AN14148

Secure Boot on MCX Nx4x

Rev. 3.0 — 6 March 2025 Application note

Document information

Information	Content
Keywords	AN14148, MCX, secure boot, security, Secure Provisioning Tool
Abstract	This application note covers the steps to do secure boot using the Secure Provisioning Tool for MCX Nx4x.

1 Introduction

This application note describes the steps for secure boot using the Secure Provisioning Tool (SEC).

2 Secure boot overview

Secure boot ensures the authenticity, integrity and confidentiality of the device bootloader, firmware, and other software during the boot process. It also ensures that the intended secure lifecycle state is reached.

3 Introducing the SEC tool

The MCUXpresso Secure Provisioning Tool is a GUI-based application provided to simplify the 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 using and providing access to existing utilities. Advanced scripting can be achieved using the command-line interface, while even more advanced secure provisioning flows can be accomplished by modifying scripts generated by the tool.

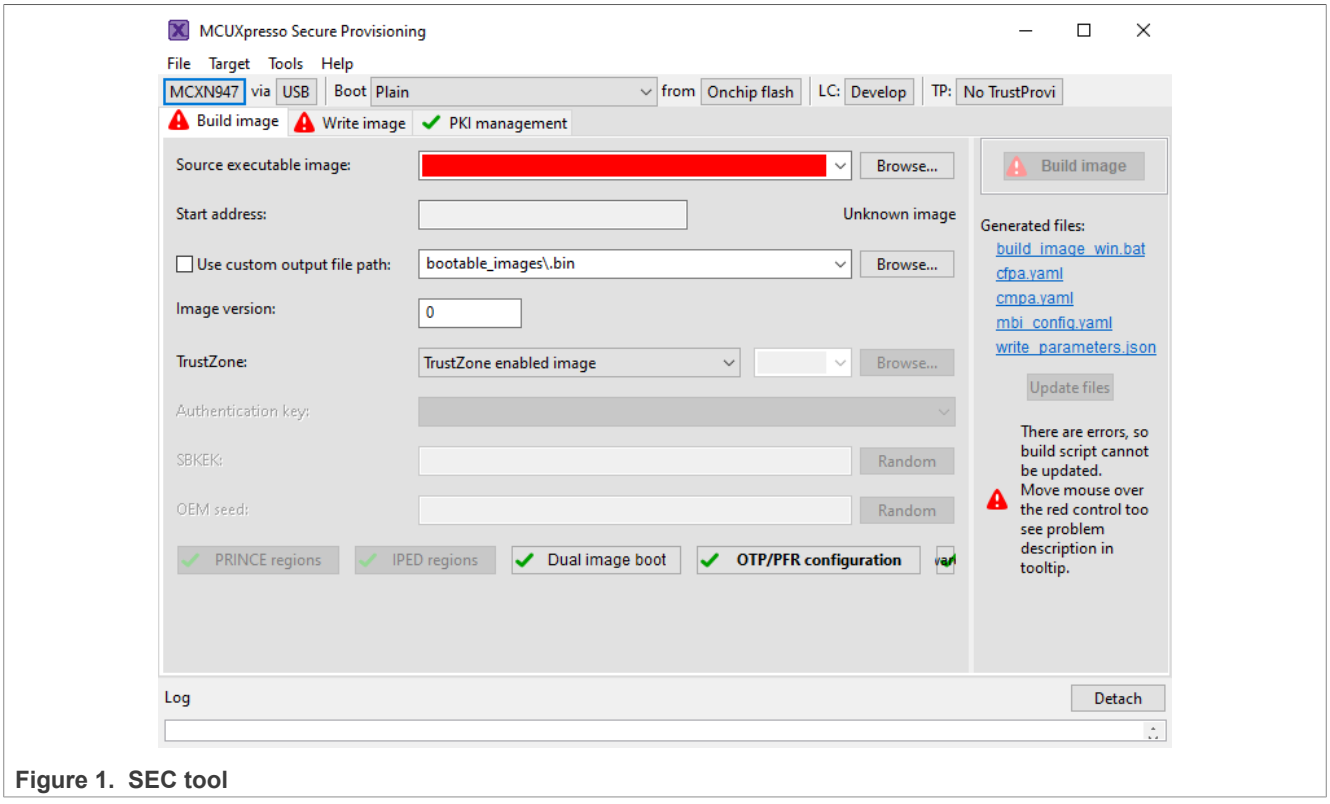

Figure 1. SEC tool

4 Key management

The device supports up to 4 Root of Trust Keys (RoTK) that can be used for different authentication purposes. Besides the boot image authentication itself, the individual RoT keys can be used for debug authentication or FW update authentication purposes.

5 Keys

Warning: *All visualized keys are used as examples only. Generate your own keys for securing your target devices.*

The chapters below contain information about keys and on how to create keys with NXP tools. NXP tools use standard key formats and the customer can generate/load their own keys.

6 ROTKH

ROTKH (Root of Trust Key Hash table) is a table generated once by the OEM and permanently stored in device fuses.

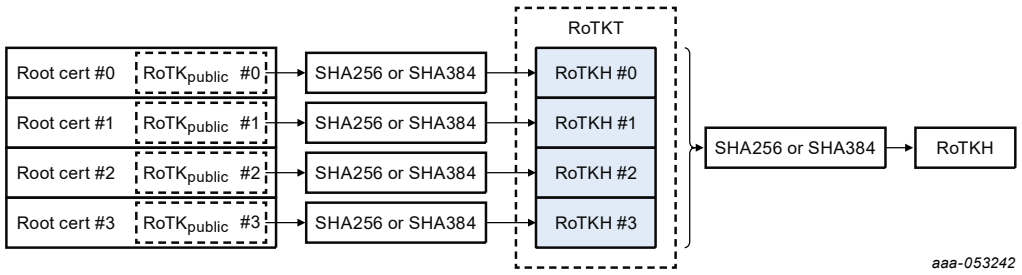

Figure 2. ROTKTH

If only 1 RoTK is specified for RoTKTH, the RoTKTH value is directly SHA256 or SHA384 of RoTKpublic.
If more than 1 root certificates are specified, RoTKTH is calculated as a hash of hashes of RoTKpublic.
Whether SHA256 or SHA384 is applied depends on the selected EC of RoTK. If secp256r1, SHA256 is used, if secp384r1, SHA384 is used as the hash algorithm.

