UM12294

Installing and using CodeWarrior IDE for NPM8

Rev. 1.1 — 9 July 2025

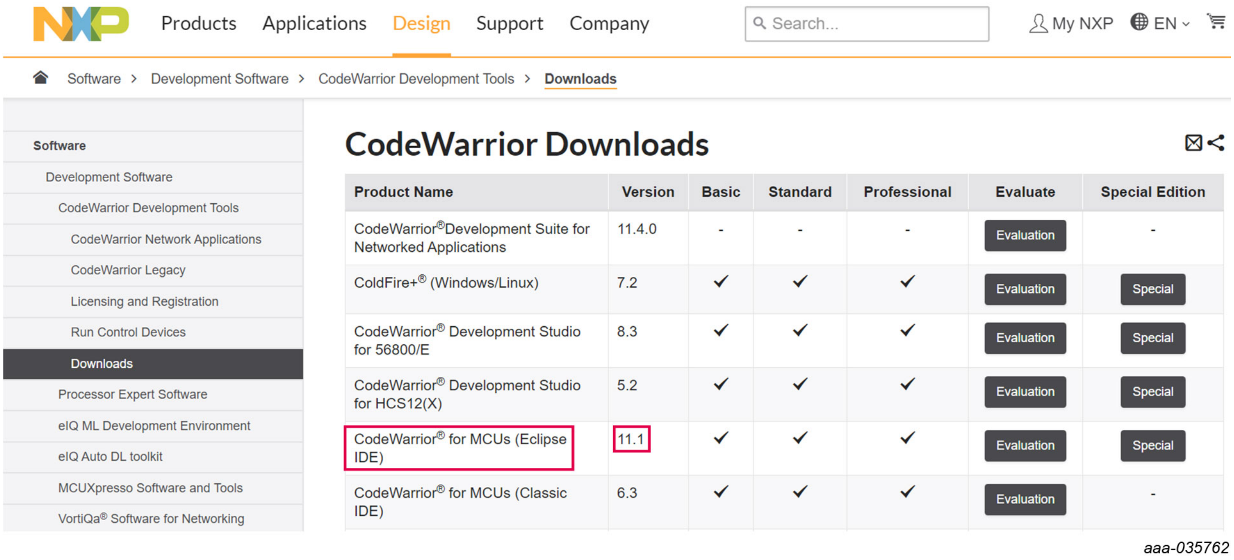
User manual

Document information

Information	Content
Keywords	CodeWarrior, IPMS, NPM8, industrial pressure sensor
Abstract	NXP offers a variety of CodeWarrior (CW) development tools on NXP.com. NXP's industrial pressure monitoring system (IPMS) solutions for NPM8Kx4S use CodeWarrior for MCUs, the Eclipse integrated development environment (IDE).

1 Installing CodeWarrior v11.1 or above

NXP offers a variety of CodeWarrior (CW) development tools on NXP.com. NXP's industrial pressure monitoring system (IPMS) solutions for NPM8Kx4S use CodeWarrior for MCUs, the Eclipse integrated development environment (IDE). To download a free, no time limit version of the IDE, visit CodeWarrior Downloads on NXP.com or navigate to NXP.com, *Support > All Developer Resources > Software > CodeWarrior Development Tools > Evaluation Software*. From the CW downloads page, select the **Special** button to download the Special Edition version of CodeWarrior for MCUs (Eclipse IDE) as shown in Figure 1.


Products
Applications
Design
Support
Company
Search...
My NXP
EN
Software > Development Software > CodeWarrior Development Tools > Downloads
Software
Development Software
CodeWarrior Development Tools
CodeWarrior Network Applications
CodeWarrior Legacy
Licensing and Registration
Run Control Devices
Downloads
Processor Expert Software
eIQ ML Development Environment
eIQ Auto DL toolkit
MCUXpresso Software and Tools
VortiQa® Software for Networking
aaa-035762

CodeWarrior Downloads

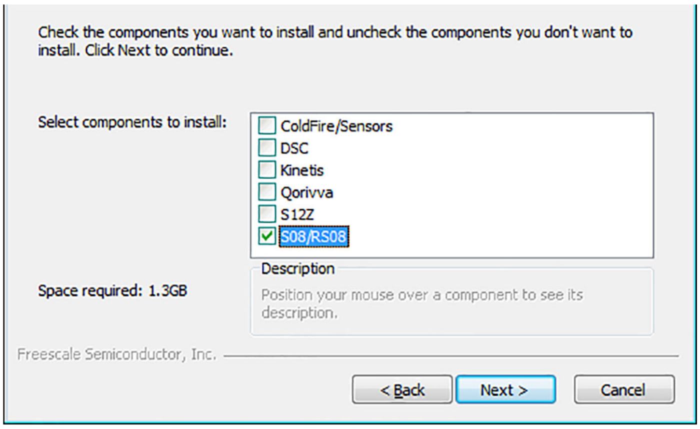
Product Name	Version	Basic	Standard	Professional	Evaluate	Special Edition
CodeWarrior®Development Suite for Networked Applications	11.4.0	-	-	-	Evaluation	-
ColdFire+® (Windows/Linux)	7.2	✓	✓	✓	Evaluation	Special
CodeWarrior® Development Studio for 56800/E	8.3	✓	✓	✓	Evaluation	Special
CodeWarrior® Development Studio for HCS12(X)	5.2	✓	✓	✓	Evaluation	Special
CodeWarrior® for MCUs (Eclipse IDE)	11.1	✓	✓	✓	Evaluation	Special
CodeWarrior® for MCUs (Classic IDE)	6.3	✓	✓	✓	Evaluation	-

Figure 1. CodeWarrior Downloads page on NXP.com

The special edition is code-size limited, but the limit is 64 kB, more than the 16 kB IPMS flash size, so the limitation does not apply to IPMS products. On the download page, if the **offline** installer is selected, the whole suite will be downloaded with the installer, and access to the internet will not be required during the installation process. If the **online** installer is selected, only the core tools will be downloaded with the installer, and the rest of the necessary components will be downloaded during the installation process; the latter thus requires access to the internet during the installation process.

Note: *During CodeWarrior software installation retain,* ***do not change*** *the default installation path proposed by the installer application. The IPMS patch installation described in Section 2 depends on the default installation path. Changing the default installation path will prevent the proper installation of the IPMS patches. For CW v11.1, the default installation path is C:\Freescale\CW MCU v11.1.*

Note: *The software installation prompts the user to select components as part of the installation as shown in Figure 2. For IPMS applications,* ***S08/RS08*** *is the only required component.*



Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- □ ColdFire/Sensors
- □ DSC
- □ Kinetis
- □ Qorivva
- □ S12Z
- ☑ S08/RS08

Space required: 1.3GB

Description

Position your mouse over a component to see its description.

aaa-035763

Figure 2. Component selection window

2 Installing the NTM88_LIB target

In the CodeWarrior IDE, the IPMS Target *NTM88_LIB* is not included by default. It has to be installed by the user patch. Instructions on how to install the patch are provided in this section.

Downloading and installing the *NTM88_LIB* patch:[1]

The patch NPM_CW_PATCH.zip installs the target NTM88_LIB to support library-based NPM8 applications. To obtain this patch, visit the NPM8 product page on NXP's website and download the NPM_CW_PATCH.zip file from the Software section.

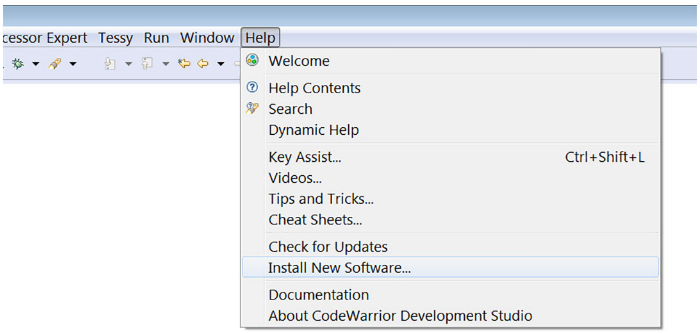
Do not unzip the patch file. The installation uses the files in zip format.

Checking the installed software version for CodeWarrior:

1. Open the CW IDE.
2. Select **Help > About CodeWarrior Development Studio.**
3. Verify CW version 11.1 or higher is installed.

CodeWarrior patch installation:

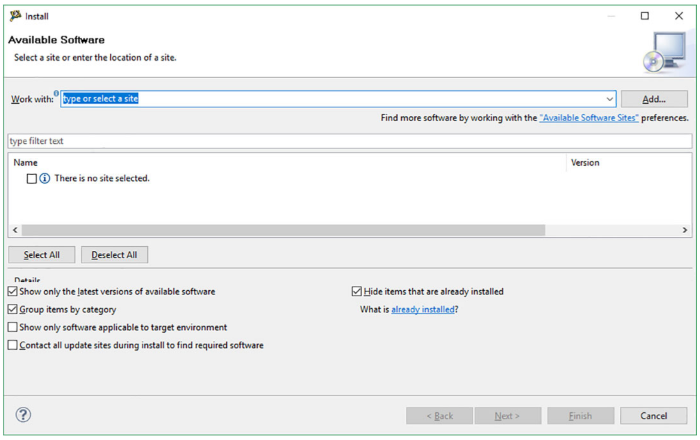
1. Ensure CW version 11.1 or higher has been installed, see Checking the installed software version for CodeWarrior.
2. Open the CW IDE.
3. Select **Help > Install New Software**.



aaa-035764

4. Selecting **Install New Software** opens the **Available Software** window.

[1] Compatible with CW version 11.1 or above



Install

Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
□ There is no site selected.	

