assigned COM port in the **Port** field under the **UART** configuration section.
   c. Click **Test connection** and confirm that the result displays **OK**.

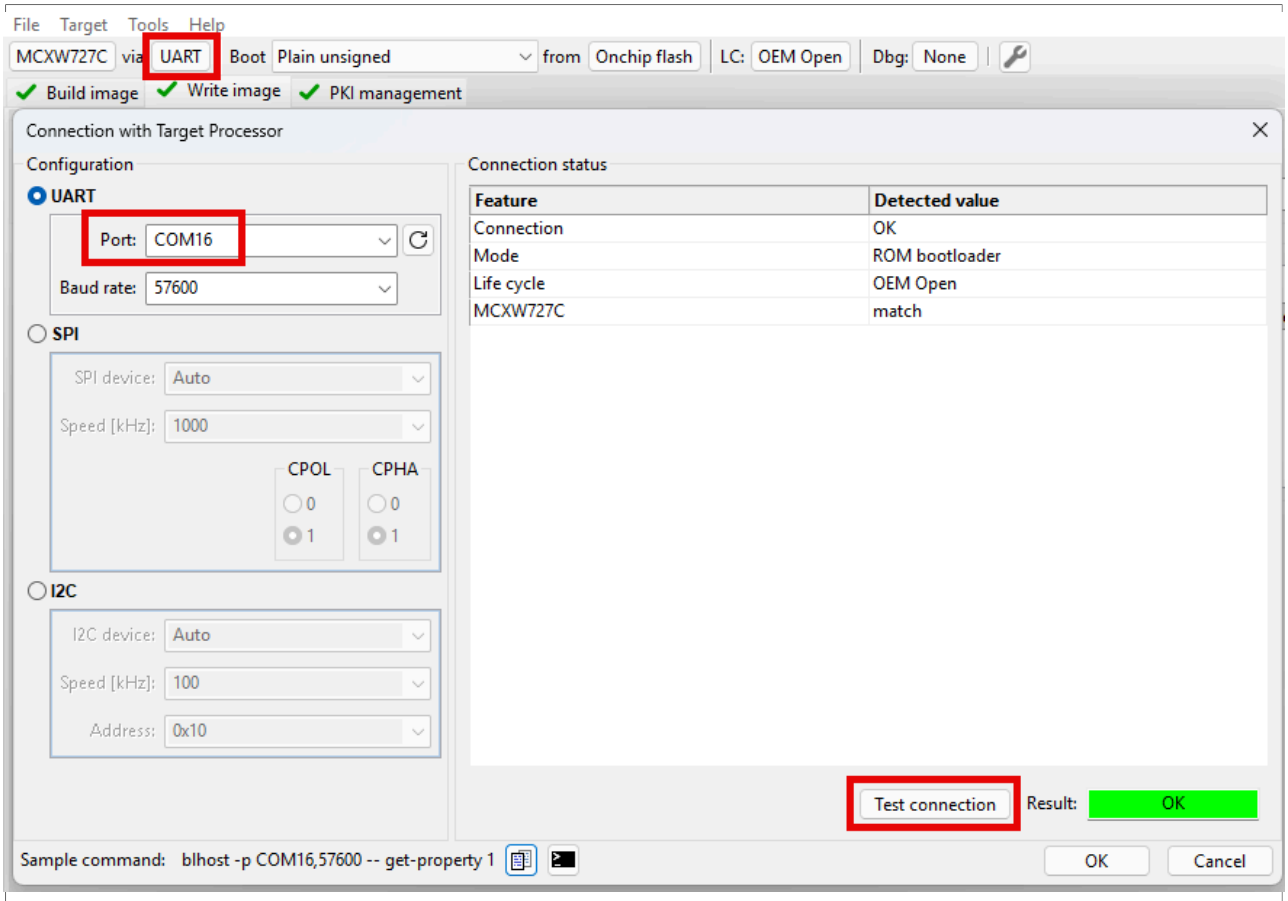


Figure 24. UART configuration and successful connection in SEC Tool

3. In the toolbar, select the boot type as **Plain unsigned** or **Plain with CRC**.
4. In the **Build Image** view, load the binary file in **Source executable image**, enter the start address as `0x48800000` (which corresponds to the NBU memory base), and click **Build image**.