Keys from the tool are in the standard format. The output from the tool is `.pem` and `.pub` files.

The `.pem` file contains a private key and the `.pub` file contain only a public key.

7 RoT key generation

Key generation is done only in the beginning. Based on the generated keys, the RoTKTH value is calculated and loaded in the IFR. As a result, the keys cannot be changed anymore for the device. During the development life cycle, it is expected that mass erase can be executed in IFR space, so that testing of security features can be reset. Once the device is configured for production, this space is no longer editable.

To generate a key pair, open the PKI Management tab in the tool and click **Generate Keys.**

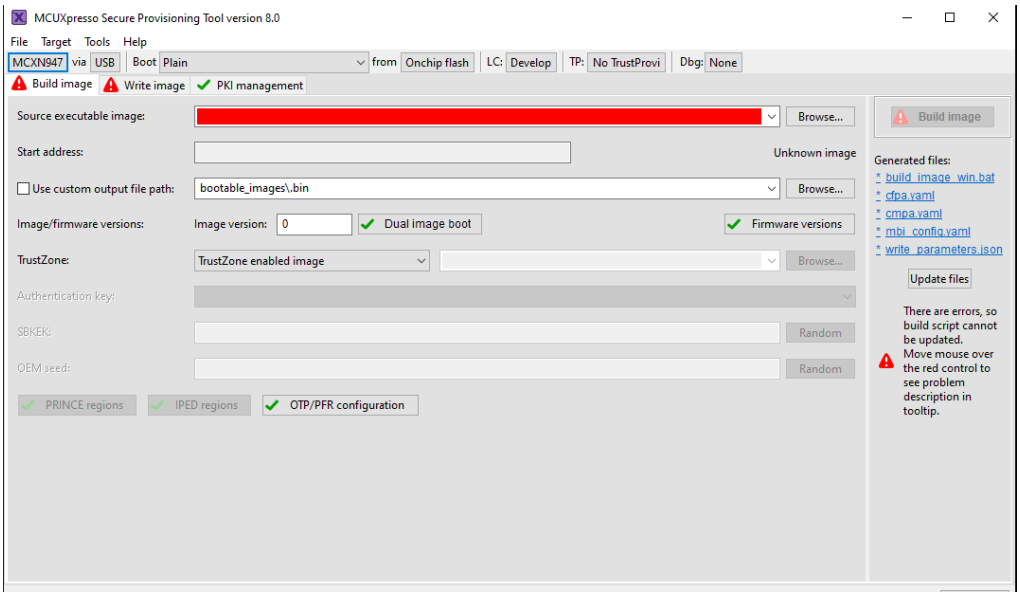

Figure 3. PKI Management tab

This button opens a window that allows specific key settings, such as the certificate chain, the size of the key and quantity.

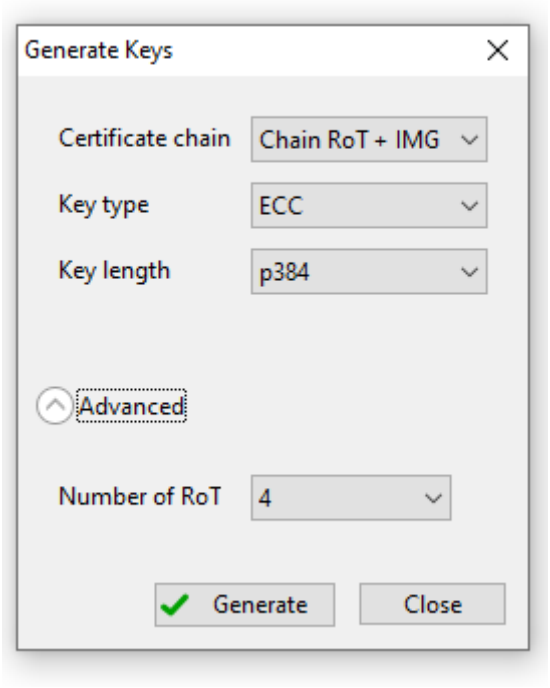

Figure 4. Generate keys

Once the settings have been chosen, click **Generate**. This button automatically creates a folder in the workspace and saves the 4 key pairs as shown below.

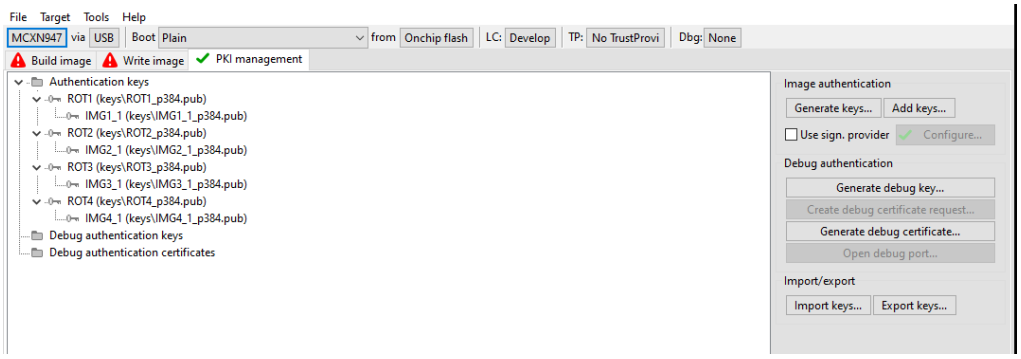

Figure 5. PKI management

8 Symmetric key generation

Our symmetric keys are used when we need encryption for the SB3.1 containers that are used in firmware updates. The tool always generates the secure binary when selecting a boot type different from the plain image.

It is important to understand that the encrypted boot selection with PRINCE is different from the encryption used in the SB3.1 and use different keys. When selecting a signed image, the secure binary is also generated.

The secure provisioning tool helps to quickly generate these symmetric keys with a click of a button. Return to the **build Image** tab and find the `CUST_MK_SK` and `OEM_seed`. They are enabled when the boot type image is different from plain. Click **Random** to generate the keys.

