AN12328

Overview of supported methods for firmware flashing on NHS31xx ICs

Rev. 1.2 — 24 January 2022 Application note

Document information

Information	Content
Keywords	NHS31xx, Firmware, Flashing, Flash Magic, MCUXpresso, SWD, NFC
Abstract	Overview of supported methods for firmware flashing on NHS31xx ICs.

Revision history

Rev	Date	Description
v1.2	20220124	Update for SDK 12.5
v1.1	20200228	Update for SDK 12.3
v1.0	20190328	Update for SDK 12
v0.4	20180329	Update for SDK 11.1
v0.3	20170911	Update for SDK 11
v0.2	20170529	Changes after review
v0.1	20170229	Initial version

1 Overview

The different methods to program an NHS31xx IC are discussed.

All NHS31xx ICs support both wired and wireless flashing to store the firmware in the non-volatile FLASH memory.

- Wired flashing uses the JTAG standard over the 2-pin electrical interface SWD.
- Wireless flashing uses the NDEF protocol over the NFC interface.

2 Preflashed

NXP offers the possibility to preflash NHS31xx ICs during production. This feature eliminates the need for wired or wireless flashing during solution assembly altogether.

The conditions to meet and the procedure to use this offer are explained in the application note "NHS31xx customer firmware flashing" (Ref. 1) available in the SDK under the docs folder.

Note: *The custom application program provided by the user cannot be overwritten. The first sector of the flash is locked after writing the custom application program. The wired and wireless flashing options as described in the chapters below are not applicable on preflashed ICs.*

3 Wired

Different tooling for wired flashing is available:

- Most commonly used while developing and debugging, is the built-in download feature in the MCUXpresso IDE. The suite connects to the target (NHS31xx) via SWD (wired) using an LPC-Link2 debug board.
- Flash Magic is an independent tool which can be used in a production environment. It only implements the minimal SW parts required to program a device. It is a tool developed and supported by Embedded Systems Academy. This tool also uses the LPC-Link2 debug board to communicate with an NHS31xx IC.
- J-Link Software can be used during development. A J-Link debug probe can interact with all NHS31xx ICs out of the box. J-Link Software can also be used in combination with an LPC-Link2 debug board. The debug board then needs to be flashed using LPCScrypt with J-Link firmware. See the "LPC-Link 2 Firmware - Getting Started" support page under "J-Link Debug Probes" on the segger.com website, and the document `UM08001` "J-Link / J-Trace User Guide", from Segger.
- A last option is to write a custom host application, using IAP commands over the SWD debug interface. An SWD programmer can use the debug interface of the chip to program the on-chip FLASH memory directly. The full specification and detailed information on the SWD protocol can be found in document `IHI0031A` "ARM Debug Interface v5 - Architecture Specification", from Arm.

The options for using the MCUXpresso IDE and Flash Magic are described in more detail below. The last option is outside the scope of this document and NXP Semiconductors provides no support.

3.1 MCUXpresso

The MCUXpresso IDE v10.2.1 is the supported IDE for developing with NHS31xx ICs.

3.1.1 Installation and setup

Installation and setup of the environment is described in the user manual "NTAG SmartSensor getting started: A guide to start developing using an NHS31xx" (Ref. 2), which can be found in the SDK, under the docs folder.

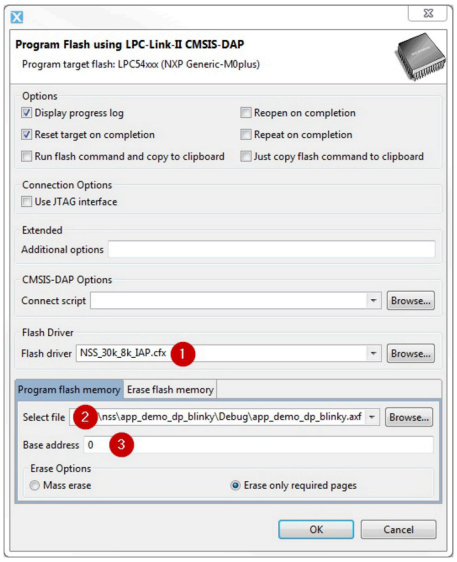
3.1.2 Usage – GUI

By default, starting a debug session automatically programs the flash. But you can also flash any `.axf` file (or `.elf` file) and `.bin` file without starting a debug session.

