AN14379

Managing Lifecycles on MCXW71

Rev. 2.0 — 23 January 2025 Application note

Document information

Information	Content
Keywords	AN14379, MCXW71, lifecycles
Abstract	The purpose of this application note is to describe the lifecycle stages that are available to the user, how to access them, the limitations of the lifecycles, and how to transition to the next lifecycle.

1 Introduction

The purpose of this application note is to describe the lifecycle stages that are available to the user, how to access them, the limitations of the lifecycles, and how to transition to the next lifecycle.

This device supports a security lifecycle state model. The current lifecycle state determines the device functionality, debug and test port availability, and asset accessibility. The lifecycle state is controlled by the LIFECYCLE fuse value and the state values are selected so that additional fuse bits are burned to advance the state.

Note: *Since fuses control the lifecycle state, moving to a more advanced state is an irreversible and permanent process. The lifecycle can only be advanced and cannot return to a previous state.*

The Boot ROM is responsible for checking the lifecycle state. Based on the lifecycle state, the ROM determines what boot flow is used, including whether the control is passed to the application code or not. The ROM also handles the opening of test and debug ports based on the lifecycle state. If the part is in the "bricked" state or any invalid lifecycle state, then the ROM locks the part.

2 NBU and lifecycles

The Narrowband Unit (NBU) is a dedicated compute subsystem for the narrowband radio. The NBU comprises the Arm Cortex-M3 core and associated peripherals. Among the supporting peripherals are timers, a messaging unit, and dedicated flash (distinct from the SoC flash). While the NBU allows for flexibility for evolving requirements, the firmware to implement radio protocols is intended to be developed and delivered by NXP. The lifecycles that the OEM is capable of handling do not affect the behavior of the NBU, including its functionality, the test/debug ports, and the asset accessibility.

For the lifecycles described in this application note, the NBU will always require that the authentication is successful to boot appropriately. This means that the NXP-provided firmware must be in a secured binary container. Additionally, the OEM does not have debug access to the NBU or its flash.

3 OEM lifecycles

The OEM lifecycles include the following:

- OEM-Open
- OEM-Secure World Closed
- OEM-Closed
- OEM-Locked
- OEM-Returned

3.1 OEM-Open

OEM-Open is the state in which NXP delivers the chip. This state is a develop stage, in which all functionality is available to the customers.

Some of the key accessibility rights are described in Figure 1.

LIFECYCLE	Lifecycle State [0:7]	CM33 authentication required?	CM33 debug authentication required?	TZ Debug Port Authentication Required?	nTZ Debug Port authentication Requiered?	ISP Cmds	NXP Assets	OEM Assets
OEM OPEN	0000_0111	No (Boot even if authentication fails)	No	No	No	All	Restricted	Available

Figure 1. OEM-Open

The main core (Cortex-M33) has all of the debug ports open and available. This means that plain images as well as signed images will run. If an image is signed, a signature verification is performed (against the OEM RoT fuses - CUST_PROD_OEMFW_AUTH_PUK) and the image will be allowed to run regardless of the signature-verification result.

This stage also allows OEM to provision their own secrets/assets/keys. These can be items for authenticating images (like the RoTKTH) or the encryption symmetric keys that are typically required for the SB3 container.

TrustZone:

- User core TZ and non-TZ debug ports are open and available.

ISP availability:

- All ISP commands are allowed.

3.2 OEM-Secure World Closed

OEM-Secure World Closed is the lifecycle state in which the user has decided to lock the TrustZone's secure world. This allows to continue development in nonsecure areas while applying protection to the secure world area. To access the secure area, the debug authentication must be performed. The typical use case for this lifecycle would normally involve two vendors in the development stages.

LIFECYCLE	Lifecycle State [0:7]	CM33 authentication required?	CM33 debug authentication required?	TZ Debug Port Authentication Required?	nTZ Debug Port authentication Requiered?	ISP Cmds	NXP Assets	OEM Assets
OEM SECURE WORLD CLOSED	0000_1111	Yes	Yes	Yes	No	Limited	Restricted	Restricted

Figure 2. OEM-Secure World Closed

TrustZone:

- The User Core TZ debug is available via authentication with the OEM RoT keys.
- The User Core non-TZ debug port is open and available.

ISP availability:

- Limited ISP commands are allowed (GetProperty, Reset, SetProperty, ReceiveSbFile, and TrustProvisioning).

3.3 OEM-Closed

OEM-Closed is the lifecycle state in which the user has decided to lock both secure and nonsecure worlds or lock the part further coming from the OEM-Secure World Closed. At this point, this is the state in which the device will be used in an end product and further development is no longer expected. To access the device, perform the debug authentication.