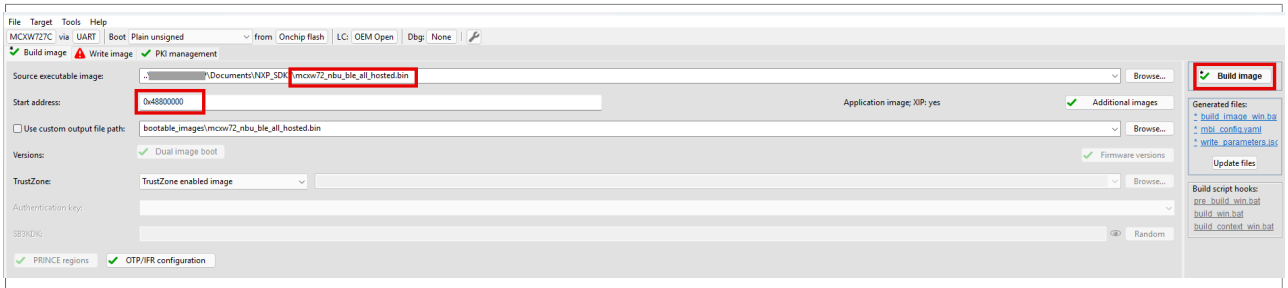


Figure 25. Build image configuration in SEC Tool

5. Navigate to the **Write Image** view, select **Use built image**, and click **Write image**.

Figure 26. Write image operation in SEC Tool

6. A success message appears when the NBU firmware loading is complete.

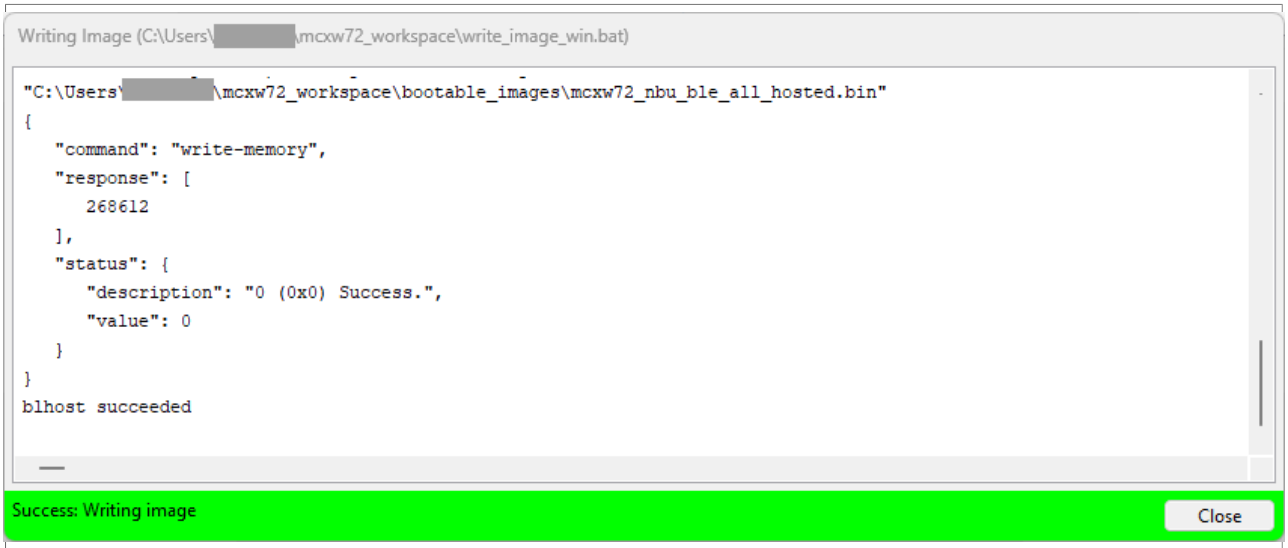


Figure 27. Successful firmware upload confirmation in SEC Tool

5.3.2.3 LinkFlash tool

A third way of updating the NBU for FRDM-MCXW72 board is using the LinkFlash Tool from LinkServer. For this method, a LinkServer debug probe must be available on the board.