aaa-032829

Figure 1. Program flash

- Within the IDE, select a compatible project. The program flash icon only becomes accessible after selecting a project, since some of the project settings are implicitly reused. Select a project that reuses the same MCU settings as the `.axf` or `.bin` file you want to flash.
- Click the Program flash icon in the toolbar (see Figure 1).
- In the dialog that pops up, verify these settings (see Figure 2):
  1. **Flash driver**: `NSS_30k_8k_IAP.cfx` This file was copied to the MCUXpresso installation directory under *<install path>/Ide/bin/Flash* during the installation of the NHS31xx plugin. If the selected project matches your MCU, it is already correctly filled in here.
  2. **Select file**: The `.axf` or `.bin` application file to flash.
  3. **Base address**: `0`



aaa-032830

Figure 2. Program flash dialog

After clicking OK, the flash is programmed. At the end, a dialog pops up displaying the log and the result.

3.1.3 Usage – command line

Using the option "Just copy the flash command to clipboard" in the dialog above, the correct command-line usage can be readily retrieved.

More details about the different command-line options and their arguments can be found at nxp.com: https://community.nxp.com/thread/389139

Example

```
crt_emu_cm_redlink.exe -flash-load-exec "C:\path\to
\app_demo_dp_blinky.axf" -g -2 -vendor=NXP -pGeneric-M0plus -
load-base=0 -reset=vectreset -flash-driver=NSS_30k_8k_IAP.cfx -x
C:/path/to/application/projectfolder
```

This single-line command flashes the given `.axf` file.

Notes

- The command-line option can only be used on a PC where an MCUXpresso installation has been activated.
- `crt_emu_cm_redlink.exe` can be found in the MCUXpresso installation folder, under `ide\bin`.
- A path to a folder which contains these files can replace the path to the project folder (option `-x`):
  - `CM0_peripheral.xme`
  - `crt_common.xme`
  - `Generic-M0plus.xml`
  - `Generic-M0plus_part.xml`

3.2 Flash Magic

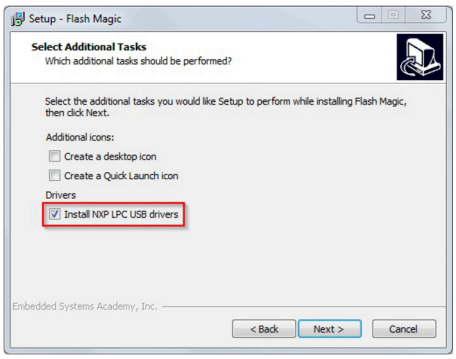
Flash Magic is a third-party PC tool for programming Flash-based microcontrollers from NXP Semiconductors via a serial protocol using Intel HEX files. It can be used freely during development or for programming small batches. Using the tool on a production line is also possible, but requires a purchase. More information is available at the dedicated webpages of the Flash Magic tool.

The use of this tool is not enforced, but helps to program ICs that use pre-built firmware images quickly. It may at times help with recovering ICs which have become inaccessible due to a bug in the SW. For details, check the documentation in the SDK: "SW debug considerations" in *<SDK>/docs/firmware.html*.