Details

☑ Show only the latest versions of available software

☑ Group items by category

□ Show only software applicable to target environment

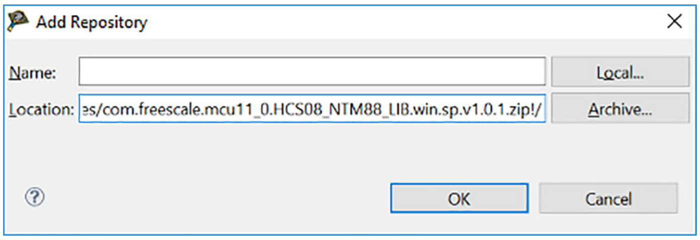
□ Contact all update sites during install to find required software

☑ Hide items that are already installed

What is already installed?

aaa-035765

5. On the **Available Software** window, select the **Add** button. The **Add Repository** window displays.



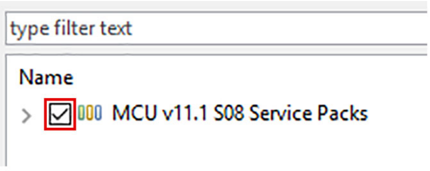
Add Repository

Name:

Location: es/com.freescale.mcu11_0.HCS08_NTM88_LIB.win.sp.v1.0.1.zip!/

aaa-035766

6. On the **Add repository** window, select the **Archive** button.
7. Navigate to the location of the first patch file to install.
8. Select the first patch zip file. Do **not** unzip the file. The file path and file name of the zip file appears in the location field of the **Add Repository** window.
9. Select the **OK** button on the **Add Repository** window. The focus returns to the **Available Software** window.
10. On the **Available Software** window, select the check box for **MCU v11.1 S08 Service Packs**.



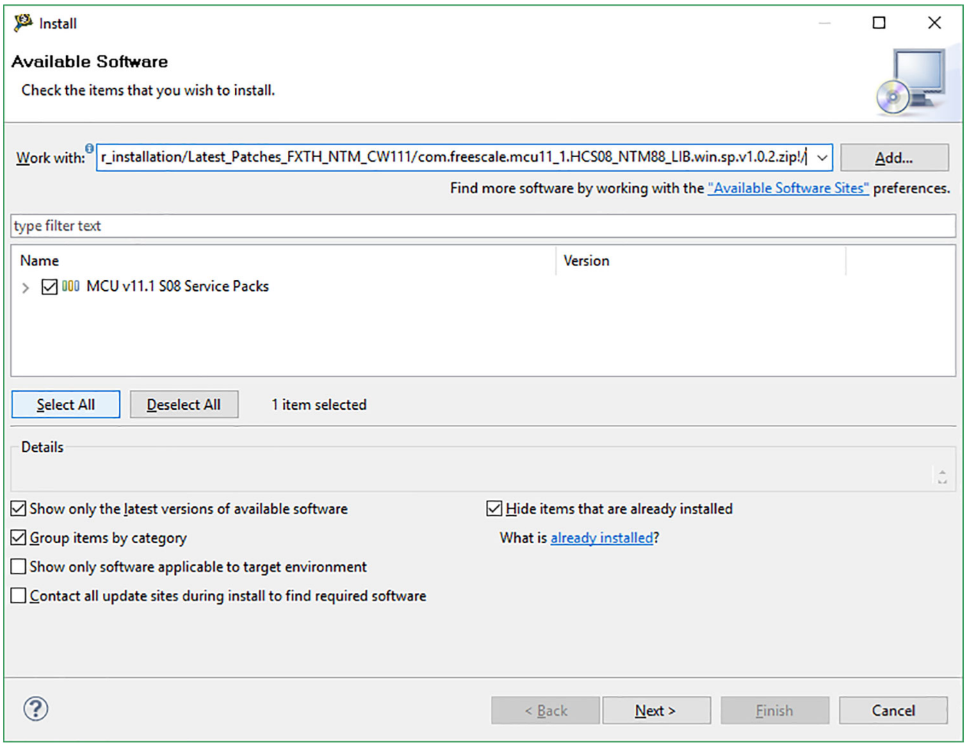
type filter text

Name

☑ MCU v11.1 S08 Service Packs

aaa-035767

11. At the bottom of the **Available Software** window, select the **Next >** button to finish the installation. If the **Next** button is grayed out and cannot be selected, click on the **Select All** button, as shown:



Available Software

Check the items that you wish to install.

Work with: r_installation/Latest_Patches_FXTH_NTM_CW111/com.freescale.mcu11_1.HCS08_NTM88_LIB.win.sp.v1.0.2.zip!/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ MCU v11.1 S08 Service Packs	

1 item selected

Details

☑ Show only the latest versions of available software

☑ Group items by category

□ Show only software applicable to target environment

□ Contact all update sites during install to find required software

☑ Hide items that are already installed

What is already installed?

aaa-035768

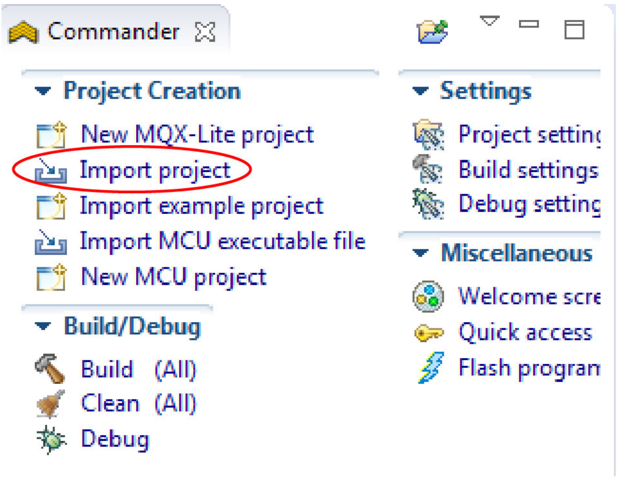
12. With the CodeWarrior IDE open, repeat Step 3 through Step 11 for any remaining patches.
13. Close and restart CodeWarrior.

Troubleshooting: during the patch installation, if the error **No repository found** occurs, this means CW IDE was not installed using the default installation path. Resolve the error by uninstalling CW and reinstalling CW using the default installation path proposed by the installer. ***Do not select a different installation path, or it will not be possible to install the patches.*** For CW v11.1, the default installation path is *C:\Freescale\CW MCU v11.1*.

3 Using CodeWarrior IDE

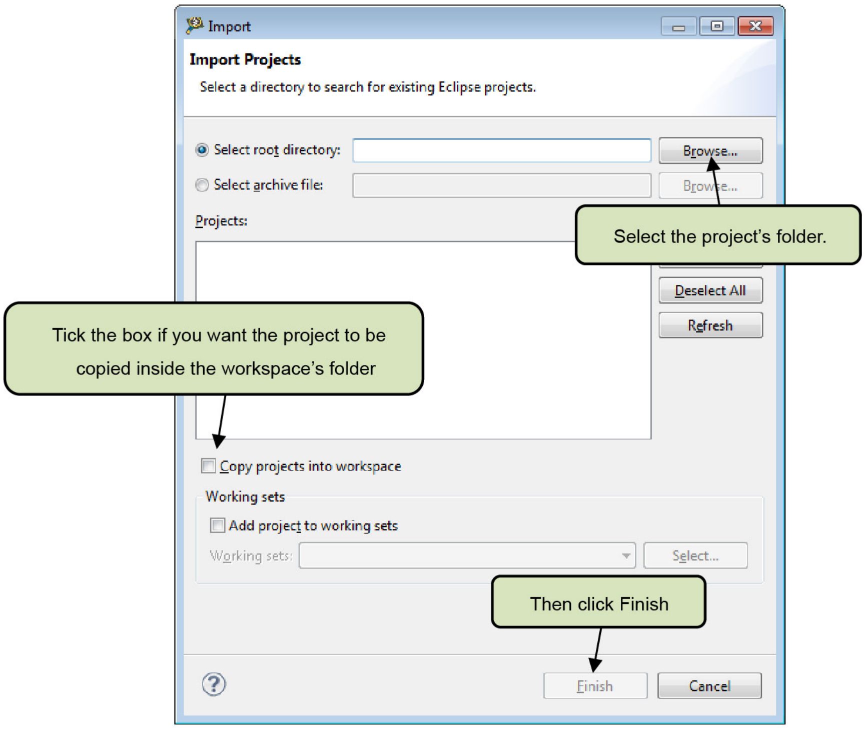
3.1 Importing a project

1. Start the CodeWarrior IDE.
2. Locate and open the **Commander** tab on the bottom left of the window.
3. Select **Import project** as shown to open the **Import** window.



aaa-035769

4. On the **Import project** window, select the **Browse** button and navigate to the location of the project to import.



Select the project's folder.