LIFECYCLE	Lifecycle State [0:7]	CM33 authentication required?	CM33 debug authentication required?	TZ Debug Port Authentication Required?	nTZ Debug Port authentication Requiered?	ISP Cmds	NXP Assets	OEM Assets
OEM Closed	0001_1111	Yes	Yes	Yes	Yes	Limited	Restricted	Restricted

Figure 3. OEM-Closed

TrustZone:

- The User Core TZ and non-TZ debug are available via authentication with the OEM RoT keys.

ISP availability:

- Limited ISP commands are allowed (GetProperty, Reset, SetProperty, ReceiveSbFile, and TrustProvisioning).

3.4 OEM-Locked

OEM-Locked is the lifecycle state in which the user has decided to lock the device. This means that this device may continue to be updated in the field, but all debug/test ports are disabled permanently. From this state, it will not be possible to return to the OEM or to ask NXP for any type of failure analysis. The only lifecycle state into which this device will be able to progress would be the "bricked" state.

LIFECYCLE	Lifecycle State [0:7]	CM33 authentication required?	CM33 debug authentication required?	TZ Debug Port Authentication Required?	nTZ Debug Port authentication Requiered?	ISP Cmds	NXP Assets	OEM Assets
OEM Locked	1001_1111	Yes	NA	NA	NA	Limited	Restricted	Restricted

Figure 4. OEM-Locked

TrustZone:

- User core TZ and non-TZ debug ports cannot be reenabled.

Debug authentication:

- The debug authentication mechanism cannot be used.

ISP availability:

- Limited ISP commands are allowed (GetProperty, Reset, SetProperty, ReceiveSbFile, and TrustProvisioning).

3.5 OEM-Return

OEM-Return is the lifecycle state in which the end product must be returned to the OEM for failure analysis testing.

LIFECYCLE	Lifecycle State [0:7]	CM33 authentication required?	CM33 debug authentication required?	TZ Debug Port Authentication Required?	nTZ Debug Port authentication Requiered?	ISP Cmds	NXP Assets	OEM Assets
OEM Return	0011_1111	NA	No	No	No	None	Restricted	Erased

Figure 5. OEM-Return

OEM assets are erased before opening the debug ports. The user core debug is then allowed and opens both TZ and non-TZ domains (open as in no need of authentication).

4 Transitioning lifecycle with SPT

Note: *The following sections describe destructive fuse changes. Since fuses control the lifecycle state, moving to a more advanced state is an irreversible and permanent process. The lifecycle can only be advanced and cannot return to a previous state.*

The MCUXpresso Secure Provisioning Tool is a GUI-based application to simplify generation and provisioning of bootable executables on NXP MCU devices. The graphical interface provides a streamlined development flow, making it simpler to prepare, flash, and fuse images while leveraging and providing access to existing utilities.

During the development of the application the "Develop" button is at the right-hand end of the top buttons.

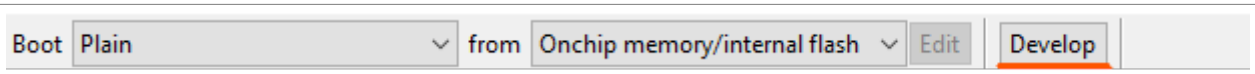

Figure 6. "Develop" button

Clicking on this button opens a prompt, as shown in Figure 7.