Note: *Only versions from v9.72 onward support the NHS31xx ICs. Until Flash Magic is updated on other platforms, only the Windows platform is supported.* Figure 7 *is taken from Flash Magic v11.16.*

3.2.1 Installation

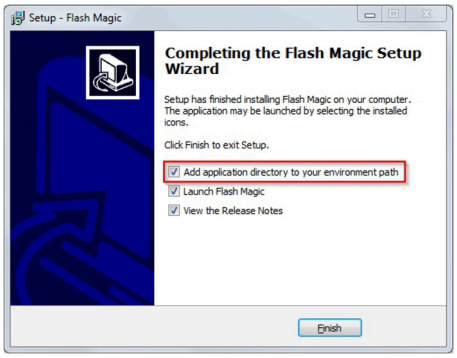
- Download Flash Magic. A direct download link to a recent version, known to work correctly, can be found in the SDK under in the *<SDK>/tools/flashmagic*.
- Install. When prompted during installation:
  1. Install the LPC USB drivers of NXP Semiconductors.



aaa-032831

Figure 3. The option to install extra drivers can be selected just before the actual installation starts

  2. Add the application directory to your application path



aaa-032832

Figure 4. Updating the environment path simplifies later command-line use

After installation, Flash Magic is ready to be launched and used.

aaa-032833

Figure 5. Flash Magic as it appears in the Windows Start menu

3.2.2 Physical setup

The physical setup requires an LPC-Link2 board, which is shipped together with the NHS31xx development boards in the various kits offered by NXP Semiconductors.

1. Remove `JP1` from the LPC-Link2 board to soft-load the debugger firmware on the LPC-Link2 board. See Figure 6. With the jumper fitted, debugger firmware must have been flashed into the NVM using the LPCScrypt tool. This jumper is only looked at at startup when the LPC-Link2 board is powered.
2. If necessary (that is, when no battery is connected), make sure `JP2` is fitted.
3. Connect the LPC-Link2 board to he PC using a mini-USB cable.
4. Connect the demo PCB to the LPC-Link2 board using a JTAG cable.