Tick the box if you want the project to be copied inside the workspace's folder

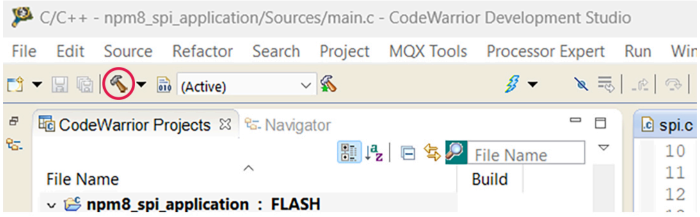
Then click Finish

aaa-035770

a. Select **Copy project into workspace** if you want the project copied inside the workspace folder.
b. Select the **Finish** button and the project is imported inside the workspace.

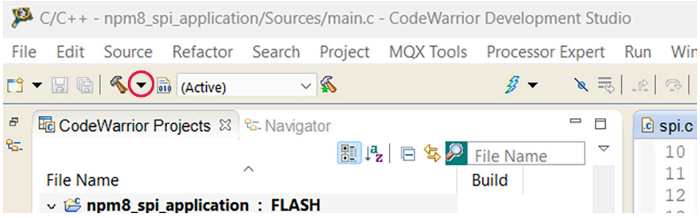
3.2 Building a project

1. To build a project, select the *Hammer* icon.
   If the *Hammer* icon is not available, double click on any source or header file of a project which will make the icon selectable.


C/C++ - npm8_spi_application/Sources/main.c - CodeWarrior Development Studio
File Edit Source Refactor Search Project MQX Tools Processor Expert Run Wir
(Active)
CodeWarrior Projects
Navigator
File Name
Build
File Name
npm8_spi_application : FLASH
spi.c
10
11
12

aaa-060434

2. In some projects, several targets are available. This is used to compile the same project with different options. To display the list of available targets, click on the *down arrow* icon next to the *Hammer* icon.


C/C++ - npm8_spi_application/Sources/main.c - CodeWarrior Development Studio
File Edit Source Refactor Search Project MQX Tools Processor Expert Run Wir
(Active)
CodeWarrior Projects
Navigator
File Name
Build
File Name
npm8_spi_application : FLASH
spi.c
10
11
12

aaa-060435

Troubleshooting: If the error **No rule to make target** occurs, clean the project before building it. To clean the project, right click on the project and select **Clean Project.**

After the project has been built, several files are created in the output folder:

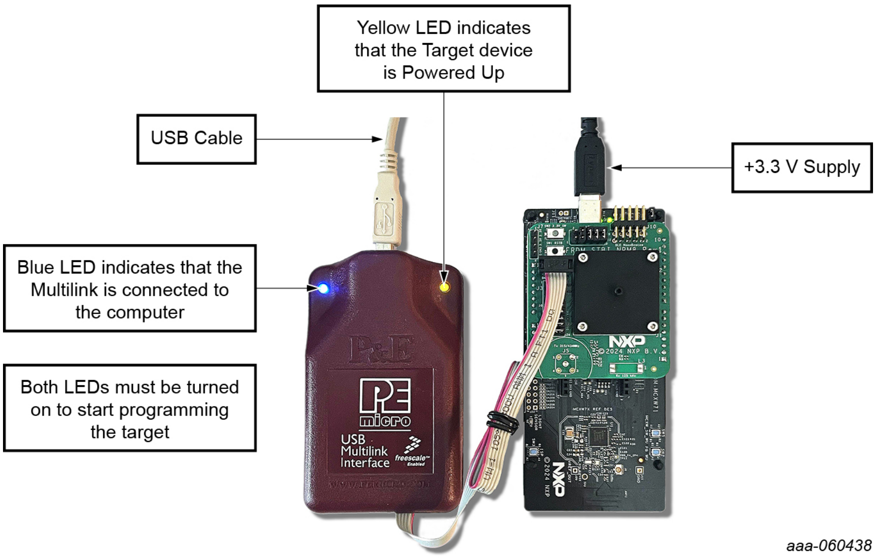
- **.abs file:** The .abs file contains the executable information and debugging information. The .abs file is used as input by the debugger.
- **.s19 file:** The .s19 file contains executable information only. The .s19 file is used as input by the flash programmer tool.
- **.map file:** The .map file shows the memory location of the objects used by the application. The .map file addresses of all global variables, constants and functions are indicated in this file.

3.3 Hardware setup to program a project

To program the IPMS, a USB Multilink Universal device from P&E Micro[2] is required. For more information about this product, visit the project summary page at NXP.com or search http://www.nxp.com for "*U-MULTILINK*".

The USB Multilink Universal device connects to the IPMS Shield EVB via the BDM connector (2x3) as shown in the following figure.

[2] The P&E Micro Multilink Universal Multilink Development Interface is not a production tool. For more information, visit the PE Micro website.


Yellow LED indicates that the Target device is Powered Up
USB Cable
+3.3 V Supply
Blue LED indicates that the Multilink is connected to the computer
Both LEDs must be turned on to start programming the target
USB Multilink Interface

aaa-060438

Figure 3. P&E Multilink device connected to IPMS Shield EVB via BDM connector.

3.4 Programming firmware

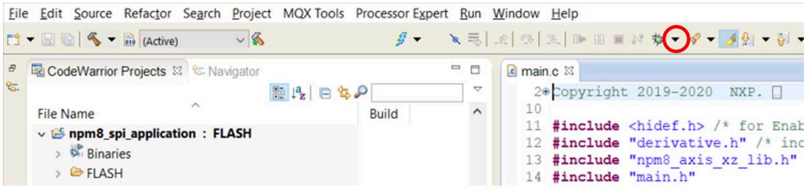
The firmware generated from this project can be programmed using either the debugger or the Flash Programmer tool.

3.4.1 Programming firmware using the debugger

If the project is programmed with the debugger, a debug session will start after the IPMS is programmed. From the debug session, the user can debug the application by executing instructions step by step, reading the content of the registers, reading the content of the memory, etc. For more information on how to keep a debug session active when the application enters STOP1 mode, refer to AN14602 - *IPMS Code Architecture for Debug* provided with the demo starter projects.

To program a project with the debugger:

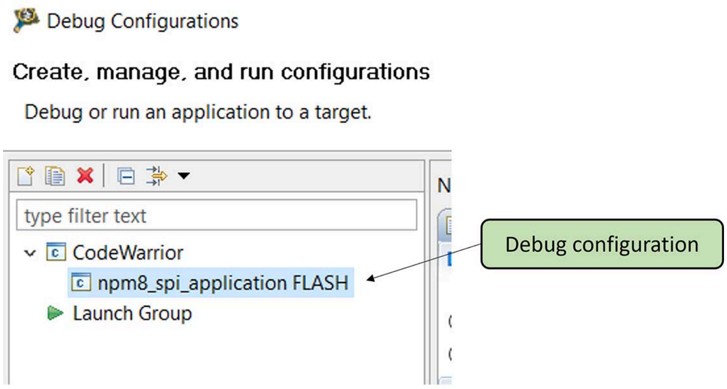
1. Open the **Debug Configurations** window by selecting the *down arrow* next to the green *debug* icon as shown:


File Edit Source Refactor Search Project MQX Tools Processor Expert Run Window Help
(Active)
CodeWarrior Projects
Navigator
File Name
Build
npm8_spi_application : FLASH
Binaries
FLASH
main.c

```
Copyright 2019-2020  NXP.

#include <hidef.h> /* for Enab
#include "derivative.h" /* inc
#include "npm8_axis_xz_lib.h"
#include "main.h"
```

aaa-060441

2. On the **Debug Configurations** window, select the configuration below **CodeWarrior**.



Debug Configurations

Create, manage, and run configurations

Debug or run an application to a target.