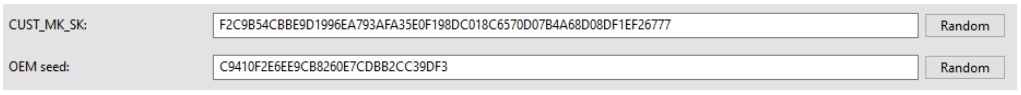

Figure 6. CUST_MK_SK and OEM_seed on the build image tab

9 Boot image type and configurations

After the roots of trust keys are created, return to the main window to configure the boot options and any other settings.

The main settings are connection, image, boot location, lifecycles, and trust provisioning settings. All of these settings are found at the top of the main window.

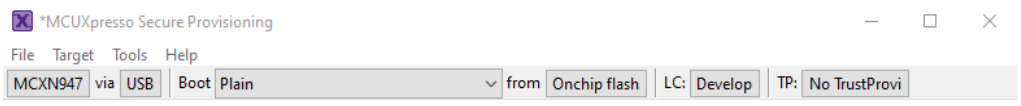

Figure 7. Toolbar

9.1 Device selection

The **MCXN947** setting has already been selected when the workspace was initially created. If there is a need to change the device, select this box and it prompts you to create a new workspace and select a new device.

9.2 ISP communication

The **USB** setting is the communication peripheral that is used via ISP mode. In this case, change this to UART setting. Select the UART port that the board is connected to. In addition, test the connection by clicking **Test**

Connection. It is important to have the ISP connection enabled. This can be done by pressing SW3, pressing RESET, releasing RESET, and releasing SW3 on the MCXN9xx-EVK or FRDM-MCXN9xx. It is also possible to enter ISP mode by pressing SW3 and connect the USB cable.

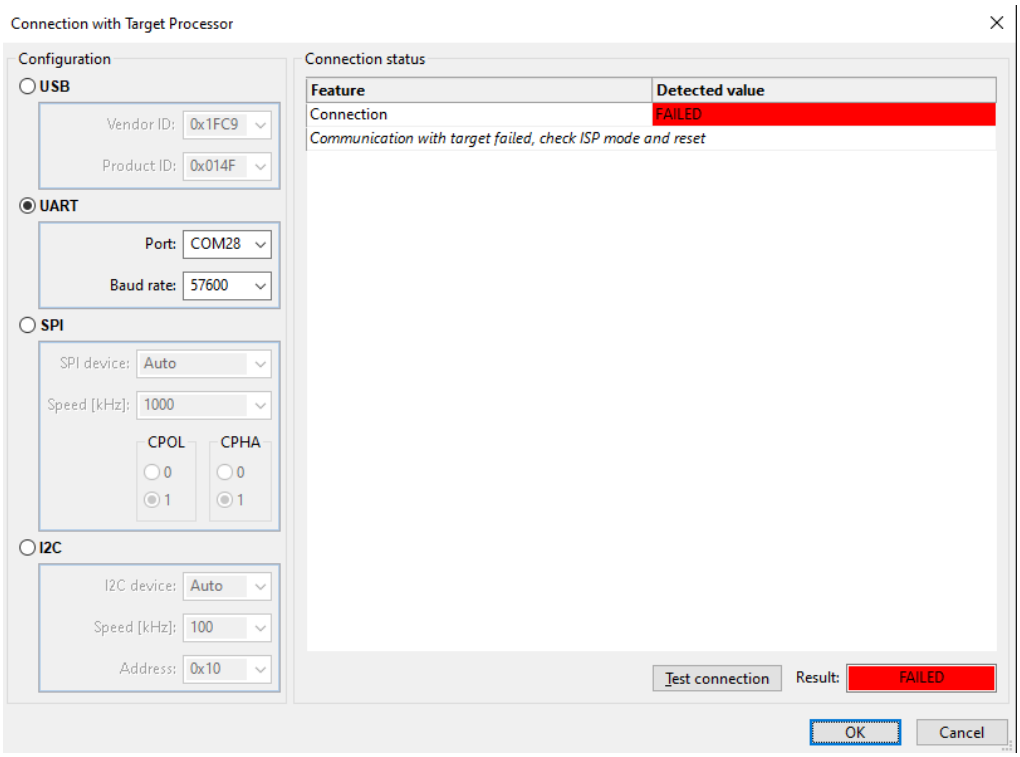

Figure 8. Connection dialog with errors and proposed next steps

9.3 Boot selection

The **Plain** image is selected by default. No keys are needed to build and write to the device. However, since we are demonstrating the use of the debug authentication, we must enable secure boot and therefore can only write signed images to our device. Select **Signed** from the drop-down menu.

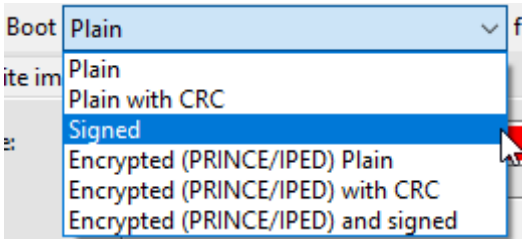

Figure 9. Boot type

9.4 Boot location

The **On Chip flash** is the internal memory in which the boot image is written to. If available, the boot location can be an external memory instead of on-chip flash. For the purposes of this application note, the default setting and internal flash are used.

9.5 Life cycle management

The **Develop** lifecycle is our default setting. For the purposes of the application note, we use the develop lifecycle. This allows us to revert the changes without erasing the PFR settings. In production, it is recommended to advance the lifecycle by selecting one of the options listed.

9.6 Trust provisioning

The **Device HSM** option for trust provisioning is automatically selected when the boot image type is different from a plain image. This selection tells the tool that the secure method of providing symmetric keys for future image updates must be used.

Note: *For further details on Device HSM provisioning and other configuration settings, see Section 12*

10 Build image

The **Build Image** tab helps to set up the main application image, generate scripts that run and write to the PFR regions, import the source image and TrustZone preset data if present. It also creates the trust provisioning image that securely sends the symmetric keys selected.