To install LinkServer, download it from the NXP webpage LinkServer for Microcontrollers and select the package corresponding to your operating system. Alternatively, install it using the MCUXpresso installer from the VS Code extension by selecting the Linkserver component.

1. Navigate to the LinkServer folder and open the path `MCU-LINK_installer\scripts`, which contains two scripts, one for programming the CMSIS debug probe and another for the J-Link debug probe. For LinkServer, use the CMSIS script.

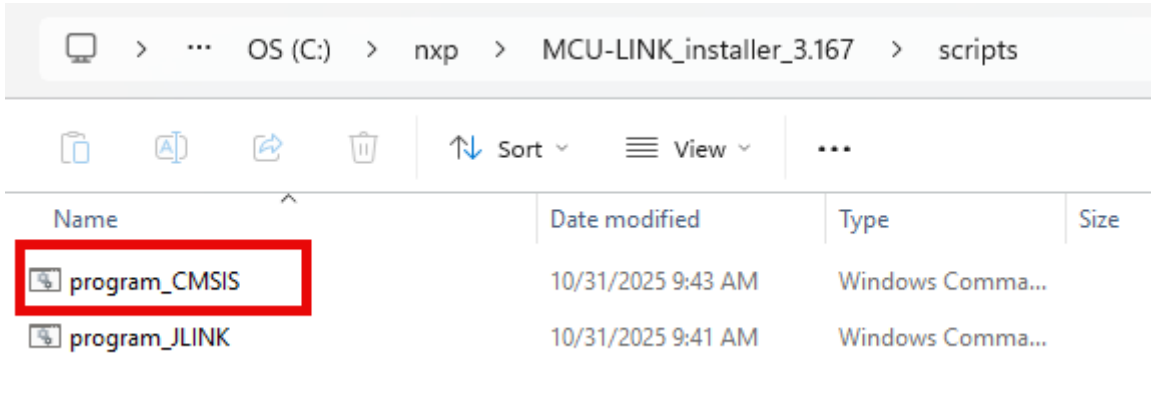


Figure 28. CMSIS programming script in MCU-link scripts folder

2. Open the **program_CMSIS** script and place the board in ISP USB mode by placing a jumper on JP5.
3. Navigate back to the main LinkServer folder and execute the `LinkFlash.exe` file.
4. Ensure that the device is in **ISP mode** by pressing and holding **SW3**, connecting the USB cable to the **J10 connector**, and then releasing **SW3** after the connection is established.
5. After the device is connected, click the **Refresh** button to update and set the probe target of the board. Ensure that the JP5 jumper is removed before performing this step.

