AN12328

Overview of supported methods for firmware flashing on NHS31xx ICs

Rev. 1.1 — 28 February 2020 Application note

Document information

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

Revision history

Rev	Date	Description
v1.1	20200228	Update for SDK 12.3
Modifications:	• Text and graphics have changed throughout the document • Reference to AN12251 added in Section 2.	
v1.0	20190328	Update for SDK 12
Modifications:	Major format update and refresh of contents	
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 W8 (bumped die with 8 functional bumps) (NHS3100) 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 AN12251 (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 wafers.*

3 Wired

Different tooling for wired flashing is available:

- Most commonly used while developing and debugging, is the built-in download feature in the LPCXpresso 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 a third party: see http://www.flashmagictool.com/ for more info and contact details. This tool also uses the LPC-Link2 debug board to communicate with an NHS31xx IC.
- 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", created and maintained by ARM. See http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0031a/index.html.

The options for using the LPCXpresso 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 LPCXpresso

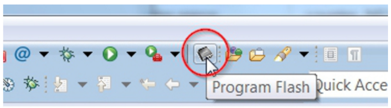
The LPCXpresso IDE v8.2.2 is the supported IDE for developing with NHS31xx ICs.

3.1.1 Installation and setup

Installation and setup of the environment is described in the user manual UM11153 (Ref. 2), which can be found in the SDK, under the docs folder.

3.1.2 Usage – GUI

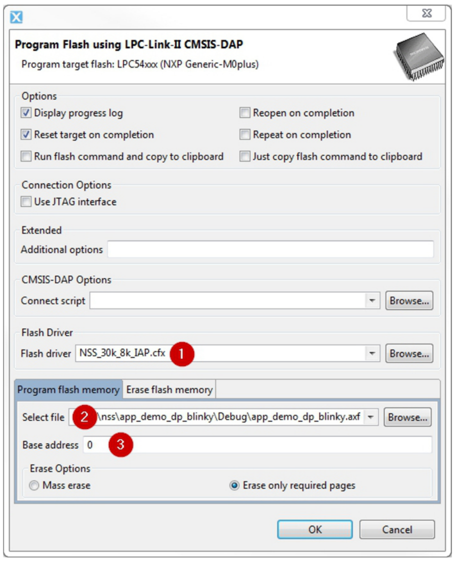
By default, starting a debug session automatically programs the flash. But you can also flash any `.axf` file (or `.elf` file) and `.bin` file without starting a debug session.



aaa-032829

Figure 1. Program flash

- Within the LPCXpresso 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 LPCXpresso installation directory under *<install path>/lpcxpresso/bin/Flash* during the installation of the NHS31xx plugin, and is already correctly filled in here if the selected project matches your MCU.
  2. **Select file**: the `.axf` or `.bin` application file to flash.
  3. **Base address**: `0`



Program Flash using LPC-Link-II CMSIS-DAP

Program target flash: LPC54xxx (NXP Generic-M0plus)

Options

☑ Display progress log
☐ Reopen on completion
☑ Reset target on completion
☐ Repeat on completion
☐ Run flash command and copy to clipboard
☐ Just copy flash command to clipboard

Connection Options

☐ Use JTAG interface

Extended

Additional options

CMSIS-DAP Options

Connect script

Flash Driver

Flash driver NSS_30k_8k_IAP.cfx ①

Program flash memory | Erase flash memory

Select file ② \nss\app_demo_dp_blinky\Debug\app_demo_dp_blinky.axf

Base address 0 ③

Erase Options

○ Mass erase
● Erase only required pages

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 LPCXpresso installation has been activated.
- `crt_emu_cm_redlink.exe` can be found in the LPCXpresso installation folder, under `lpcxpresso\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

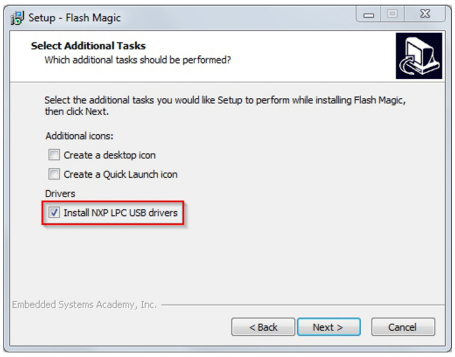
Flash Magic is a PC tool for programming Flash-based microcontrollers from NXP Semiconductors via a serial protocol using Intel HEX files. It can be used freely during development or for programming small batches. Using the tool on a production line is also possible, but requires a purchase. More information is available at http://www.flashmagictool.com/productionsystem.html

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.



Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Flash Magic, then click Next.

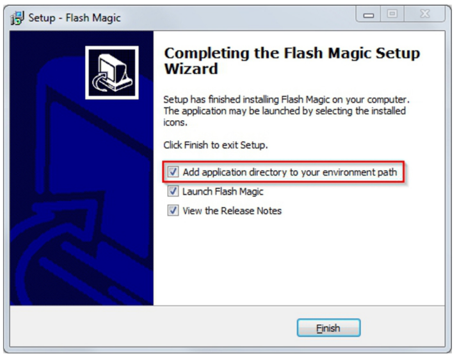
Additional icons:
☐ Create a desktop icon
☐ Create a Quick Launch icon

Drivers
☑ Install NXP LPC USB drivers

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



Completing the Flash Magic Setup Wizard

Setup has finished installing Flash Magic on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Add application directory to your environment path
☑ Launch Flash Magic
☑ View the Release Notes

aaa-032832

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

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 (see Figure 6).
2. If necessary (i.e. when no battery is connected), make sure `JP2` is present.
3. Connect the demo PCB with the LPC-Link2 board and the LPC-Link2 board with the PC.

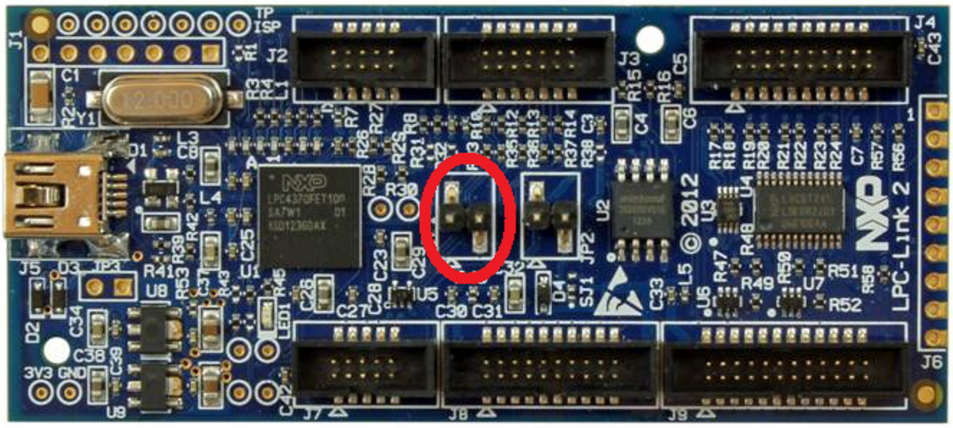
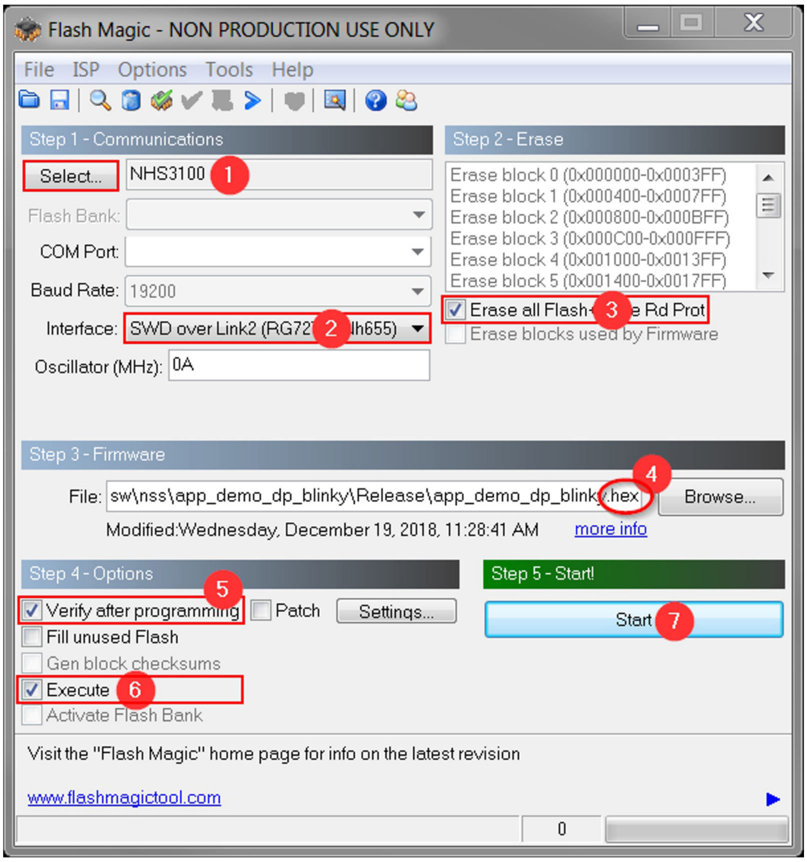


Figure 6. Highlighting the position of JP1, JP2 next to it

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.