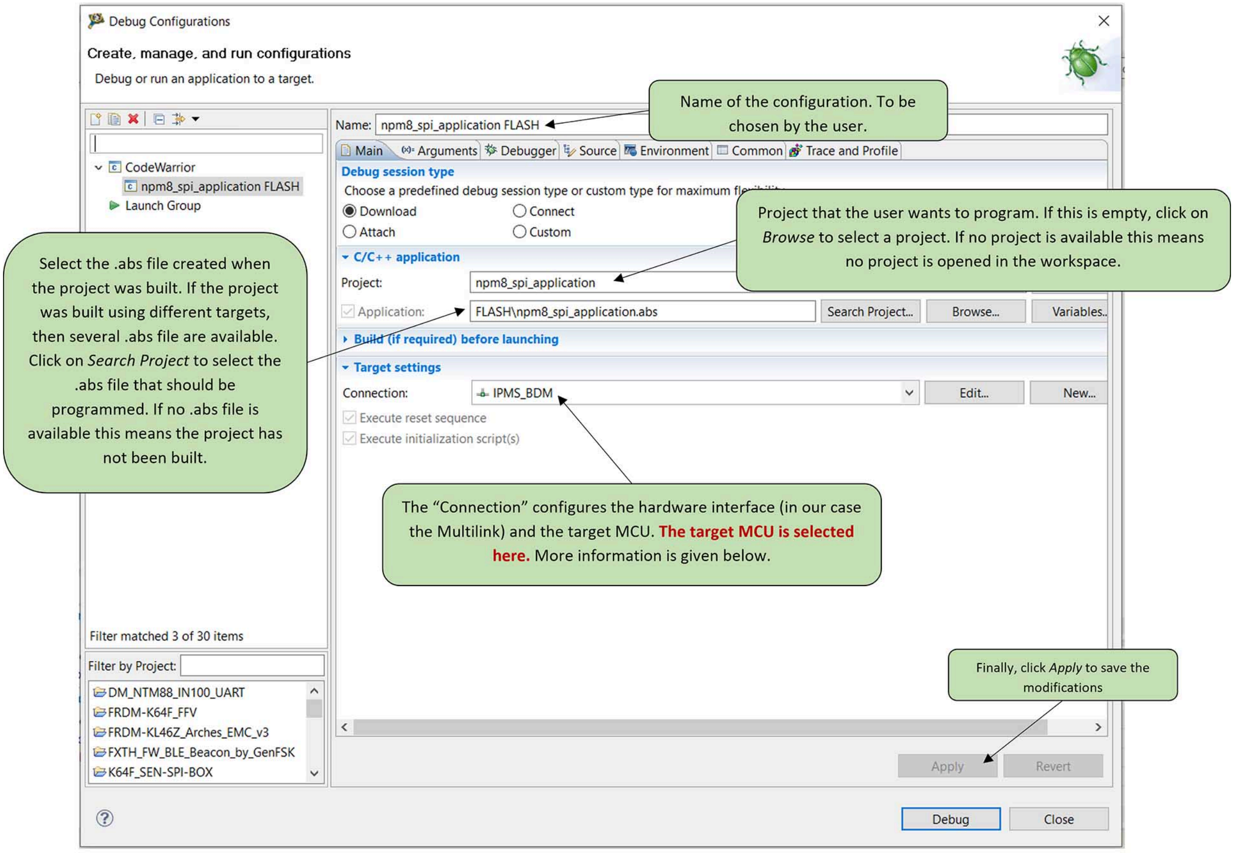
type filter text

- CodeWarrior
  - npm8_spi_application FLASH
- Launch Group

Debug configuration

aaa-060366

3. If no configuration displays below **CodeWarrior** in the window, double click on **CodeWarrior** to create one.
4. Complete the configuration in the window as shown:



Debug Configurations

Create, manage, and run configurations

Debug or run an application to a target.

- CodeWarrior
  - npm8_spi_application FLASH
- Launch Group

Name: npm8_spi_application FLASH

Main | Arguments | Debugger | Source | Environment | Common | Trace and Profile

Debug session type

Choose a predefined debug session type or custom type for maximum flexibility

● Download ○ Connect

○ Attach ○ Custom

C/C++ application

Project: npm8_spi_application

☑ Application: FLASH\npm8_spi_application.abs

Search Project... Browse... Variables...

Build (if required) before launching

Target settings

Connection: IPMS_BDM

Edit... New...

☑ Execute reset sequence

☑ Execute initialization script(s)

Filter matched 3 of 30 items

Filter by Project:

DM_NTM88_IN100_UART
FRDM-K64F_FFV
FRDM-KL46Z_Arches_EMC_v3
FXTH_FW_BLE_Beacon_by_GenFSK
K64F_SEN-SPI-BOX

Apply Revert

Debug Close

Name of the configuration. To be chosen by the user.

Project that the user wants to program. If this is empty, click on *Browse* to select a project. If no project is available this means no project is opened in the workspace.

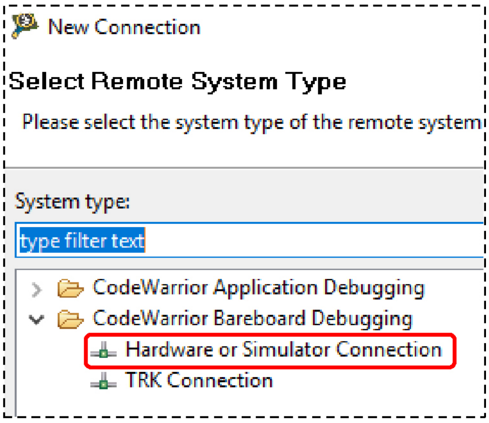
Select the .abs file created when the project was built. If the project was built using different targets, then several .abs file are available. Click on *Search Project* to select the .abs file that should be programmed. If no .abs file is available this means the project has not been built.

The “Connection” configures the hardware interface (in our case the Multilink) and the target MCU. The target MCU is selected here. More information is given below.

Finally, click *Apply* to save the modifications

aaa-060367

5. **Connection** configuration:
   a. Click **Edit** to view the configuration of the Connection.
   b. … or create a new **Connection** configuration: Select the **New** button.
   c. On the **New Connection** window, select **Hardware or Simulator Connection**.



New Connection

Select Remote System Type

Please select the system type of the remote system

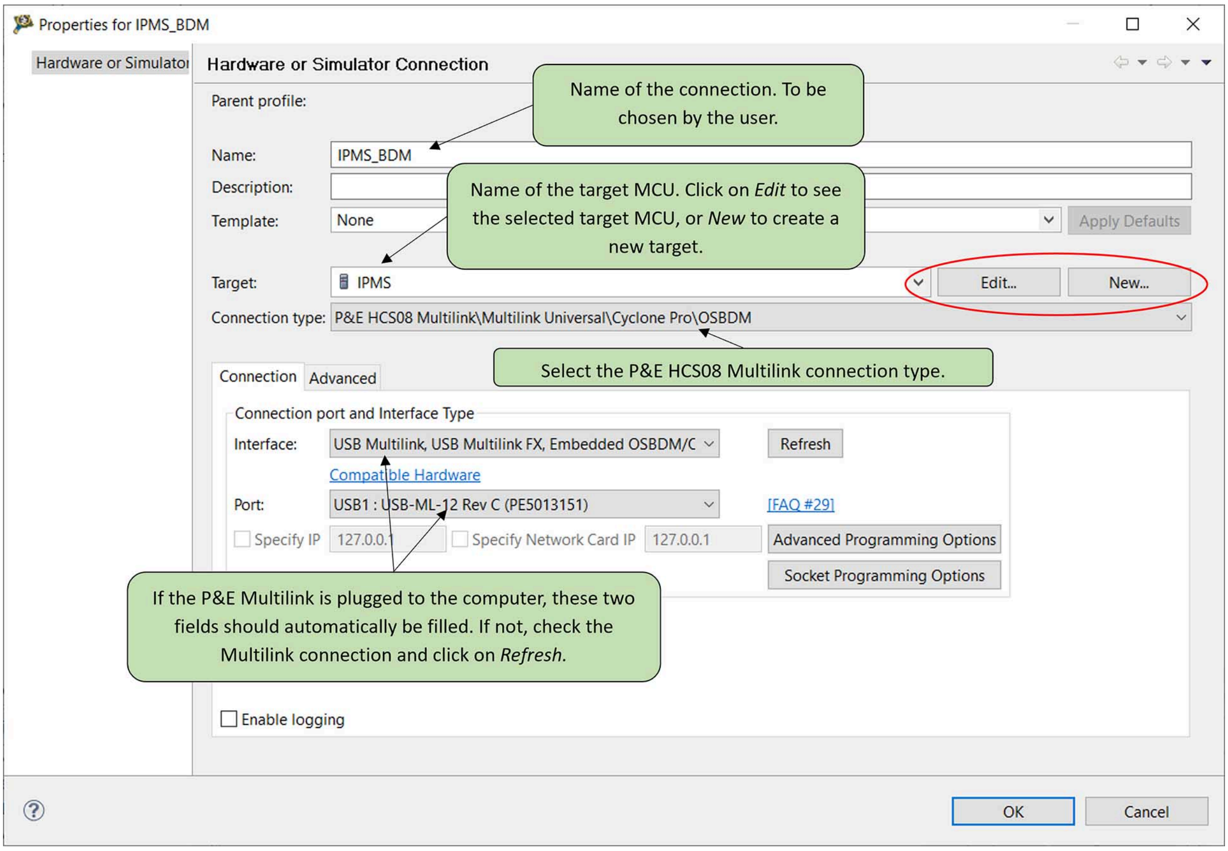
System type:

type filter text

- CodeWarrior Application Debugging
- CodeWarrior Bareboard Debugging
  - Hardware or Simulator Connection
  - TRK Connection

aaa-035783

d. The **Properties for IPMS_BDM** window appears:



Properties for IPMS_BDM

Hardware or Simulator

Hardware or Simulator Connection

Parent profile:

Name: IPMS_BDM

Description:

Template: None

Apply Defaults

Target: IPMS

Edit... New...

Connection type: P&E HCS08 Multilink\Multilink Universal\Cyclone Pro\OSBDM

Connection Advanced

Connection port and Interface Type

Interface: USB Multilink, USB Multilink FX, Embedded OSBDM/C

Refresh

Compatible Hardware

Port: USB1 : USB-ML-12 Rev C (PE5013151)

[FAQ #29]

□ Specify IP 127.0.0.1 □ Specify Network Card IP 127.0.0.1

Advanced Programming Options

Socket Programming Options

□ Enable logging

OK Cancel

Name of the connection. To be chosen by the user.