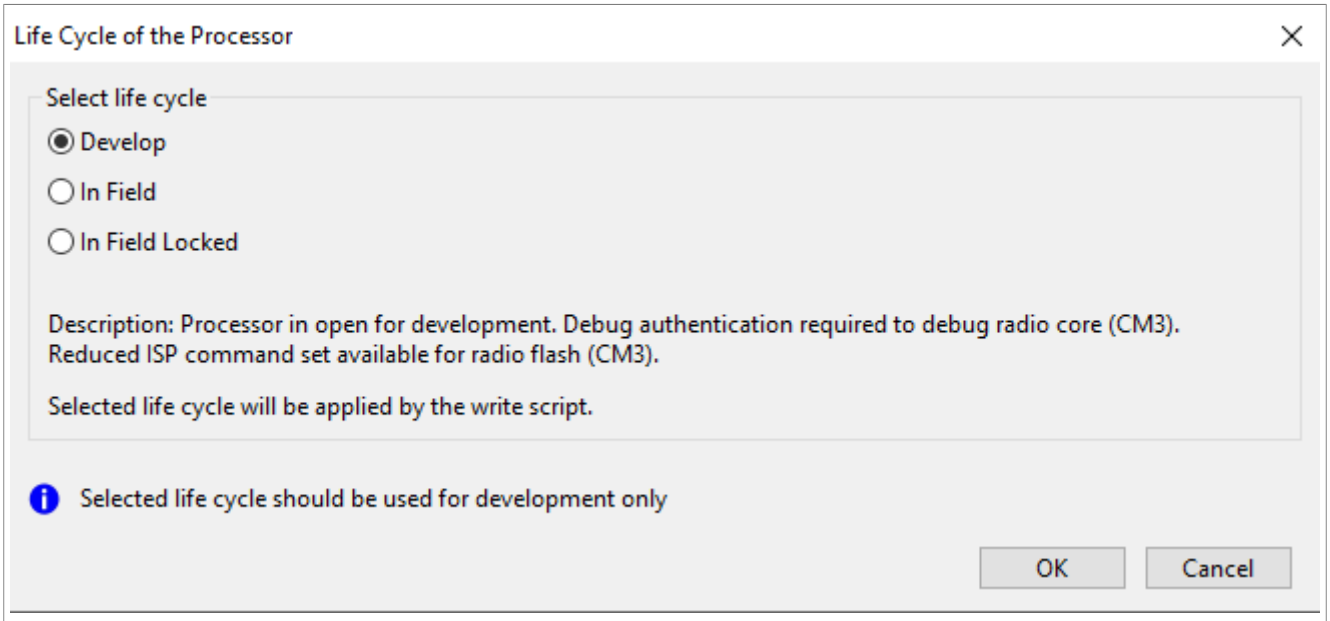

Figure 7. Prompt

In this image, "Develop" refers to OEM-Open, "In-Field" refers to OEM-Closed, and "In-Field Locked" refers to OEM-Locked.

After changing the setting and clicking "OK", the SPT tool will automatically generate the script that will run a series of commands that burn the lifecycle fuses.

5 Transitioning lifecycle with SPSDK

The latest releases of the Secure Provisioning Tool leverage low-level functionality based on the open-source Secure Provisioning SDK. SPSDK is a unified, reliable, and easy to use Python SDK library working across the NXP MCU portfolio, providing a strong foundation from quick customer prototyping up to production deployment.

The library allows the user to connect and communicate with the device, configure the device, prepare, download, and upload data, including security operations.

This application note does not describe how to install the environment needed for the SPSDK. Follow the installation steps on the SPSDK website https://spsdk.readthedocs.io/en/latest/usage/installation.html.

Once SPSDK is installed and your virtual environment is created, simply activate (venv\Scripts\activate) the existing virtual environment from your command prompt to run SPSDK.

Run the following command:

```
spsdk --help
```

If your environment and SPSDK are installed correctly, you should see the following output and you are ready to begin:

```
(venv) C:\bitbucket_spsdk>spsdk --help
Please install SPSDK with pip install 'spsdk[tp]' in order to use tphost and tpconfig apps
Usage: spsdk [OPTIONS] COMMAND [ARGS]...

  Main entry point for all SPSDK applications.

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  spsdk                                Main entry point for all SPSDK applications.
  ├── blhost                           Utility for communication with the bootloader on target.
  │   ├── batch                        Invoke blhost commands defined in command file.
  │   ├── call                         Invokes code at an address, passing an argument to it.
  │   ├── configure-memory             Sets a config at internal memory to memory with ID.
  │   ├── efuse-program-once           Writes data to a specific efuse word.
  │   ├── efuse-read-once              Returns the contents of a specific efuse word.
  │   ├── execute                      Jumps to code at the provided address.
  │   ├── fill-memory                  Fills the memory with a pattern.
  │   ├── flash-erase-all              Performs an erase of the entire flash memory.
  │   ├── flash-erase-all-unsecure     Erase complete flash memory and recover flash security section.
  │   ├── flash-erase-region           Erases one or more sectors of the flash memory.
  │   ├── flash-image                  Write the formatted image in <FILE> to the memory specified by memoryID.
  │   ├── flash-program-once           Writes provided data to a specific program once field.
  │   ├── flash-read-once              Returns the contents of a specific program once field.
  │   ├── flash-read-resource          Read resource of flash module.
  │   ├── flash-security-disable       Disable flash security by using of backdoor key.
  │   ├── fuse-program                 Program fuse.
  │   ├── fuse-read                    Reads the fuse and writes it to the file or stdout.
  │   ├── generate-key-blob            Generates the Key Blob, and writes it to the file.
  │   ├── get-property                 Queries various bootloader properties and settings.
  │   ├── key-provisioning             Group of sub-commands related to key provisioning.
  │   │   ├── enroll                   Enrolls key provisioning feature. No argument for this operation.
```

Figure 8. SPSDK installed correctly

5.1 OEM-Open to OEM-Secure World Closed

A use case for a device moved from the “Open” lifecycle to the “Secure World Closed” lifecycle would be when the OEM1 is finished with its initial development. It is planned to send the device for further development by a second OEM (OEM2). At this point, it is necessary for OEM1 to have its code/secrets/assets to be protected.