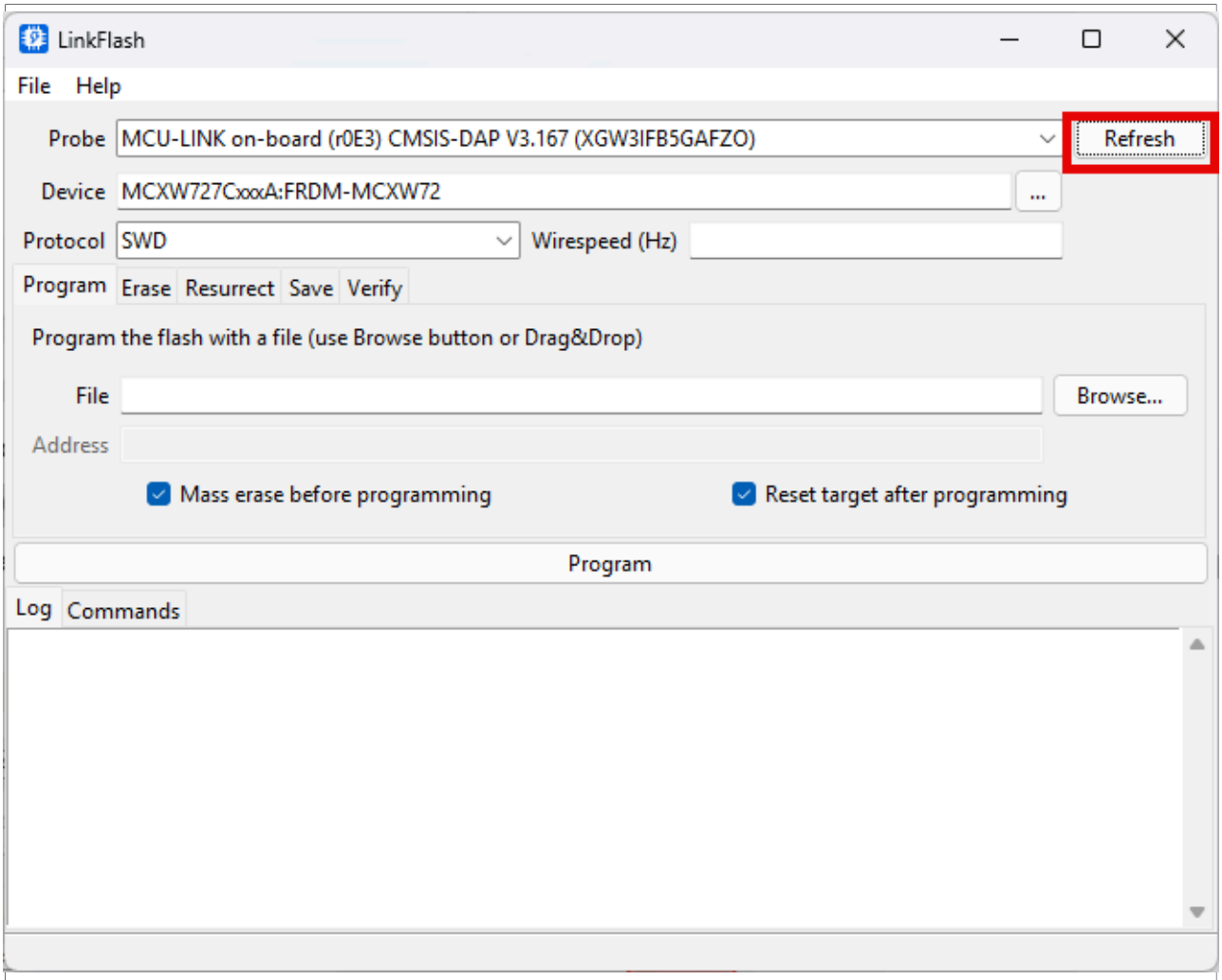


Figure 29. Device detection and refresh in LinkFlash tool

6. Go to the **Program** tab, click **Browse**, select the NBU firmware file, and enter the start address `0x48800000`.
7. Ensure to check the boxes **Mass erase before programming** and **Reset target after programming**.