The tool comes with some test binaries that are available for different boot locations. This device includes three different images for the LED blinky demo for internal flash, external flash, and RAM. In this example internal memory is used, so select the source image that matches this boot location.

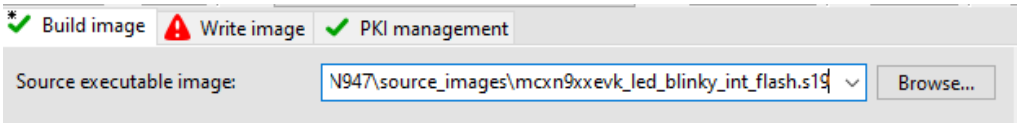

Figure 10. Choose a source image to build

The tool recognizes some of the elements from the binary chosen, such as the **Start address**. This is filled out automatically.

Figure 11. Start address

It is also possible to select a custom output file path and name. In this example, the default name written by the tool is used. The plain image is stored in the **source_images** folder of the workspace and the signed image is stored in the **bootable_images** folder.

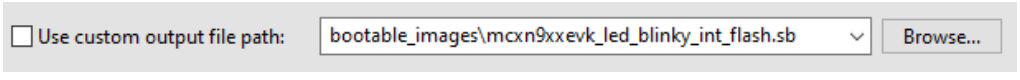

Figure 12. Output file path

The **Image version** is used for dual-boot images, so that the ROM knows which image is new. For this demonstration, it is not used so it remains a 0.

Figure 13. Image version

The **TrustZone** setting is enabled by default. Disable it as we are not using it. To make it enabled, select enable or load from a preset file. If needed, the tool also provides the template of the preset file.

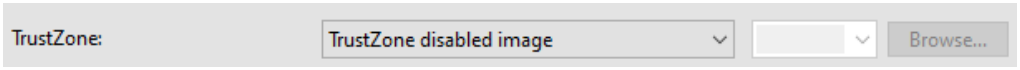

Figure 14. TrustZone settings

At this point, the keys must be already generated. Select the authentication keys and verify that the symmetric keys are available.

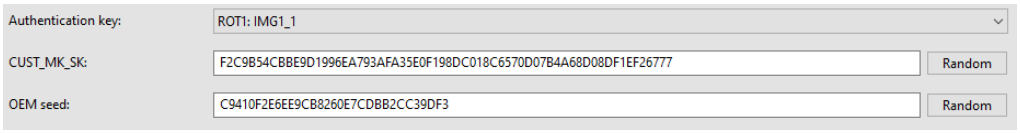

Figure 15. Key selection

There are other windows that can be opened for other settings as shown below:

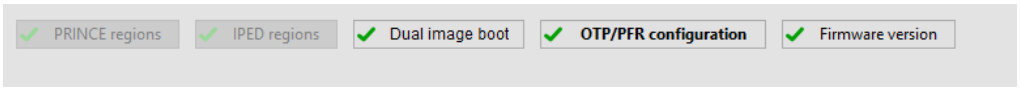

Figure 16. Other settings

In this case, modification of the **OTP/PFR configuration** is optional to give permissions after debug authentication has been enabled. The other settings are left as default.

Having the device in ISP mode, it is possible to readback the current value in the CFPA and CMPA from the device. In this case, a mass erase was done prior to beginning this exercise; therefore, the current values are filled with FFs. The required values are automatically filled based on some of the earlier configurations. Also, the permissions provided in `DCFG_CC_SOCU_NS_PIN`, `DCFG_CC_SOCU_NS_DFLT`, `DCFG_CC_SOCU_PIN` and `DCFG_CC_SOCU_DFLT` are to be modified. Do this by modifying the required value in the given field.

By making this change, we are able to verify the behavior of the debug ports as they get blocked unless debug authentication is executed.

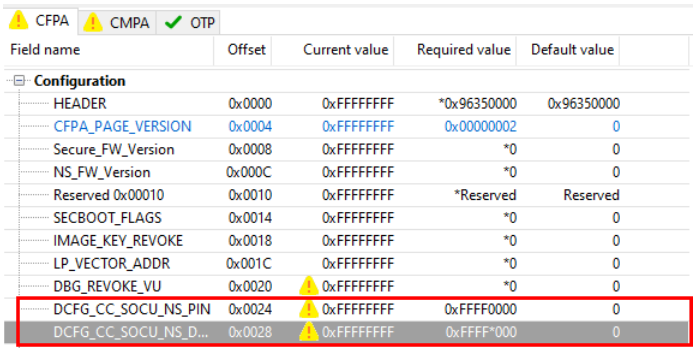

Figure 17. SOCU PIN/DFLT Nonsecure world settings

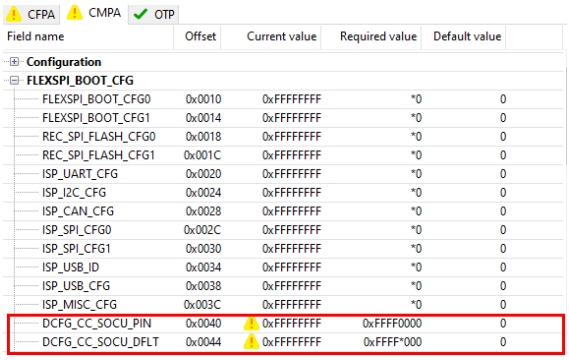

Figure 18. SOCU PIN/DFLT Secure world settings

Once the changes are completed, click **OK** to exit this window and build the image.