Before transitioning to the next lifecycle, OEM1 must have the following assets configured:

Secure boot:

- Root of Trust Key Hash must be configured: CUST_PROD_OEMFW_AUTH_PUK.
- Optionally, configure the SB3 encryption key: CUST_PROD_OEMFW_ENC_SK.
- IFR0 should be configured according to the use case.

TrustZone:

- TrustZone configuration can be done as part of the secure application code.
- TrustZone preset data can be included as part of the image manifest area.

PRINCE encryption/decryption for on-chip flash:

- IFR0 must be configured as needed.

Debug authentication settings:

- DBG_AUTH_VU[15:0]
- DCFG_CC_SOCU_L1[8:0]

Once OEM1 is ready to transition the lifecycle to the next one, they must place the device into the ISP mode. This can be done using the BOOT_CFG pin that is enabled by default (if available) or by issuing a debug mailbox command through SWD. In the current state (OEM-Open), it is not necessary to do debug authentication to use the DM-AP commands. Use the following command:

```
nxpdebugmbox ispmode -m 0
```

Once the device is in the ISP mode, the communication will be done through one of the following peripherals: UART, USB, I2C, SPI, or CAN. You can view the current lifecycle by reading the fuse at the index 0xA. In this example, UART is used to communicate.

```
blhost -p com7 fuse-read 0xa 4
```

The transition has three steps:

- Increase the voltage:

```
blhost -p com7 set-property 22 1
```

- Program the fuse:

```
blhost -p com7 fuse-program 0xa "{{F}}"
```

- Decrease the voltage:

```
blhost -p com7 set-property 22 0
```

5.2 OEM-Open to OEM-Closed

A use case to move a device from the “open” lifecycle to the “closed” lifecycle would be when the OEM1 is finished with its full development. It is not intended for a second OEM to further develop the device and they are ready to deploy their end product. At this point, it is necessary for OEM1 to have its complete code/secrets/assets to be protected.

Transitioning to the “closed” lifecycle state would allow for the field return to be possible. If the OEM does not want to keep this as a possibility, then the device must be advanced to the “locked” lifecycle after the device is in the “closed” state.

Before transitioning to the next lifecycle, OEM1 must have the following assets configured, if not previously done:

Secure boot:

- Root of Trust Key Hash must be configured: CUST_PROD_OEMFW_AUTH_PUK.
- Optionally, configure the SB3 encryption key: CUST_PROD_OEMFW_ENC_SK.
- IFR0 configurations would must be done via IAP calls.

TrustZone:

- TrustZone configuration can be done as part of the secure application code.
- TrustZone preset data can be included as part of the image manifest area.

PRINCE encryption/decryption for on-chip flash:

- IFR0 configurations would must be done via IAP calls.

Debug authentication settings:

- DBG_AUTH_VU[15:0]
- DCFG_CC_SOCU_L1[8:0]
- DCFG_CC_SOCU_L2[8:0]

Once the OEM is ready to transition the lifecycle to the next one, place the device into the ISP mode. This can be done using the BOOT_CFG pin that is enabled by default (if available) or by issuing a debug mailbox command through SWD. In the current state (OEM-Open), it is not necessary to do debug authentication to use the following DM-AP command:

```
nxpdebugmbox ispmode -m 0
```

Once in the ISP mode, the communication will be done through one of the following peripherals: UART, USB, I2C, SPI, or CAN. You can view the current lifecycle by reading the fuse at index 0xA. In this example, UART is used to communicate.

```
blhost -p com7 fuse-read 0xa 4
```

The transition has three steps.

- Increase the voltage:

  ```
  blhost -p com7 set-property 22 1
  ```

- Program the fuse:

  ```
  blhost -p com7 fuse-program 0xa "{{1F}}"
  ```

- Decrease the voltage:

  ```
  blhost -p com7 set-property 22 0
  ```

5.3 OEM-Secure World Closed to OEM–Closed

Another use case is moving from the “secure world closed” lifecycle to the “closed” lifecycle. It is a bit different from the previous case. Here we consider that the second OEM2 has finished with its development and the end product is ready to be deployed. Since OEM1’s assets are already protected, it is only necessary for OEM2 to protect its code/secrets/assets.