Flash Magic - NON PRODUCTION USE ONLY

Step 1 - Communications

Select... NHS3100

Flash Bank:

COM Port:

Baud Rate: 19200

Interface: SWD over Link2 (RG72 lh655)

Oscillator (MHz): 0A

Step 2 - Erase

Erase block 0 (0x000000-0x0003FF)
Erase block 1 (0x000400-0x0007FF)
Erase block 2 (0x000800-0x000BFF)
Erase block 3 (0x000C00-0x000FFF)
Erase block 4 (0x001000-0x0013FF)
Erase block 5 (0x001400-0x0017FF)

☑ Erase all Flash+ e Rd Prot
☐ Erase blocks used by Firmware

Step 3 - Firmware

File: sw\nss\app_demo_dp_blinky\Release\app_demo_dp_blinky.hex

Modified:Wednesday, December 19, 2018, 11:28:41 AM more info

Step 4 - Options

☑ Verify after programming ☐ Patch
☐ Fill unused Flash
☐ Gen block checksums
☑ Execute
☐ Activate Flash Bank

Step 5 - Start!

Visit the "Flash Magic" home page for info on the latest revision

www.flashmagictool.com

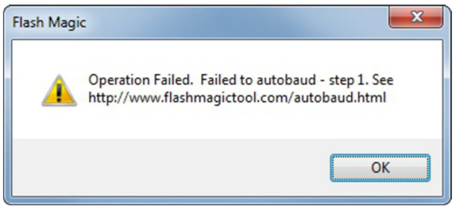
0

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 and it 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 ARM core cannot be halted, Flash Magic alerts you with an unrelated error message (see Figure 8).



Operation Failed. Failed to autobaud - step 1. See http://www.flashmagictool.com/autobaud.html

aaa-032837

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

The start condition is an NHS31xx IC which is physically connected to an external battery and an NFC antenna. The IC is powered off, i.e. 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 goes to the power-off state to preserve the battery.

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

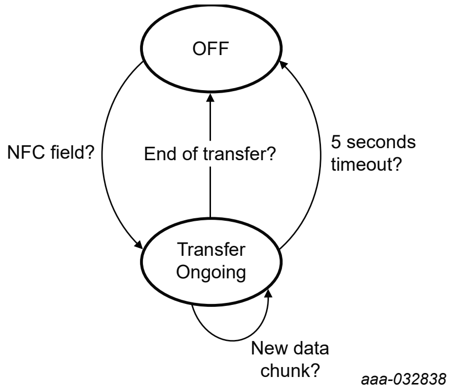


Figure 9. NFC program downloader state machine diagram

4.2 Host: Python

A Python script, which acts as the host side for firmware flashing via the NFC interface, is present in the SDK under *<SDK>/tools/nfcloader/python*. This tool reads a given binary file and loads it (chunked) into the target using a USB connected NFC reader/writer.

The script provides its own usage instructions and operating overview (see the accompanying file README.pdf in the SDK).

4.3 Host: Android

NXP Semiconductors has also released an Android app with the same functionality as the Python implementation. This app is publicly available in the Google Play Store. Its installer file is included in the SDK under *<SDK>/sw/android/dwn*.

Instructions on how to use this APP can be found in the user manual UM11136 (Ref. 3), which can be found on the DOCUMENTATION tabs of the different NHS31xx product pages, or on the Get Started pages of each development kit, for example, here: nxp.com/pages/:GS-NHS3100TEMOADK.

4.4 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.

5 References

[1]	**AN12251 application note**	— NHS3100W8 customer firmware flashing; 2018, NXP Semiconductors
[2]	**UM11153 user manual**	— NTAG SmartSensor getting started: A guide to start developing using an NHS31xx; 2019, NXP Semiconductors
[3]	**UM11136 user manual**	— NTAG SmartSensor getting started: Using the one-time NFC program downloader; 2019, NXP Semiconductors.

6 Legal information

6.1 Definitions

Draft — The document is a draft version only. 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 herein 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.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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 — While NXP Semiconductors has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP Semiconductors accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

6.3 Trademarks

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

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: 28 February 2020
Document identifier: AN12328