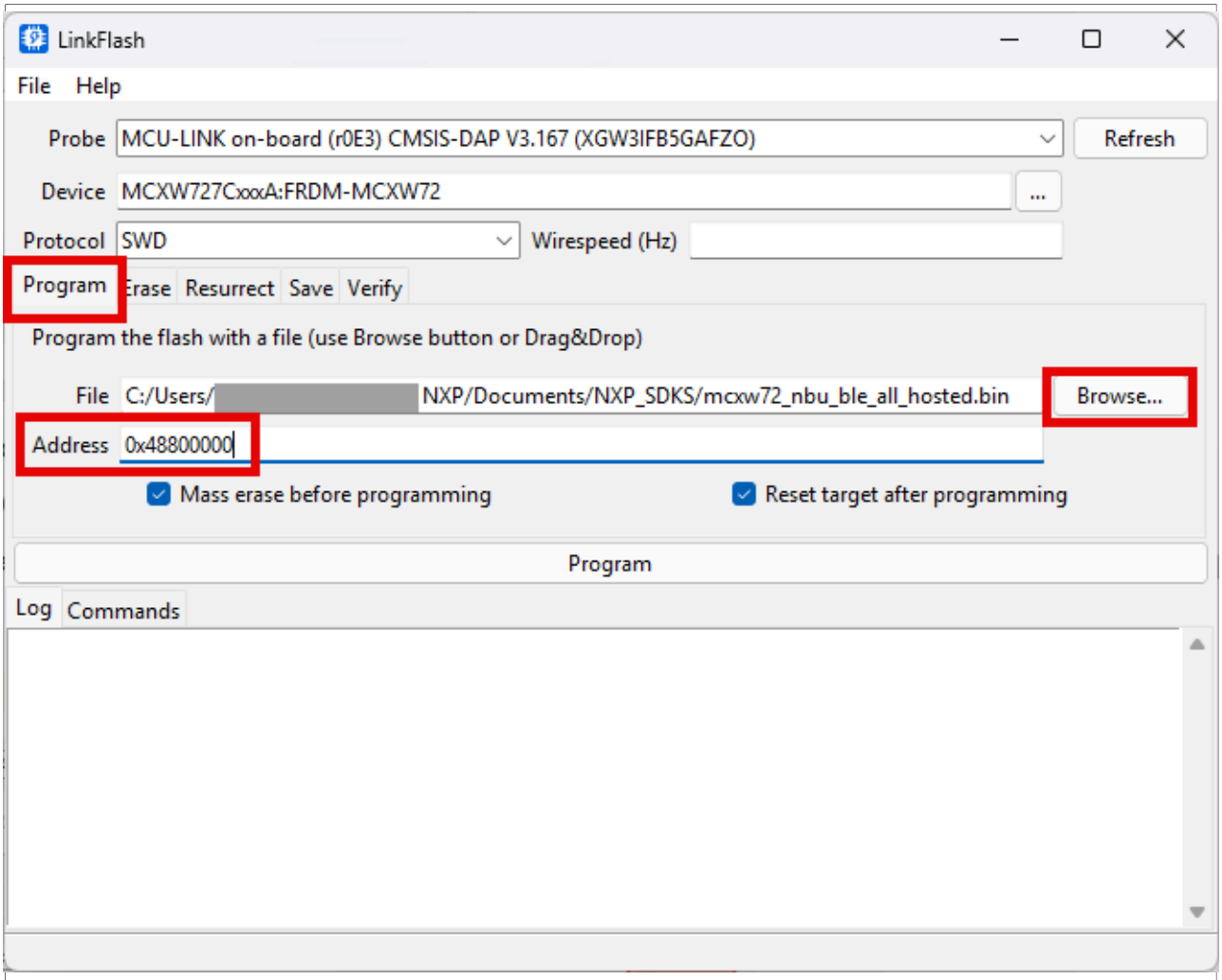


Figure 30. Firmware programming configuration in LinkFlash

8. To load the NBU firmware, click *Program*.

5.4 Bluetooth LE migration project

This section describes a method for migrating an existing Bluetooth LE application from MCX W71 to MCX W72. The example used as a reference is the *frdmmcxw71_beacon_freertos* application included in the SDK.

To migrate an application for MCX W72, it is recommended to use an example from the SDK as a base for the destined MCU to ensure proper configuration. The application level can vary depending on the specific use case, and SDK examples require application-level modifications implemented by the user to meet specific needs.

To migrate examples that involve a specific type of peripheral connection, other modifications can be required, including updates to declarations, instances, and functionality in the corresponding hardware files. Examples involving custom boards require corresponding adjustments to accommodate hardware modifications implemented by the end user.