Before transitioning to the next lifecycle, OEM2 must configure the following assets, if not done previously:

Secure boot:

- Optionally, configure the SB3 encryption key: CUST_PROD_OEMFW_ENC_SK.
- IFR0 configurations would must be done via IAP calls.

TrustZone:

- A TrustZone preset data can be included as part of the image manifest area.

PRINCE encryption/decryption for on-chip flash:

- IFR0 configurations would must be done via IAP calls.

Debug authentication settings:

- DBG_AUTH_VU[15:0]
- DCFG_CC_SOCU_L2[8:0]

For this lifecycle transition, it is important to consider the limitations while in the “Secure World Closed” state. At this point, certain restrictions are in place to protect the device. One of these restrictions is that the ISP commands are limited, meaning that the fuse program is not available to burn the fuses.

If the software application in the device does not make any calls to the ROM’s API to program the fuses, then it is possible to use the following ISP command to receive a new image that includes the calls to burn the appropriate lifecycle fuse to move forward to the “Closed” lifecycle.

```
blhost -p com7 receive-sb-file new_image.sb3
```

The ROM APIs are described in the user manual. See the sections that describe the "nboot_fuse_program" and "nboot_fuse_read" for a proper use of these APIs.

5.4 OEM-Closed to OEM-Locked

A possible use case in this state is that it is used for the deployment of products to end customers in the field for products that do not support field return or failure analysis. Most of the behavior in this mode is the same as in the OEM-Closed state, but the debug and test ports can never be fully opened again. It has been determined that the field return functionality for this product would not be needed in the future.

Before transitioning to the next lifecycle, OEM1 must have the following assets configured, if not done previously:

Secure boot:

- IFR0 configurations would must be done via IAP calls.

PRINCE encryption/decryption for the on-chip flash:

- IFR0 configurations would must be done via IAP calls.

For this lifecycle transition, it is important to consider the limitations while in the “closed” state. At this point, there are certain restrictions to protect the device. One of these restrictions is that the ISP commands are limited, meaning that the fuse program is not available to burn the fuses.

If the software application in the device does not make any calls to the ROM’s API to program the fuses, then it is possible to use the following ISP command to receive a new image that includes the calls to burn the appropriate lifecycle fuse to move forward to the “locked” lifecycle.

```
blhost -p com7 receive-sb-file new_image.sb3
```

The ROM APIs are described in the user manual. See the sections that describe the "nboot_fuse_program" and "nboot_fuse_read" for a proper use of these APIs.

5.5 OEM-Closed to OEM-Return

A use case to move a device from the “closed” lifecycle to the “return” lifecycle would be when the end product is failing in the field and a failure analysis would be done on the device. Transitioning to this lifecycle will disable normal device operation and reenable the testing of the device. The failure analysis may be done by OEM1 or it can be shipped back to NXP in this lifecycle state so that NXP may transition to an internal state for further testing.

Once the OEM is ready to transition the lifecycle to the next one, it is necessary to use the debug authentication to communicate with the device. Once the device is authenticated, it is necessary to create a failure analysis message and sign it with the RoT keys.

The steps are as follows:

```
nxpdebugmbox start
```

Then you must authenticate the connection:

```
nxpdebugmbox -v -p 2.1 -i jlink auth -b 0x0 -c dc_certificate.cert -k
 private_dc_key.pem
```

Execute the famode command, which will erase customer-sensitive assets and the user flash area and transition the lifecycle to 0b0011_1111.

```
nxpdebugmbox famode -m OEM_signed_FA_msg.bin
```

The plain message can be up to 64 bytes. There are two words that are needed in this message. The first is 0x5, which defines the class for MCXW71, and the second word is 0x1, which defines OEM-Return. The rest of the message is all 0s. This must be signed by the OEM to proceed successfully.

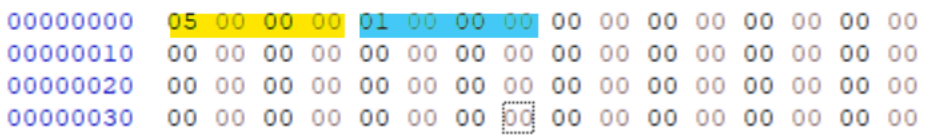

Figure 9. Plain message

6 Note about the source code in this document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7 Revision history

Table 1 summarizes the revisions to this document.

Table 1. Revision history

Document ID	Release date	Description
AN14379 v.2.0	23 January 2025	Initial public release
AN14379 v.1.0	10 September 2024	Initial NDA release

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