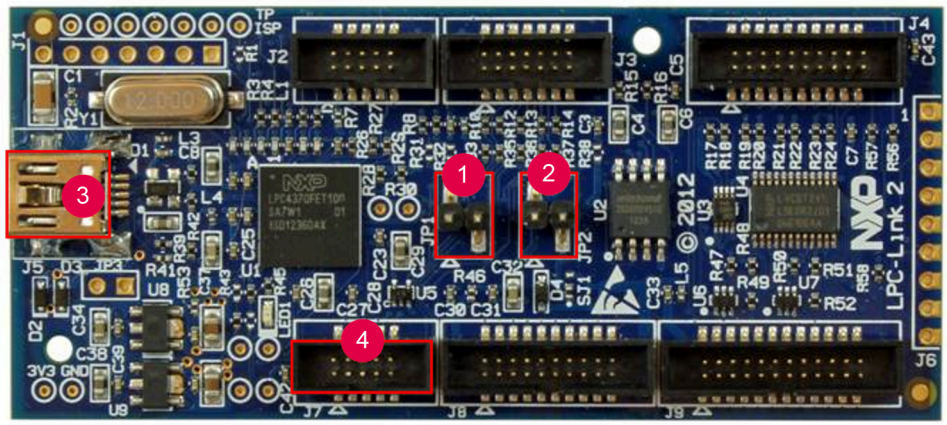


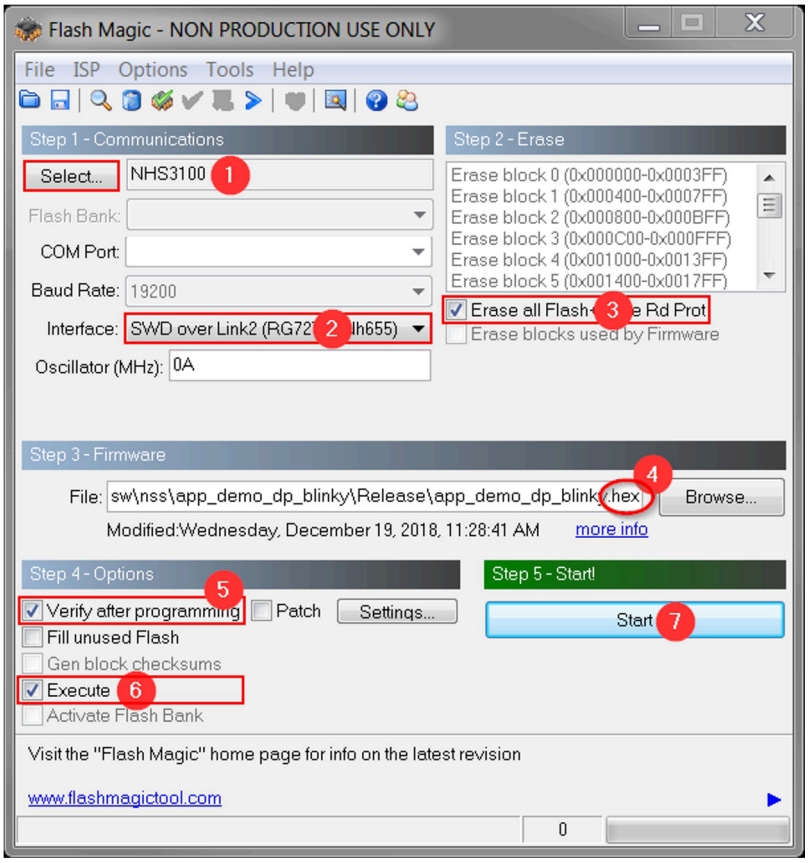
Figure 6. Physical interface of the LPC-Link 2

If Flash Magic cannot find your LPC-Link2 board, check your setup, power cycle the board, and restart Flash Magic

3.2.3 Usage – GUI

The GUI is best suited for flashing one or more samples during the development phase or to prepare for demonstrations.

First connect one or more LPC-Link2 boards to the PC, then launch the Flash Magic GUI. The recommended settings to use are shown in Figure 7.



aaa-032835

Figure 7. Recommended options within the Flash Magic tool

1. Select the correct target: `NHS3100` or `NHS3152`
2. Use `SWD over Link2` as interface. If no LPC-Link2 board was connected during start-up of Flash Magic or if the LPC-Link2 board was running the CMSIS-DAP protocol, this option is not displayed. In that case, connect a debugger board, or power-cycle the connected debugger board, and restart Flash Magic.
3. The safest option is to erase all Flash sectors. This action also erases all sectors that may still contain (part of) the one-time NFC program downloader. It also ensures that the firmware does not have to perform this costly operation itself.
4. Select the desired `.hex` file to Flash.
5. Optionally, tick the checkbox next to "Verify after programming".
6. By ticking the checkbox next to "Execute", Flash Magic ensures that the chip immediately starts executing the newly programmed application.
   With this option turned off, the IC remains in a halted state, waiting for an external RESETN trigger. It usually amounts to the user requiring to press the reset button on the demo PCB.

7. Finally click "Start" to carry out the requested operations.

Note: *When the error "Unable to open Hex file" is given, check the path of your selected firmware image file. Try to reduce the length of the path and ensure that the full path only contains ASCII characters.*

Note: *When the Arm core cannot be halted, Flash Magic alerts you with an unrelated error message (see Figure 8).*

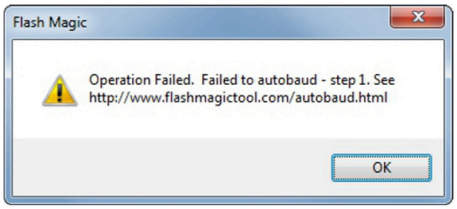


Figure 8. Pop-up error dialog when the Arm core could not be accessed

Possible causes are:

- The IC is not attached or not powered at all.
- The IC has entered a low-power state, deep power-down state, or power-off state, where the SWD pins are no longer active.
- The firmware image actively disables SWD access.