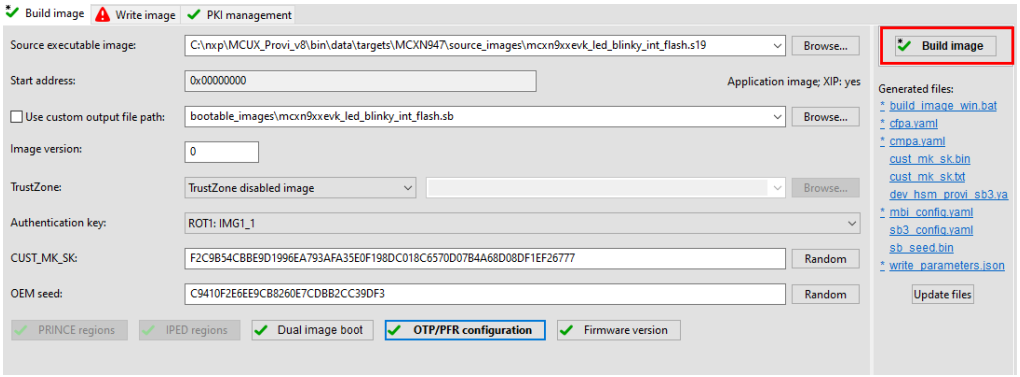

Figure 19. Build image

If the image is built successfully, a green message appears. Close the message and move on to writing the image to the device.

Figure 20. Successful image built

11 Write image

Once the image is built successfully, the option to write the image is enabled. In the bootable image options, there is the image name that was recently built in the previous tab. Click **Write image** to program the device with the signed image.

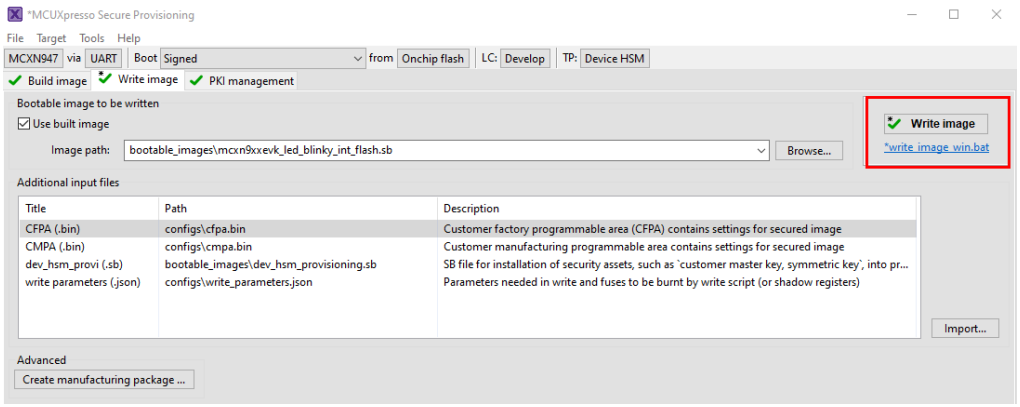

Figure 21. The Write image tab

After pressing reset, the LED blinky application boots up.

12 Device HSM

MCXN microcontrollers are equipped with the Device HSM trust provisioning feature. This allows the assets of the OEMs and their software IP to be transferred securely to production factories. Secure programming and provisioning can be achieved even under an unsecure manufacturing environment. A hardware security module (HSM) is typically a device used for securely managing, processing, and storing cryptographic keys inside a hardened, tamper-resistant hardware. NXP implements this concept at the device level so that the HSM capabilities are found in our microcontrollers with the purpose of managing secrets for OEMs. We call this trust provisioning solution "Device HSM".

In this application note, the two steps are broken down based on the example shown in the image below. Step 1 is creating the manufacturing package that can be done by the OEM design center. Step 2 is importing the manufacturing package that can be done at a third-party site. In this example, it is an unsecure location.

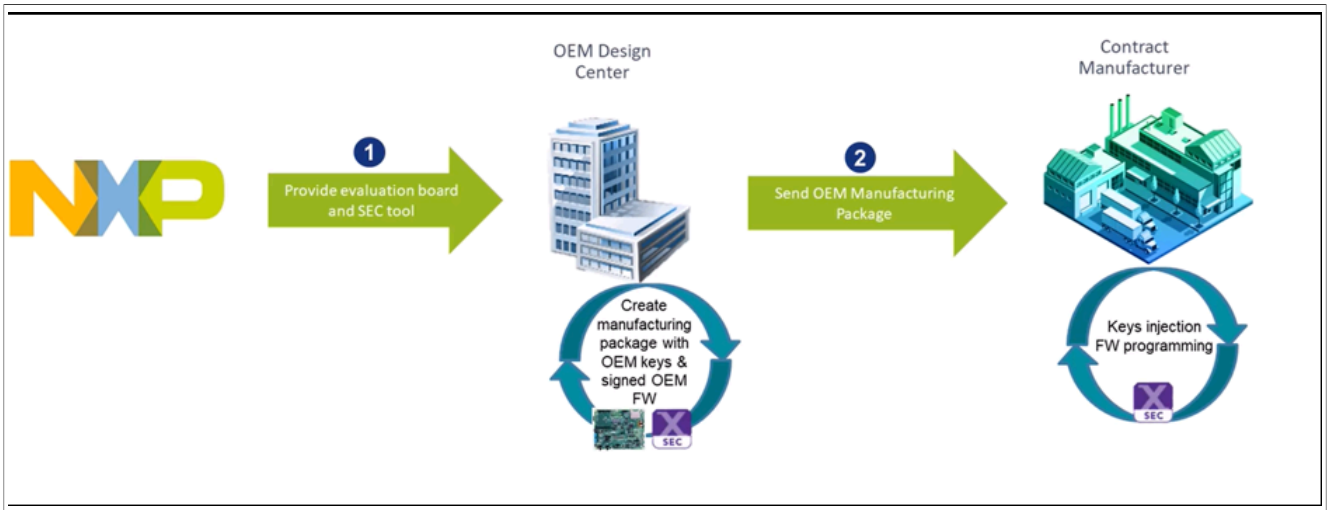

Figure 22. Device HSM flow

12.1 Creating a manufacturing package

The SEC tool is the only required tool to implement Device HSM. As mentioned in the previous section, selecting signed or encrypted images get automatically set the “Trust Provisioning” to Device HSM.

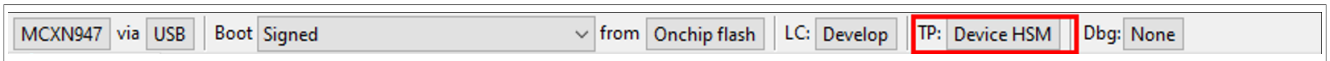

Figure 23. Trust Provisioning option

1. In addition to the secure boot steps described in this application, Device HSM must import restricted data for the specific device. The restricted data package can be downloaded from the SEC Tool product page. Check for version 8 or later that supports MCXN devices.