Taking as a base the existing MCX W71 Bluetooth LE project Updating Beacon Data, it can be migrated to **MCX W72** by following the steps below:

1. From the *Quickstart Panel* window, click **Import Example from Repository**. Select the **MCUXpresso SDK** Repository, choose the **FRDM-MCXW72** board, and select the **`beacon_freertos`** example.
2. For the **App type**, select **Freestanding application**. Modifying SDK files within a freestanding (standalone) project is a cleaner and more maintainable approach. Click **Import**.

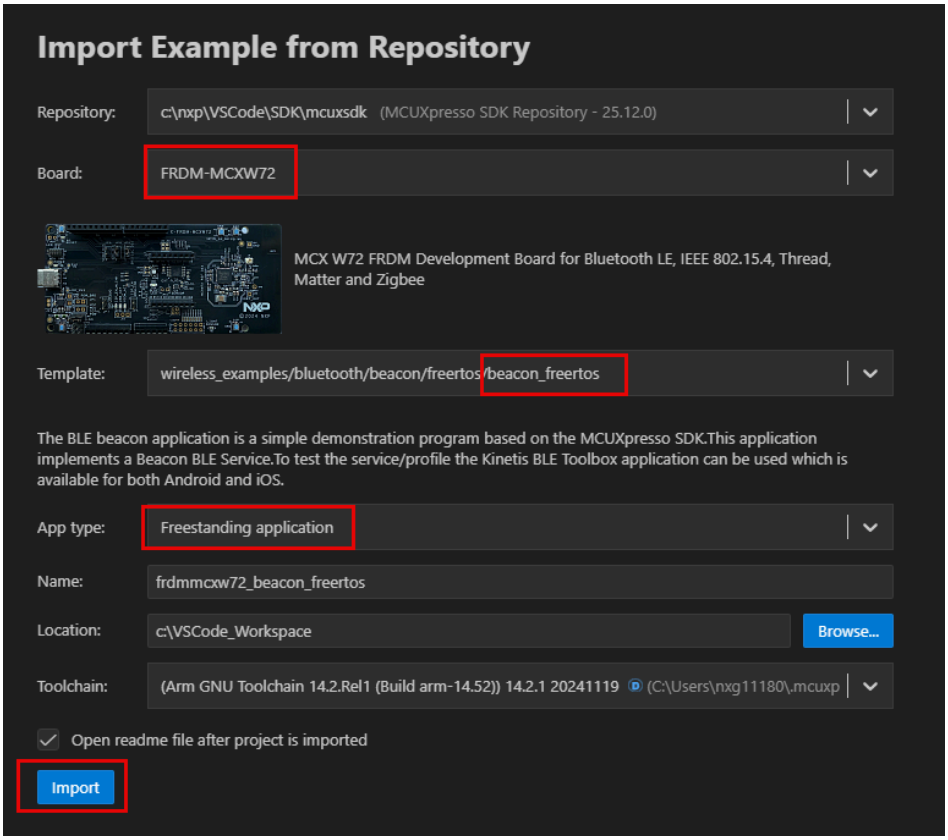


Figure 31. Importing beacon_freertos example from MCUXpresso SDK repository

3. From the preexisting FRDM-MCXW71 Beacon project, select the following files (for example, `beacon.c`, `app_preinclude.h`, `app_advertiser.c`, and `app_advertiser.h`):

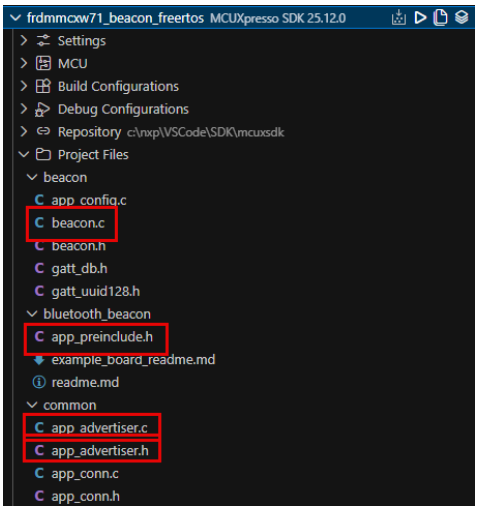


Figure 32. Source files selected from MCXW71 Beacon project

4. Drag and drop the selected files into the *Project files* section of the FRDM-MCXW72 project. When prompted, click **Copy Files** and select **Yes to All** to replace the existing files.