3.2.4 Usage – command line

Flash Magic also provides extensive command-line support via a separate executable `FM.EXE`. A full description of the supported commands and arguments can be found in the manual of Flash Magic. This file, Manual.pdf, is present in the installation folder of Flash Magic and can also be opened via the GUI: Help > Manual.

Example:

```
FM.EXE INTERFACE(SWDLINK2) DEVICE(NHS3100, 0.000000, 0)
ERASE(DEVICE, PROTECTISP) HEXFILE(app_demo_dp_blinky.hex,
NOCHECKSUMS, NOFILL, PROTECTISP) VERIFY(app_demo_dp_blinky.hex,
NOCHECKSUMS) RESET
```

This single-line command:

1. Connects to a NHS3100 via an LPC-Link2 board.
2. When connected, erases the complete flash first.
3. Programs the contents of the given hex file.
4. Verifies the sections occupied by the new binary against the same hex file.
5. Resets the target such that the newly flashed firmware becomes active.

3.2.5 Usage – gang programming

Flash Magic has also gained support (since v10.40) for flashing multiple targets at once, using the command line. This process is called gang programming or mass programming. It is the best option to quickly flash many targets in a production environment. Multiple Windows batch files, each containing command-line text as shown in Section 3.2.4, can be run to perform multiple flashing operations simultaneously.

3.2.5.1 Physical setup

Connect a number of LPC-Link2 boards to a programming PC. To connect more debugger boards to the same PC, you can use one or more external USB hubs. Be sure to use a self-powered hub that guarantees a steady supply voltage for each port that is in use.

Each LPC-Link2 board can then program one NHS31xx IC in parallel.

3.2.5.2 Targeting a specific LPC-Link2 board

To enable gang programming, the unique serial number of the LPC-Link2 debug boards must be used. Flash Magic supplies the tool `USBManager.exe` which can be used to retrieve these serial numbers. `USBManager.exe` in turn relies on the presence of a few DLLs and other files in the installation folder of Flash Magic.

The interface serial numbers of all connected LPC-Link2 boards can then be obtained from the command line using:

```
USBManager.exe --seriallist --nobanner
```

3.2.5.3 Batch file example

The SDK provides an example batch file which demonstrates how gang programming can be implemented on a programming PC: `<SDK>/tools/flashmagic/gangprogramming.bat`

To retrieve the usage instructions, use `/?` or `-h` or `-help` as command-line argument.

Internally, the batch file is fully documented which helps you to tailor it completely to your mass-programming requirements.

Now, a fully automatic mass-production session can be started with this simple call:

```
gangprogramming.bat C:\path\to\applicationfirmware.hex
```

The batch file performs four tasks:

- If no user input is provided on the command line, gather this input. If the application firmware image is supplied as command-line argument, the program can run automatically.
- Generate temporary batch files with the correct flashing instructions using the command-line support of Flash Magic, one for each connected LPC-Link2 board.
- Start the temporary batch files and assemble the different logs generated by the flashing processes.
- Summarize and display the result and exit:
  - Number of programming operations completed
  - Number of failed attempts
  - Path of the firmware image that was used for flashing
  - Interface serial numbers of the LPC-Link2 boards used

4 Wireless

All NHS31xx ICs are flashed during production with a “second stage boot loader”, called the NHS31xx NFC program downloader, which offers the ability to program an NHS31xx target once over the NFC interface. This option allows for late programming outside a production environment, even after all sealing and packaging has been completed.

Note: *Your application firmware can also contain an update module which replicates the functionality of the “second stage boot loader”. This option is not available from the SDK and must be implemented by the customers themselves.*

This section gives a high-level overview of the working of the NFC loader and how to use the host side offering, available in the SDK.

4.1 Target: NHS31xx

Wireless flashing requires a low-ohmic external supply connected to VBAT. Flashing is not possible on VNFC supply only.

The start condition is an NHS31xx IC which is physically connected to an external battery and an NFC antenna. The IC is powered off, that is, with its disconnect circuitry in the open state.