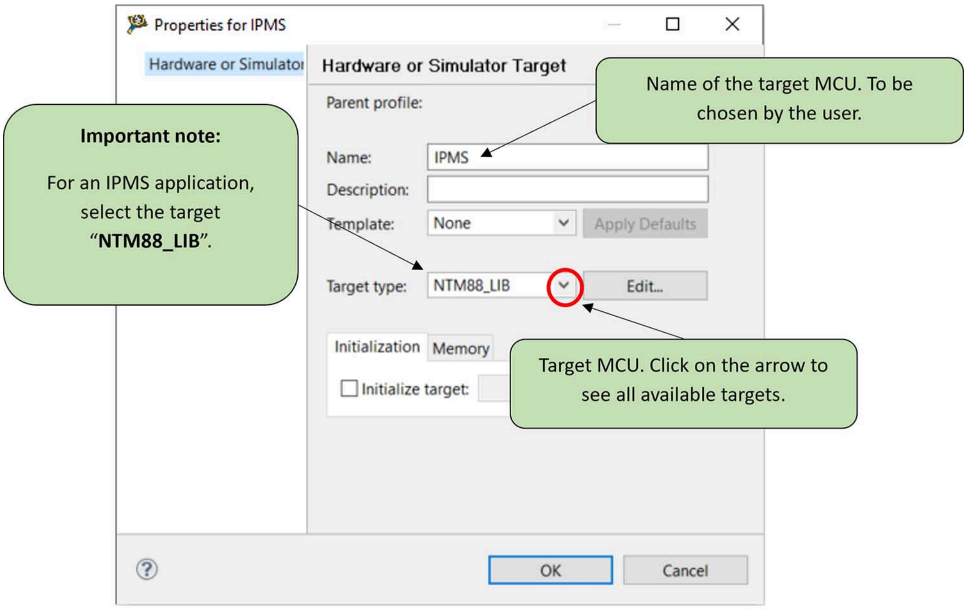
Name of the target MCU. Click on *Edit* to see the selected target MCU, or *New* to create a new target.

Select the P&E HCS08 Multilink connection type.

If the P&E Multilink is plugged to the computer, these two fields should automatically be filled. If not, check the Multilink connection and click on *Refresh.*

aaa-060368

e. Selecting the **New** or **Edit** button next to **Target** opens a window to configure the target.



Properties for IPMS

Hardware or Simulator Target

Parent profile:

Name: IPMS

Description:

Template: None

Target type: NTM88_LIB

☐ Initialize target:

Important note:

For an IPMS application, select the target “**NTM88_LIB**”.

Name of the target MCU. To be chosen by the user.

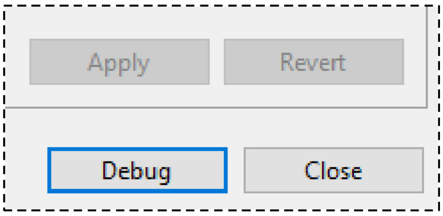
Target MCU. Click on the arrow to see all available targets.

aaa-060369

f. Complete the debug configuration window and select the **OK** button. The window focus returns to the **Properties for IPMS_BDM** window.

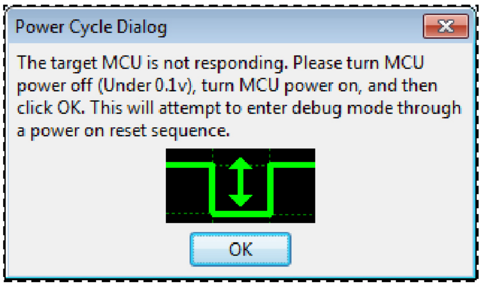
g. Apply the configuration and modifications using the **Apply** button.

6. To start the programming and debug session, click the **Debug** button.



aaa-035786

Note: During programming, the **Power Cycle Dialog** window may display with the following message:



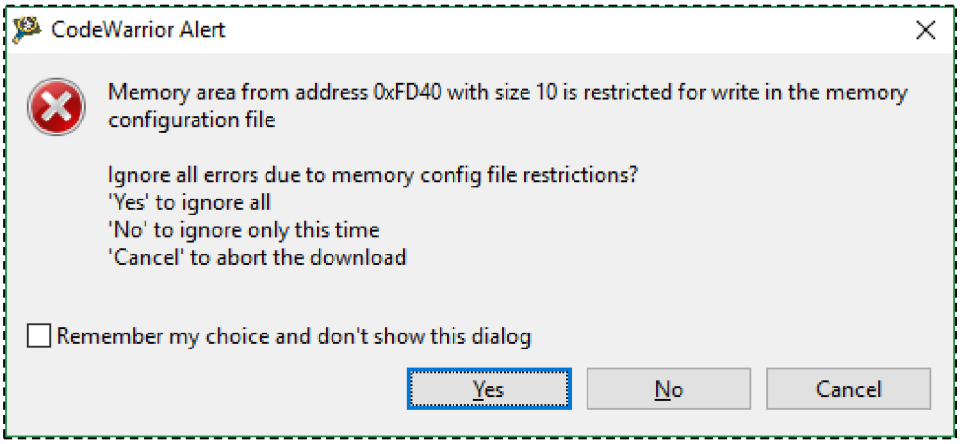
Power Cycle Dialog

The target MCU is not responding. Please turn MCU power off (Under 0.1v), turn MCU power on, and then click OK. This will attempt to enter debug mode through a power on reset sequence.

aaa-035787

Disconnect the board power supply for a short time (about one second) by turning the power supply OFF and then ON again, without disconnecting the Multilink from the board. On the IPMS Shield board there is a Switch button (VBATT N.C.) to interrupt the power to the IPMS sensor, but this will only work when the IPMS shield board is being powered externally (J7).

However, if the IPMS Shield board is being powered via the FRDM-MCXW71, this feature will not result in the expected behavior, because the SPI lines will continue to provide power to the IPMS sensor, resulting in a non-effective power cycle. In this scenario, the best approach to power cycle the IPMS sensor is to simply unplug the USB cable from the FRDM-MCXW71 board, and plug it back in.
If using the **NTM88_LIB** target, the following **CodeWarrior Alert** window might appear:



CodeWarrior Alert

Memory area from address 0xFD40 with size 10 is restricted for write in the memory configuration file

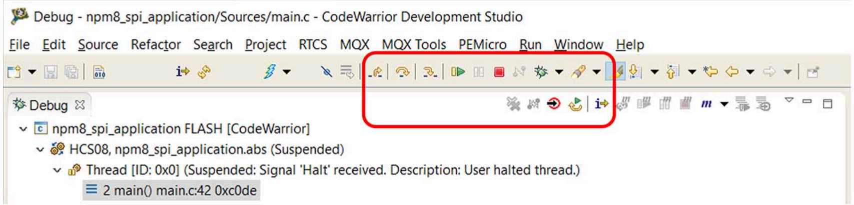
Ignore all errors due to memory config file restrictions?
'Yes' to ignore all
'No' to ignore only this time
'Cancel' to abort the download

☐ Remember my choice and don't show this dialog

aaa-035788

The **CodeWarrior Alert** window is normal for the library in use and simply alerts the user that trim coefficients are protected and cannot be overwritten. Select **Yes** to proceed to programming.

7. When the programming completes, the debug session starts. By default, the program stops at the beginning of the main function. Use the buttons shown to perform tasks such as run or pause the program, step by step debugging, reset the MCU, and terminate the debug session.



Debug - npm8_spi_application/Sources/main.c - CodeWarrior Development Studio

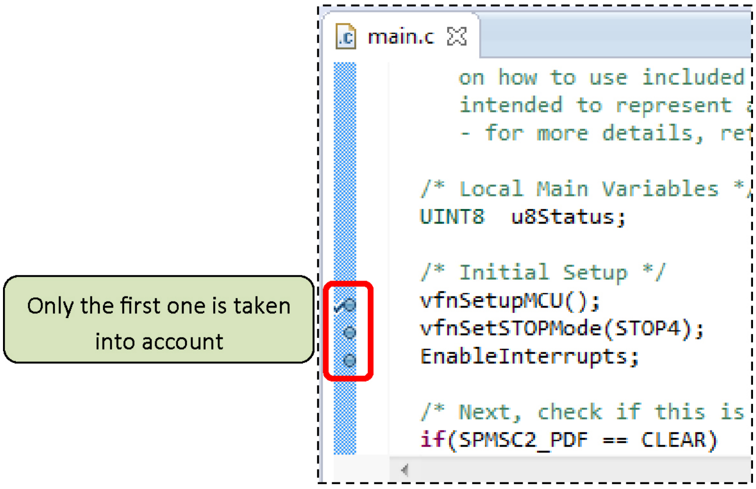
File Edit Source Refactor Search Project RTCS MQX MQX Tools PEMicro Run Window Help
Debug

npm8_spi_application FLASH [CodeWarrior]
HCS08, npm8_spi_application.abs (Suspended)
Thread [ID: 0x0] (Suspended: Signal 'Halt' received. Description: User halted thread.)
2 main() main.c:42 0xc0de

aaa-060370

Note: Only **one** breakpoint can be enabled at a time. By default, the debugger sets one breakpoint, at the beginning of the main function. It is for this reason the program stops at the beginning of the main function after the debug session launches.