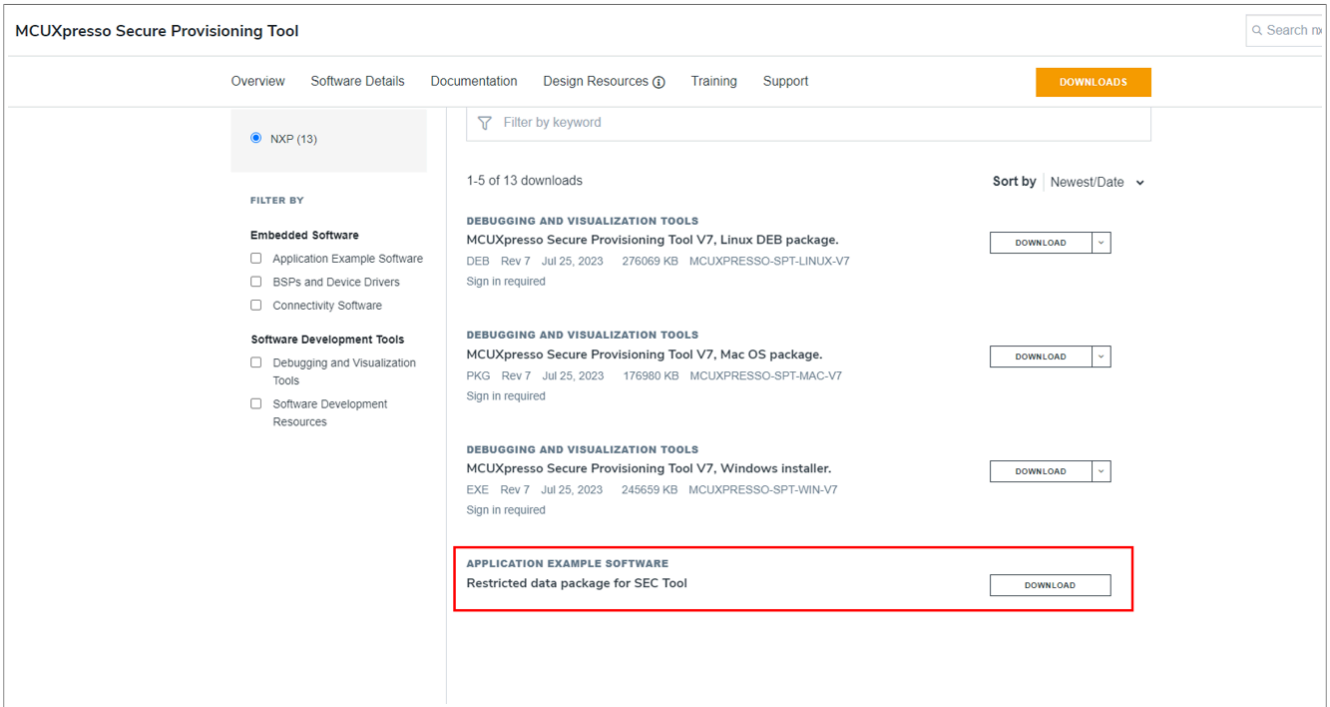

Figure 24. Restricted data package

2. To import the restricted data, select File > Preferences, this opens a new window, then click **Install restricted data.**

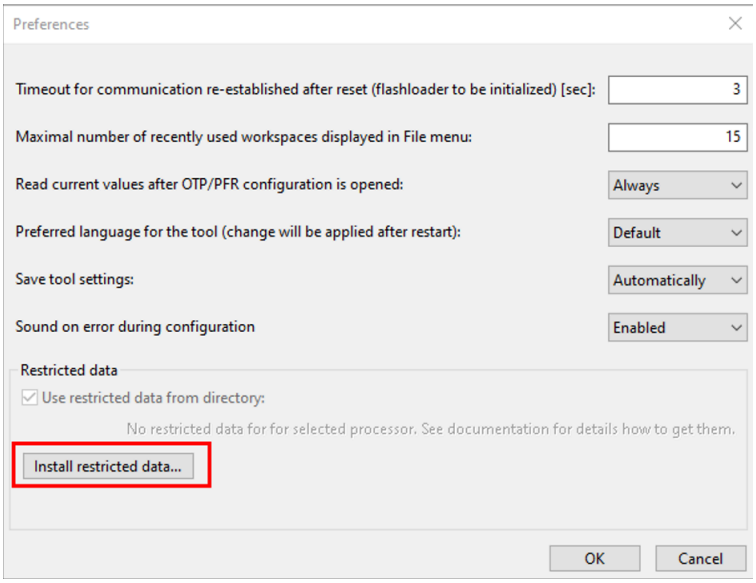

Figure 25. Import restricted data

3. Once the package has been installed correctly, exit the tool, restart the session, and open the workspace again.
4. Build the image.
5. Open the **Write** tab and select **Create manufacturing package…**
   Note: *The device must be connected to the host PC for the manufacturing package to be created successfully.*

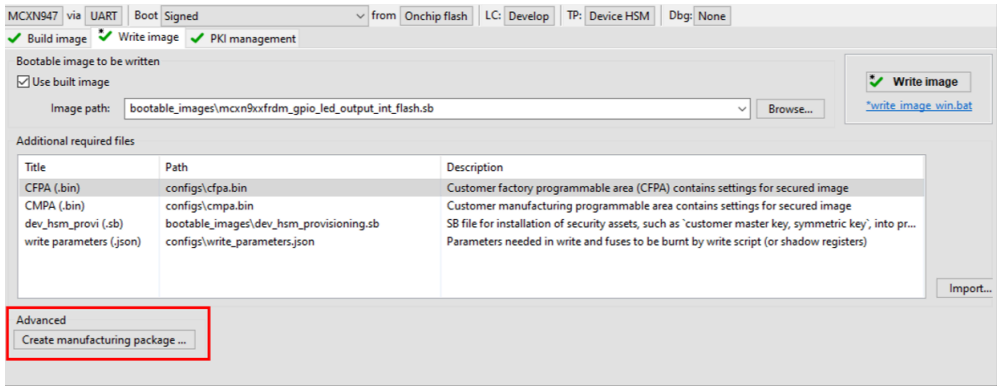

Figure 26. Create manufacturing package

6. For production devices, it is important to consider advancing the lifecycle to a secure field state. Use the **Lifecycle** button to permanently change the LC state from Develop to an Infield lifecycle state. For development purposes, it is possible to remain in the develop lifecycle as shown in this document.
7. The manufacturing package contains all the necessary files to execute the secure binary and the corresponding scripts. Click Ok, to generate the package at the output path.

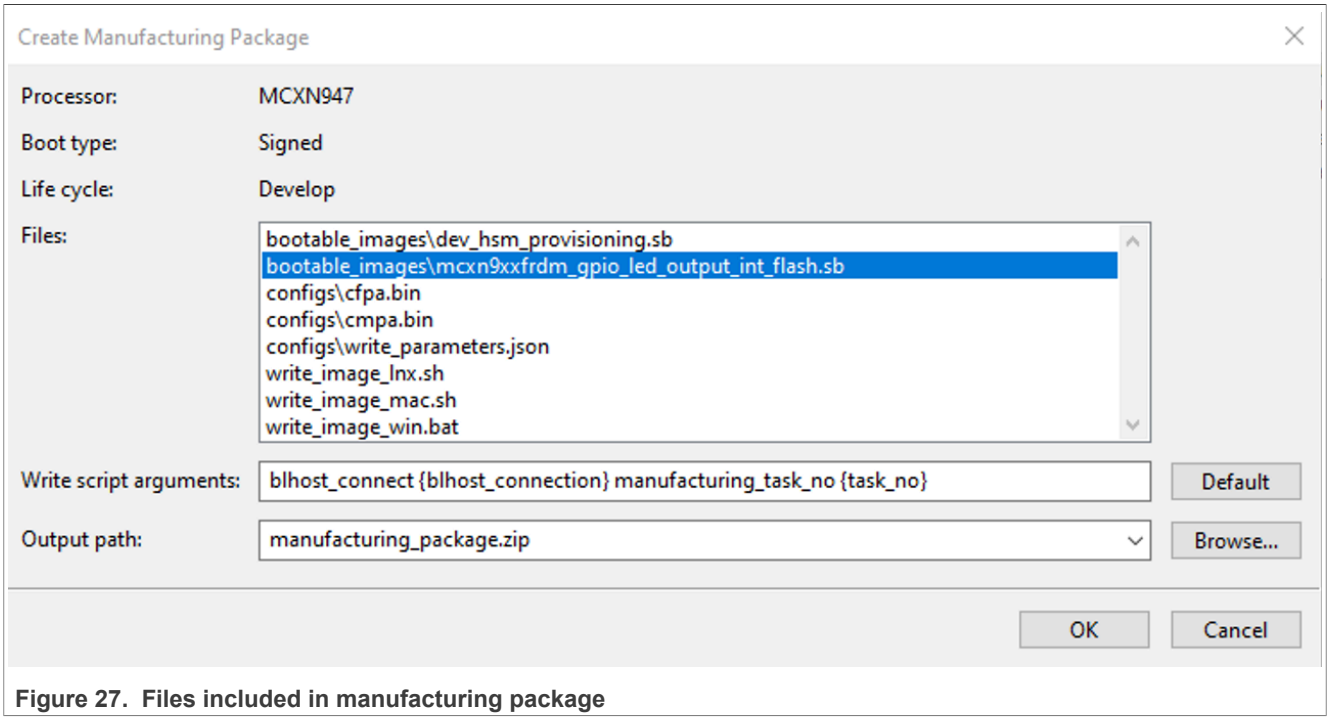

Figure 27. Files included in manufacturing package

12.2 Importing a manufacturing package

The SEC tool is the only required tool to import the manufacturing package. When opening the tool, instead of creating a workspace, import the manufacturing package by clicking the alternative option.

1. Select **Import manufacturing package** as shown below. At open workspace, select the option **from file.**

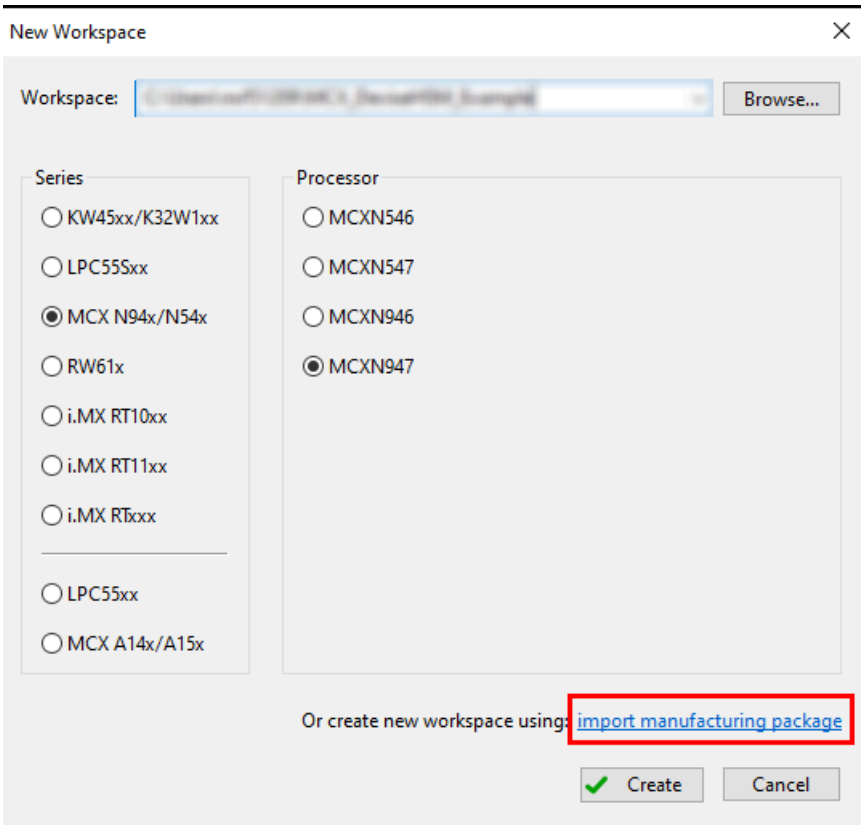

Figure 28. Create new workspace for manufacturing tool

2. Browse for the location of the package created in the previous step.

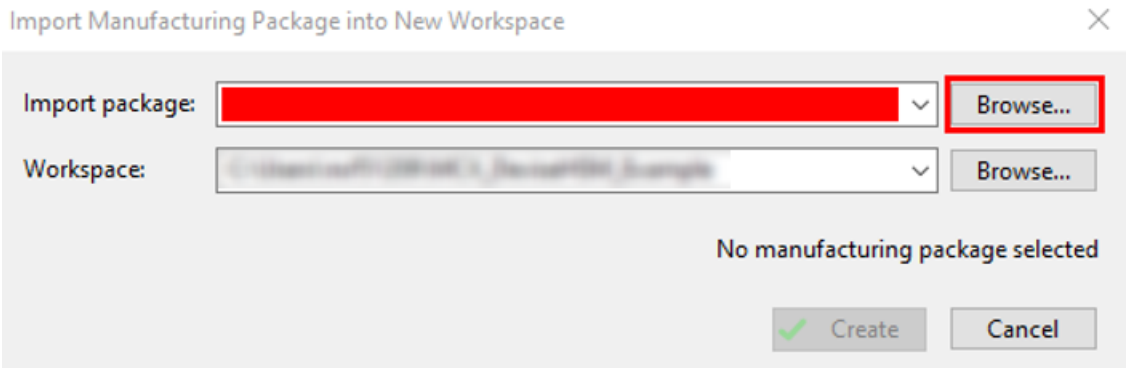

Figure 29. Browse path location for manufacturing package

3. Click **Create**
4. This opens the manufacturing tool.

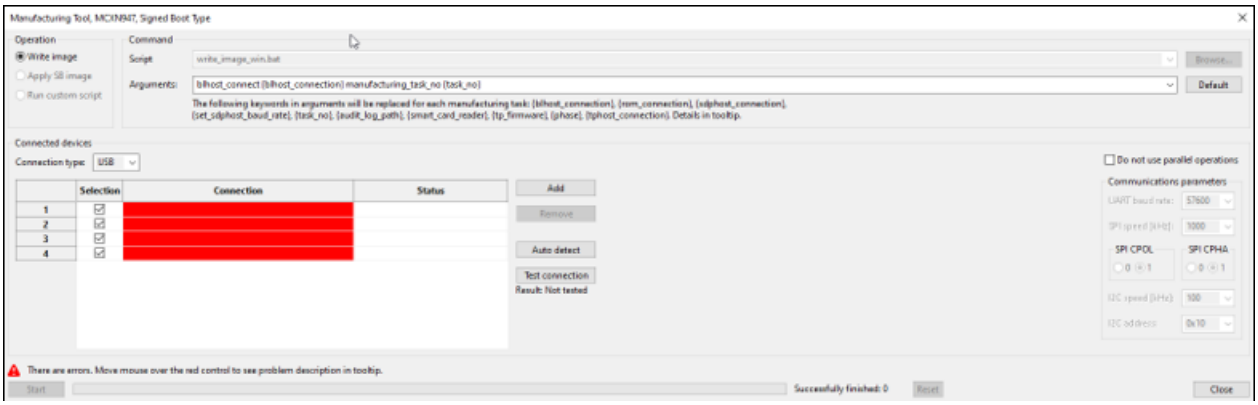

Figure 30. Manufacturing tool

5. The script provided in the manufacturing package is the only option to execute.
6. Select the connect type and then press autodetect. This helps to detect the COM port that is being used. Multiple connections are possible and can continue to add devices that must be programmed in the same manner.

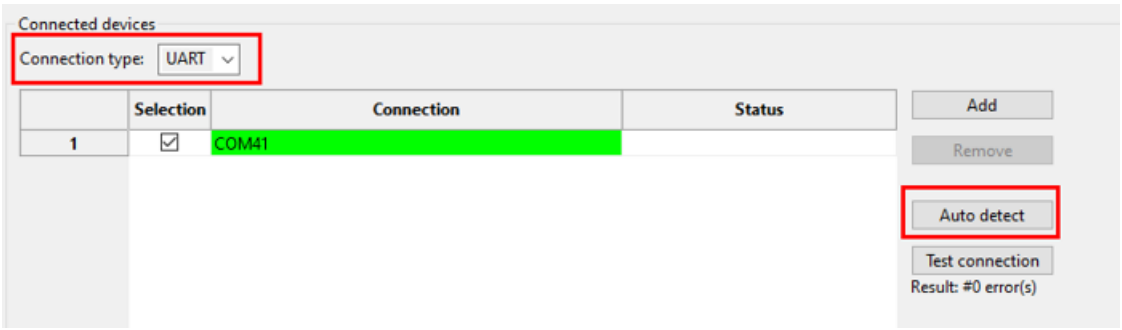

Figure 31. Test connection type

7. Click **Start** to run the script and load the secure binary to the device.

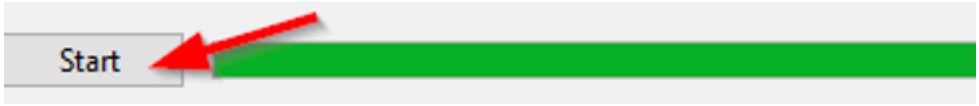

Figure 32. Start execution

At this point, your device can be reset and the application must be running. This manufacturing package can be reused for future purposes and redistributed without sharing essential assets to the application and device.

13 Revision history

Table 1. Revision history

Document ID	Release date	Description
AN14148 v.3.0	06 March 2025	Initial public release
AN14148 v.2.0	07 June 2024	Figure 28Figure 29 are updated.
AN14148 v.1.0	20 January 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