Whenever the NFC antenna detects an NFC field and is powered, the IC automatically wakes up and prepares an initial message in the NFC shared memory. The initial message contains program and version information.

After the host has read the tag and matched the content with its expectation (an NDEF formatted MIME record containing a correct version response), it starts sending commands carrying parts of the binary of the firmware image to program. After each sent command, the IC generates one response acknowledging the command, overwriting the command payload bytes in the NFC shared memory in the process. A list of all possible responses can be found below.

After creation of the response, the IC starts a 5-second timeout. If the host fails to send a new command within these 5 seconds, the IC switches off. At the end of the download, the target also enters the power-off state to preserve the battery.

Figure 9 describes the operation flow of the NFC program downloader.

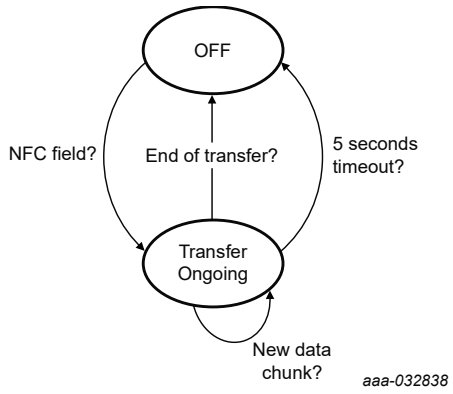


Figure 9. NFC program downloader state machine diagram

4.2 Host: Android

NXP Semiconductors has released an Android app which acts as the host side for firmware flashing via the NFC interface. This app can be found in the Google Play Store

The app does not contain any demo itself. When starting the app, you must point the app to a demo container or a single binary file. A demo container groups multiple demos together and allows for an easy selection of the correct demo to flash. The NXP demos from the latest released SDKs are available at https://www.nxp.com/NHS31XX_DEMOS.

You can also create your own container with your selection of demos and use that with the app. Information on how to do it is described inside the container offered by NXP.

4.3 Host: Win1x GUI

From v12.5 onward, a Windows application is also available inside the SDK. Installation instructions are available under *<SDK>/sw/XF/NHS31xxDownloader.html*. To use it, you also need an external USB NFC reader plugged into your computer. Operation and use are very similar to the Android version.

4.4 Host: Win1x CLI

A Windows Python script can also be used for command-line wireless flashing. It is present in the SDK under *<SDK>/tools/nfcloader/python*. This tool has the same core functionality as the GUI apps. It can read a given binary file and write it (chunked) to the target using a USB-connected NFC reader.

The script provides its own usage instructions and operating overview. See also the accompanying file *README.pdf* in the SDK.

4.5 Benefits and drawbacks

- The use of the contactless NFC interface for flashing the firmware gives more flexibility in the production process and a delayed finalization of the firmware.
- Since no physical wired connection to the programming pins (SWD) is required, a solution based on an NHS31xx IC can be fully constructed without taking into account firmware. The layout can be simplified even more and the label can be fully laminated. At a later stage, the firmware can still be programmed by using the NFC interface.
- The greatest downside with the ‘over the air’ firmware flashing is the significantly reduced transfer speed compared to a wired connected solution. It is possible to parallelize the flashing operation by having multiple ‘flash’ stations.
- The power required to perform one or more actual flash operations is pulled from the VDDBAT line. Do not attempt wireless flashing using a passive setup. An NHS31xx IC cannot be flashed reliably on NFC power only.
  To enable wireless flashing, ensure that the low-ohmic power source is connected to the VDDBAT line.

5 References

[1] **AN12251 application note** — NHS31xx customer firmware flashing; 2021, NXP Semiconductors

[2] **UM11153 user manual** — NTAG SmartSensor getting started: A guide to start developing using an NHS31xx; 2021, NXP Semiconductors

6 Legal information

6.1 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.

6.2 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 http://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 data sheet 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.

Translations — A non-English (translated) version of a 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.

6.3 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

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 24 January 2022
Document identifier: AN12328