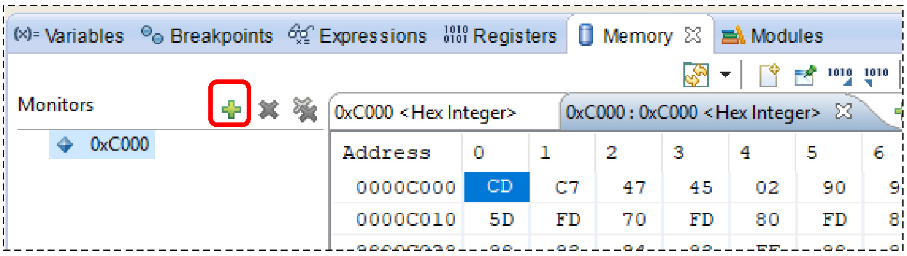
8. When the program stops at the beginning of the main function, another (and only one) breakpoint can be set elsewhere. If several breakpoints are set, only the first breakpoint is taken into account, and the other breakpoints are ignored.



```
    on how to use included
    intended to represent
    - for more details, re

/* Local Main Variables */
UINT8  u8Status;

/* Initial Setup */
vfnSetupMCU();
vfnSetSTOPMode(STOP4);
EnableInterrupts;

/* Next, check if this is
if(SPMSC2_PDF == CLEAR)
```

aaa-035791

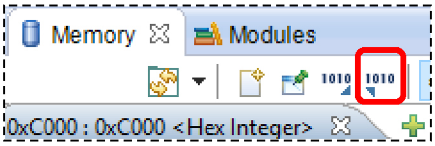
9. With the program paused or stopped at a breakpoint, the user can check the values of variables, the content of the registers and the content of the memory. Select the green *plus* sign (highlighted in red) to indicate the starting address of the section to monitor.



```
Address    0   1   2   3   4   5   6
0000C000   CD  C7  47  45  02  90  9
0000C010   5D  FD  70  FD  80  FD  8
```

aaa-035792

10. On the **Memory** tab, the content of the memory can be exported. Note that when using the **NTM88_LIB** target, the content of the trim section is hidden, so the user will see the value **0xBA** instead of the actual trim values. By default, the **Export memory** button is not enabled: enter an address to monitor using the green *plus* sign icon highlighted in the prior image. Select the now active **Export** button.



aaa-035793

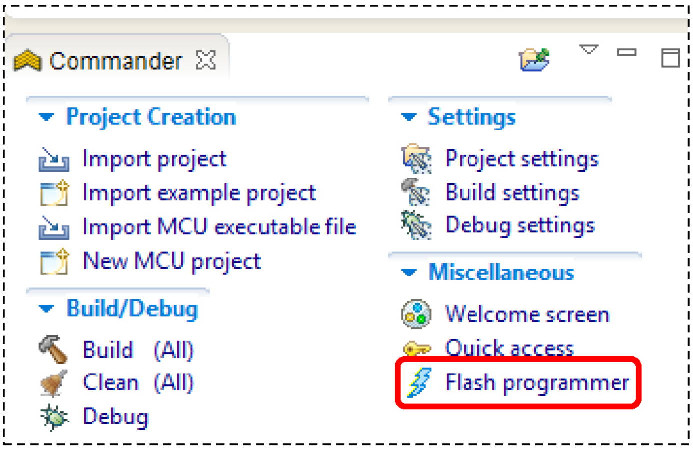
Important note: When the program starts after programming, the MCU enters a mode that consumes more power than specified in the data sheet. To restart the MCU in Normal mode, a Power-On Reset must be completed. After the user terminates the debug session, disconnect the Multilink and perform a POR to restart the MCU in Normal mode and reach the expected power consumption specified in the data sheet.

3.4.2 Programming a project using the flash programmer

To program a project using the flash programmer without starting a debug session:

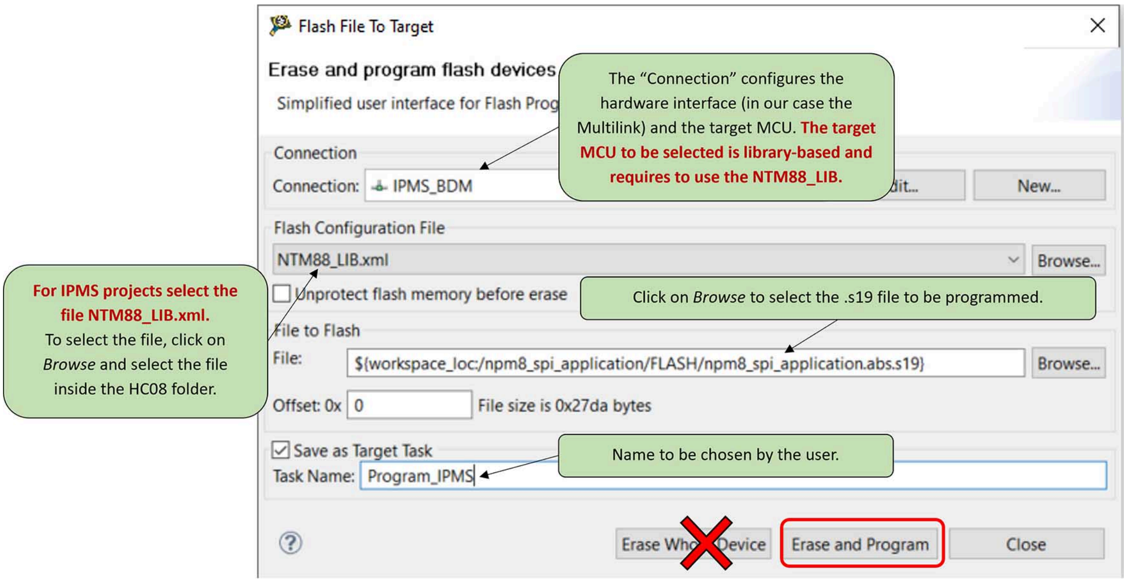
1. Start the CodeWarrior IDE.
2. Locate and open the **Commander tab** on the bottom left of the window.

3. Select **Flash programmer** as shown to open the **Flash File to Target** window.



aaa-035794

4. Configure the **Flash File to Target** window as shown:



Connection: IPMS_BDM

NTM88_LIB.xml

□ Unprotect flash memory before erase

File: ${workspace_loc:/npm8_spi_application/FLASH/npm8_spi_application.abs.s19}

Offset: 0x 0 File size is 0x27da bytes

☑ Save as Target Task

Task Name: Program_IPMS

The “Connection” configures the hardware interface (in our case the Multilink) and the target MCU. The target MCU to be selected is library-based and requires to use the NTM88_LIB.

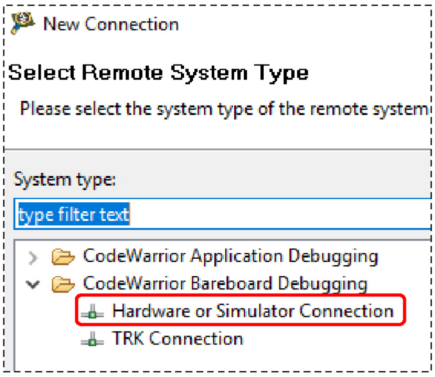
For IPMS projects select the file NTM88_LIB.xml. To select the file, click on *Browse* and select the file inside the HC08 folder.

Click on *Browse* to select the .s19 file to be programmed.

Name to be chosen by the user.

aaa-060371

5. **Connection** configuration:
   a. Select the **Edit** button to view the configuration of the connection.
   b. … or select the **New** button to create a new connection.
   c. When the **New Connection** window opens, select **Hardware or Simulator Connection**.



New Connection

Select Remote System Type

Please select the system type of the remote system

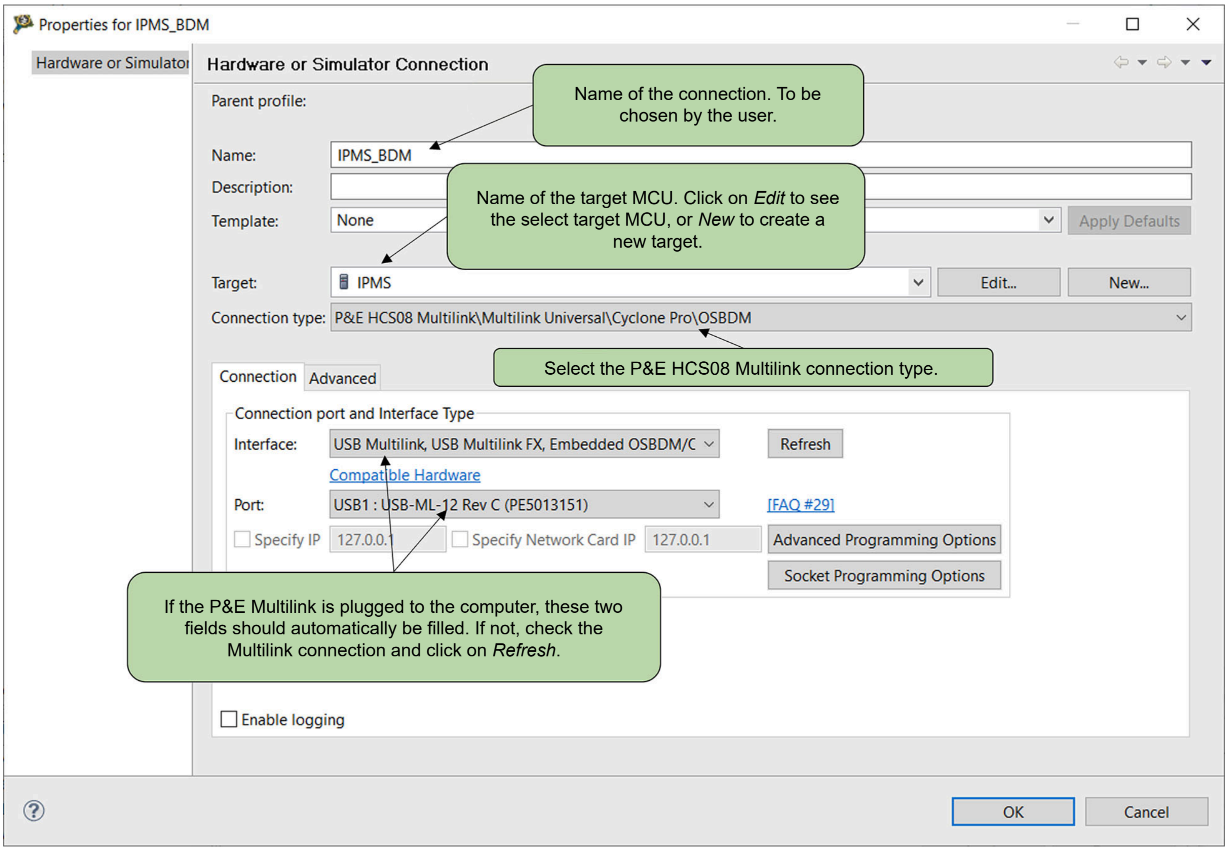
System type:

type filter text

- CodeWarrior Application Debugging
- CodeWarrior Bareboard Debugging
  - Hardware or Simulator Connection
  - TRK Connection

aaa-035796

d. The **Properties for IPMS_BDM** window opens.



Properties for IPMS_BDM

Hardware or Simulator Connection

Parent profile:

Name: IPMS_BDM

Description:

Template: None

Target: IPMS

Connection type: P&E HCS08 Multilink\Multilink Universal\Cyclone Pro\OSBDM

Connection port and Interface Type

Interface: USB Multilink, USB Multilink FX, Embedded OSBDM/C

Port: USB1 : USB-ML-12 Rev C (PE5013151)

□ Specify IP 127.0.0.1 □ Specify Network Card IP 127.0.0.1

□ Enable logging

Name of the connection. To be chosen by the user.

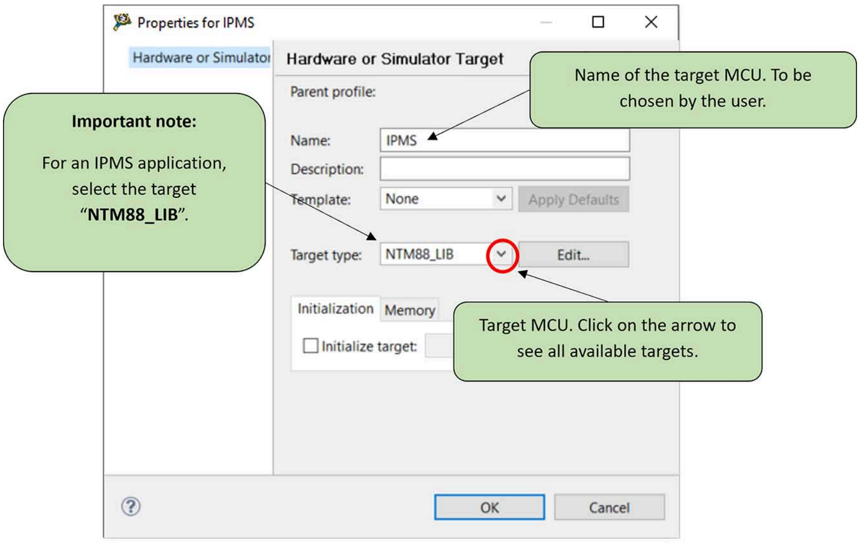
Name of the target MCU. Click on *Edit* to see the select target MCU, or *New* to create a new target.

Select the P&E HCS08 Multilink connection type.

If the P&E Multilink is plugged to the computer, these two fields should automatically be filled. If not, check the Multilink connection and click on *Refresh*.

aaa-060372

e. Clicking on the **Edit** or **New** buttons for **Target** opens the target configuration window.



Properties for IPMS

Hardware or Simulator Target

Parent profile:

Name: IPMS

Description:

Template: None

Target type: NTM88_LIB

Initialization | Memory

☐ Initialize target:

Important note:

For an IPMS application, select the target “**NTM88_LIB**”.

Name of the target MCU. To be chosen by the user.

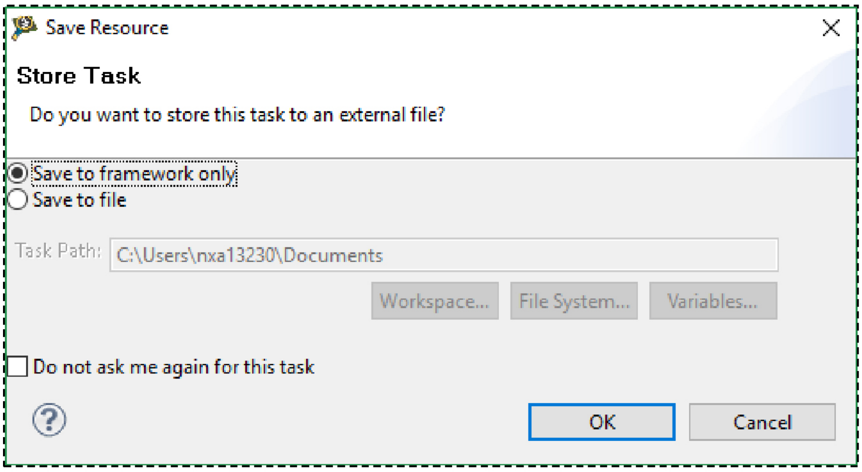
Target MCU. Click on the arrow to see all available targets.

aaa-060436

f. Configure the target as desired
   Select the **OK** button on the target configuration window.

g. Select the **OK** button on the **Properties for IPMS_BDM** window.

6. After configuring the fields, select the **Erase and Program** button.
   Do **NOT** select the **Erase Whole Device** button. Selecting this button erases ***all flash***, including the trim coefficients. If the trim coefficients are erased, the sensors will be permanently rendered non-functional.
7. The **Save Resource** windows opens.
   - To store the settings in the workplace, select the **Save to framework only** radio button.
   - To store the settings in a file, select the **Save to file** radio button. Storing the settings in a file enables the user to import these settings into other workspaces.



Save Resource

Store Task

Do you want to store this task to an external file?

● Save to framework only

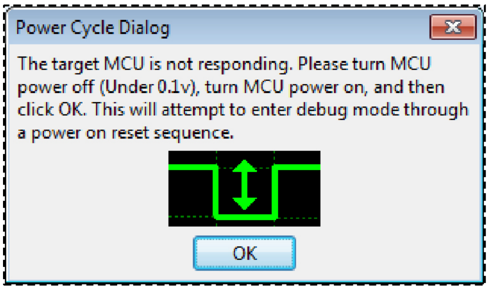
○ Save to file

Task Path: C:\Users\nxa13230\Documents

☐ Do not ask me again for this task

aaa-035799

During programming, the following **Power Cycle Dialog** message may appear:



Power Cycle Dialog

The target MCU is not responding. Please turn MCU power off (Under 0.1v), turn MCU power on, and then click OK. This will attempt to enter debug mode through a power on reset sequence.

aaa-035800

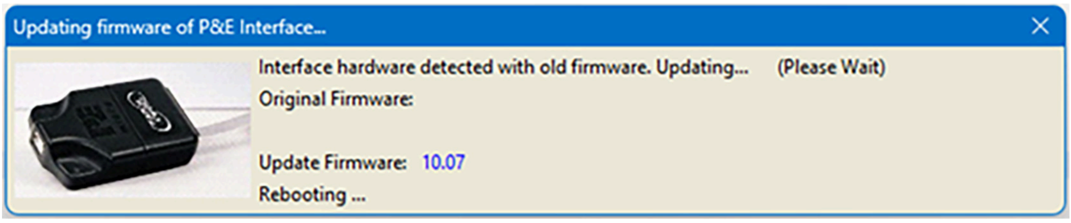
Disconnect the board power supply for a short time (about one second) by turning the power supply OFF and then ON again, without disconnecting the Multilink from the board. On NXP’s evaluation boards, a jumper enables the power supply to be disconnected.

8. When programming completes, perform a Power On Reset to restart the MCU in Normal mode to reach the expected power consumption specified in the data sheet.

3.5 Updating with new P&E Multilink firmware

P&E Multilink devices are programmed with a specific firmware for each target architecture family. The IPMS target architecture family is S08. When the firmware of the P&E Multilink is not compatible with the target to be programmed, the firmware is automatically updated before the programming is launched.

The following figure shows a P&E Multilink firmware update.



Updating firmware of P&E Interface...

Interface hardware detected with old firmware. Updating... (Please Wait)

Original Firmware:

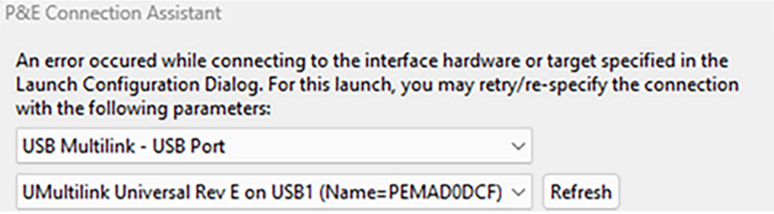
Update Firmware: 10.07

Rebooting ...

aaa-060444

Figure 4. P&E Multilink firmware update

With the latest P&E Multilink devices, the automatic firmware update may not always select the appropriate firmware version. In such a case, the following error is displayed upon completion of the firmware update.



P&E Connection Assistant

An error occured while connecting to the interface hardware or target specified in the Launch Configuration Dialog. For this launch, you may retry/re-specify the connection with the following parameters:

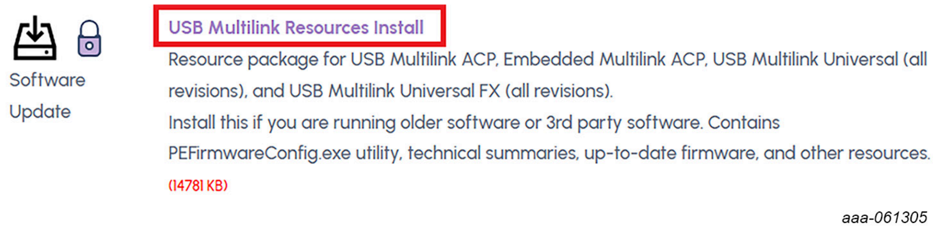
USB Multilink - USB Port

UMultilink Universal Rev E on USB1 (Name=PEMAD0DCF)

aaa-060445

Figure 5. P&E Connection Assistant

To fix this problem, do a manual update of the P&E Multilink firmware with the manufacturer's tool: Universal Multilink Firmware Update and Architecture Selection Utility. Visit the following link: https://www.pemicro.com/support/download_processor.cfm, and download and install the latest version available of the **USB Multilink Resources Install** package. See Figure 6.



USB Multilink Resources Install

Resource package for USB Multilink ACP, Embedded Multilink ACP, USB Multilink Universal (all revisions), and USB Multilink Universal FX (all revisions).

Install this if you are running older software or 3rd party software. Contains PEFirmwareConfig.exe utility, technical summaries, up-to-date firmware, and other resources.

(14781 KB)

aaa-061305

Figure 6. Getting the USB Multilink Resources Install package

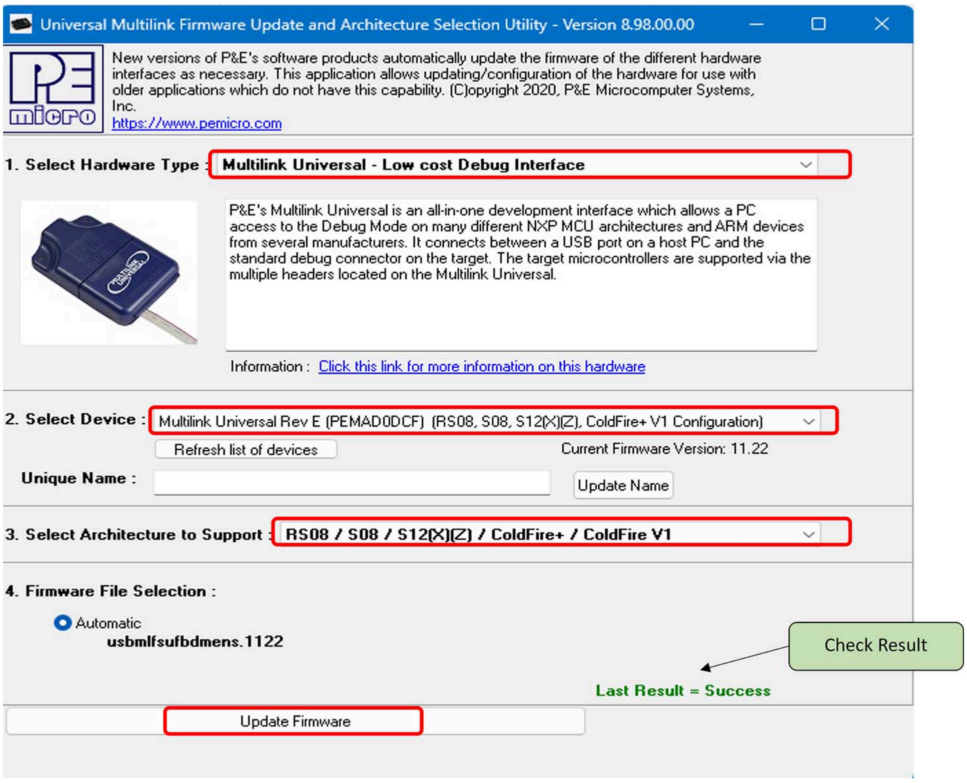
The utility can be executed from the Start Menu. Follow these steps:

Start Menu > PE Micro Multilink Resources > Firmware Update and Architecture Selection Utility

The Multilink Utility requires three configurations in order to perform the firmware Update task:

1. **Select Hardware Type**: Make sure to select the appropriate Hardware Type
2. **Select Device**: Make sure the option that contains target S08 is selected.
3. **Select Architecture to Support**: Similar to number 2, target S08 is required.

After all the settings are correct, click the **Update Firmware** button. Make sure that **Last result = Success** is displayed.



Universal Multilink Firmware Update and Architecture Selection Utility - Version 8.98.00.00

New versions of P&E's software products automatically update the firmware of the different hardware interfaces as necessary. This application allows updating/configuration of the hardware for use with older applications which do not have this capability. (C)opyright 2020, P&E Microcomputer Systems, Inc.
https://www.pemicro.com

1. Select Hardware Type : Multilink Universal - Low cost Debug Interface

P&E's Multilink Universal is an all-in-one development interface which allows a PC access to the Debug Mode on many different NXP MCU architectures and ARM devices from several manufacturers. It connects between a USB port on a host PC and the standard debug connector on the target. The target microcontrollers are supported via the multiple headers located on the Multilink Universal.

Information : Click this link for more information on this hardware

2. Select Device : Multilink Universal Rev E (PEMAD0DCF) (RS08, S08, S12(X)(Z), ColdFire+ V1 Configuration)

Current Firmware Version: 11.22

Unique Name :

3. Select Architecture to Support : RS08 / S08 / S12(X)(Z) / ColdFire+ / ColdFire V1

4. Firmware File Selection :

● Automatic
usbmlfsufbdmens.1122

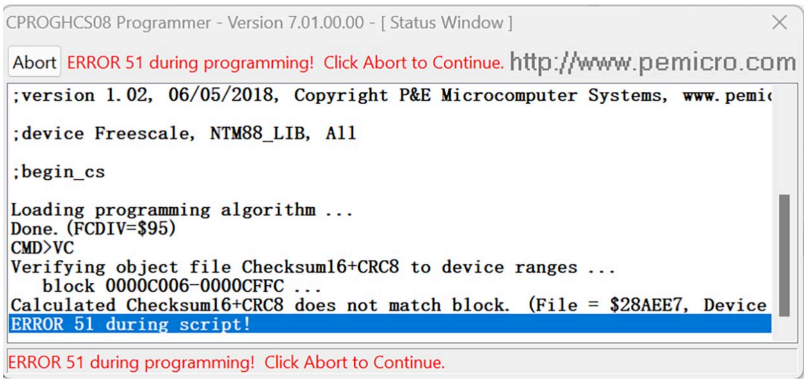
Last Result = Success

aaa-060437

Figure 7. Multilink Utility

When finished, reset the Multilink debugger and it will be ready to flash the target.

Note: *The error message in Figure 8 may pop up after Power On Reset. Such an error will not prevent a successful programming of the IPMS and can be ignored by the user.*



CPROGHCS08 Programmer - Version 7.01.00.00 - [Status Window]
Abort
ERROR 51 during programming! Click Abort to Continue.
http://www.pemicro.com

```
;version 1.02, 06/05/2018, Copyright P&E Microcomputer Systems, www.pemi

;device Freescale, NTM88_LIB, All

;begin_cs

Loading programming algorithm ...
Done. (FCDIV=$95)
CMD>VC
Verifying object file Checksum16+CRC8 to device ranges ...
   block 0000C006-0000CFFC ...
Calculated Checksum16+CRC8 does not match block.  (File = $28AEE7, Device
ERROR 51 during script!
```

ERROR 51 during programming! Click Abort to Continue.

aaa-060446

Figure 8. Error reported (see note)

4 Abbreviations

Table 1. Abbreviations

Abbreviations	Description
IPMS	Industrial pressure monitoring system
IDE	Integrated development environment
MCU	Microcontroller unit
CW	CodeWarrior
.abs	Executable and debugging information file used as input by the debugger.
.s19	Executable information file used as input by the flash programmer tool.
.map	Memory location file of all objects (global variable addresses, constants and functions) used by the application.
P&E	P&E Microcomputer Systems, Inc.
EVB	Evaluation board
POR	Power on reset

5 Revision history

Table 2. Revision history

Document ID	Release date	Description
UM12294 v. 1.1	9 July 2025	• Added Figure 6 • Added keywords and abstract
UM12294 v. 1	1 May 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.

CodeWarrior — is a trademark of NXP B.V.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

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

Document feedback

Date of release: 9 July 2025
Document identifier: UM12294