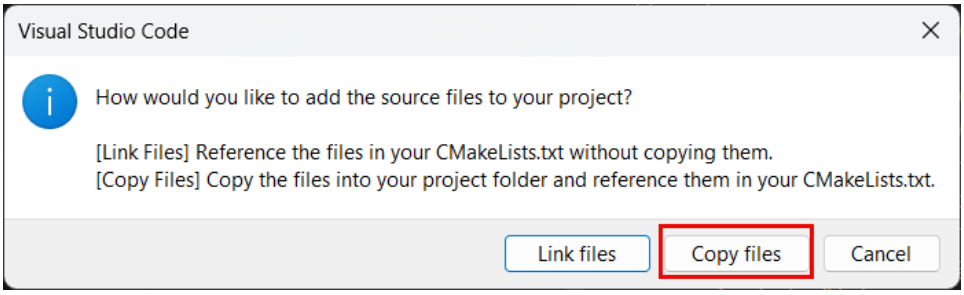


Figure 33. Confirmation prompt for replacing existing files in project

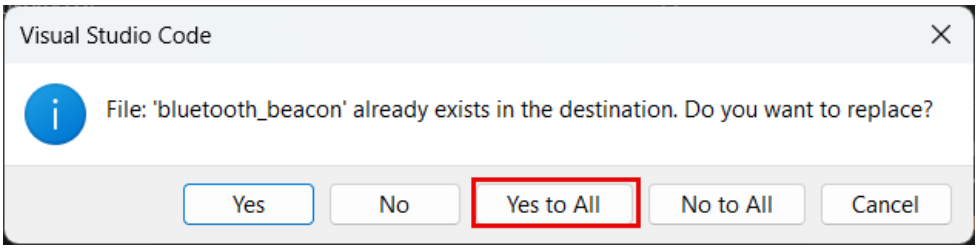


Figure 34. Replace existing file confirmation dialog

5. After integrating the files, right-click the **frdmmcxw72_beacon_freertos** project and select **Pristine Build/ Rebuild Project** to ensure that the application compiles successfully, and all dependencies are resolved. Then debug the application.

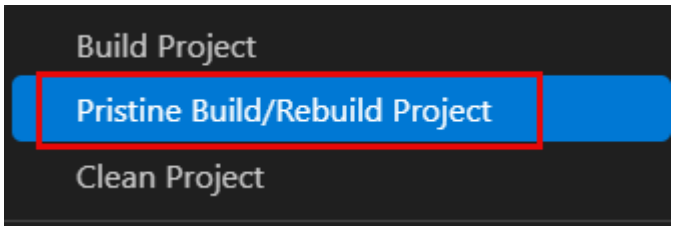


Figure 35. Pristine Build/Rebuild option in MCUXpresso

6. Open NXP **IoT Toolbox**, select **Beacons**, and click **Scan** to verify operation.

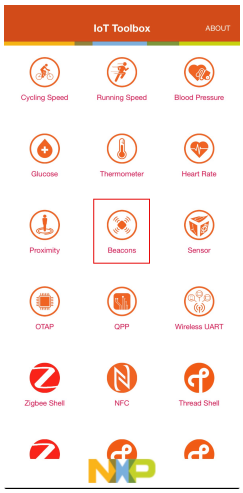


Figure 36. Beacon scan interface in NXP IoT toolbox

7. On the FRDM-MCXW72 board, press the SW2 button to start advertising.
8. In the **IoT Toolbox**, verify that the beacon data is updated. The counter value in the field **A** must increase at regular intervals (for example, every 5 seconds).

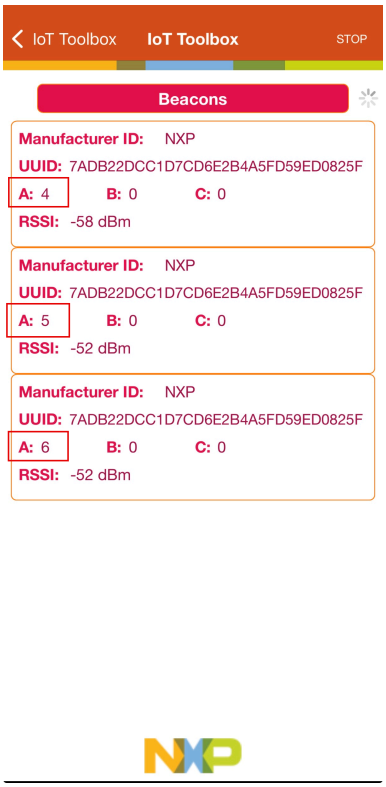


Figure 37. Beacon advertisement data observed in IoT toolbox

For detailed instructions on testing the application and validating its functionality, refer to the KW45 – Updating Beacon Data community post. This source provides step-by-step guidance on using the IoT Toolbox and interpreting beacon data.

The examples referenced in the Bluetooth LE demo applications user guide are compatible with FRDM-MCXW72 devices. Also, the examples can be migrated from FRDM-MCXW71 to FRDM-MCXW72 by applying project-specific modifications, as outlined in this section.

Note: *The required changes at the application level can vary depending on the specific use case.*

For reference information about Bluetooth LE examples and modifications, refer to the following guides:

- Bluetooth Low Energy Demo Applications User's Guide
- Bluetooth Low Energy Application Developer's Guide

6 Acronyms

Table 8 lists the acronyms used in this document.

Table 8. Acronyms

Term	Description
Bluetooth LE	Bluetooth Low Energy
CAN	Controller Area Network
FlexIO	Flexible Input/Output
I3C	Improved Inter-Integrated Circuit
IFR	Information Flash Region
ISP	In System Programming
LL	Link Layer
LPI2C	Low-Power Inter-Integrated Circuit
LPSPI	Low-Power Serial Peripheral Interface
LPUART	Low-Power Universal Asynchronous Receiver/Transmitter
NBU	Narrow-Band Unit
OEM	Original Equipment Manufacturer
RAM	Random Access Memory
RF	Radio Frequency
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter

7 References

Following is the list of references used to supplement this document:

- MCX W71 product page
- MCX W72 product page
- *MCX W71 Reference Manual* (MCXW71RM)
- *MCX W72 Reference Manual* (MCXW72RM)
- *FRDM-MCXW71 Board User Manual* (UM12063)
- *FRDM-MCXW72 User Manual* (UM12222)
- FRDM Development Board for MCX W71x Wireless MCUs
- FRDM Development Board for MCX W72x Wireless MCUs

- *MCXW71 Hardware Design Guide* (UG10146)
- *MCX W72 Hardware Design Guide* (UG10273)
- *Features, Usage, and Capabilities of Smart Power Switch on the MCX W72 Microcontroller* (AN14745)
- The best way to build a PCB first time right with KW45 (Automotive) or K32W1/MCXW71 (IoT/Industrial)
- The best way to build a PCB first time right with KW47 (Automotive) or MCX W72 (IoT/Industrial)
- NXP Bootloader Host Application (blhost)
- MCUXpresso Secure Provisioning Tool
- LinkServer for Microcontrollers
- Bluetooth Low Energy Demo Applications User's Guide
- Bluetooth Low Energy Application Developer's Guide
- Updating Beacon Data
- MCXW71 Knowledge Hub - NXP Community
- MCXW72 Knowledge Hub - NXP Community

8 Revision history

Table 9. Revision history

Document ID	Release date	Description
AN14802 v.2.0	17 July 2026	Intial public release
AN14802 v.1.0	15 September 2025	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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 B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents