AN12991

PN7160 evaluation kit quick start guide

Rev. 1.2 — 5 September 2025

Application note

Document information

Information	Content
Keywords	OM27160, PN7160, Raspberry Pi, Arduino, eval kit, NFC, P2P, Card Emulation, Linux, Android, RTOS, Null OS
Abstract	This document gives a description on how to get started with the PN7160 evaluation kit.

1 Introduction

This document gives a description on how to get started with the PN7160 evaluation kit, referenced as OM27160A1EVK for the I^2C host interface version (featuring PN7160A1HN I^2C variant) and OM27160B1EVK (featuring PN7160B1HN SPI variant).

It provides a step-by-step guide to the installation procedure of the hardware and the software.

Finally, it shows PN7160 NFC controller functionalities through demo applications.

1.1 PN7160 evaluation kit description

The PN7160 evaluation kit exists in two versions:

- OM27160A1EVK: embeds PN7161A1HN I^2C variant in HVQFN40 package
- OM27160B1EVK: embeds PN7161B1HN SPI variant in HVQFN40 package

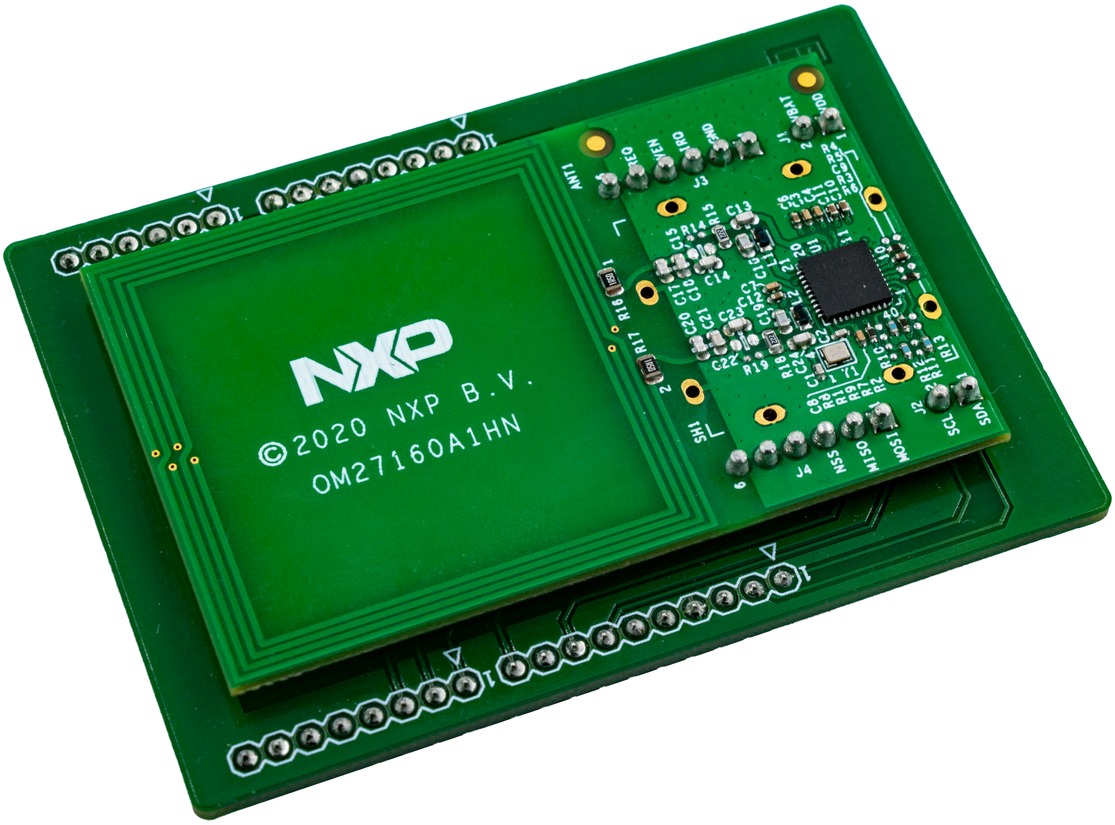
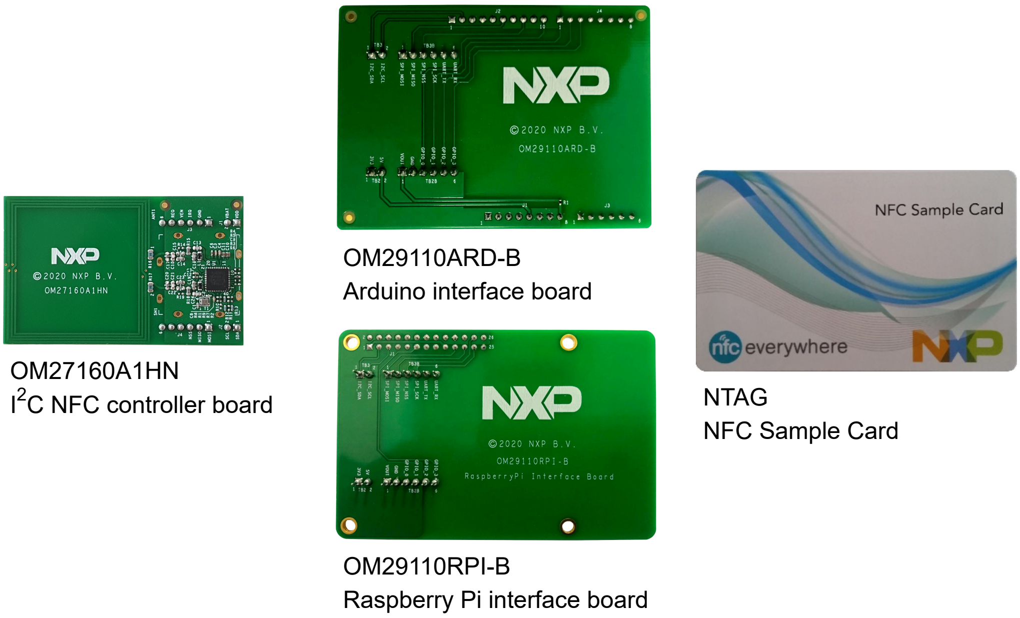


Figure 1. PN7160 evaluation kit

Both versions are composed of three printed-circuit boards and an NTAG card.



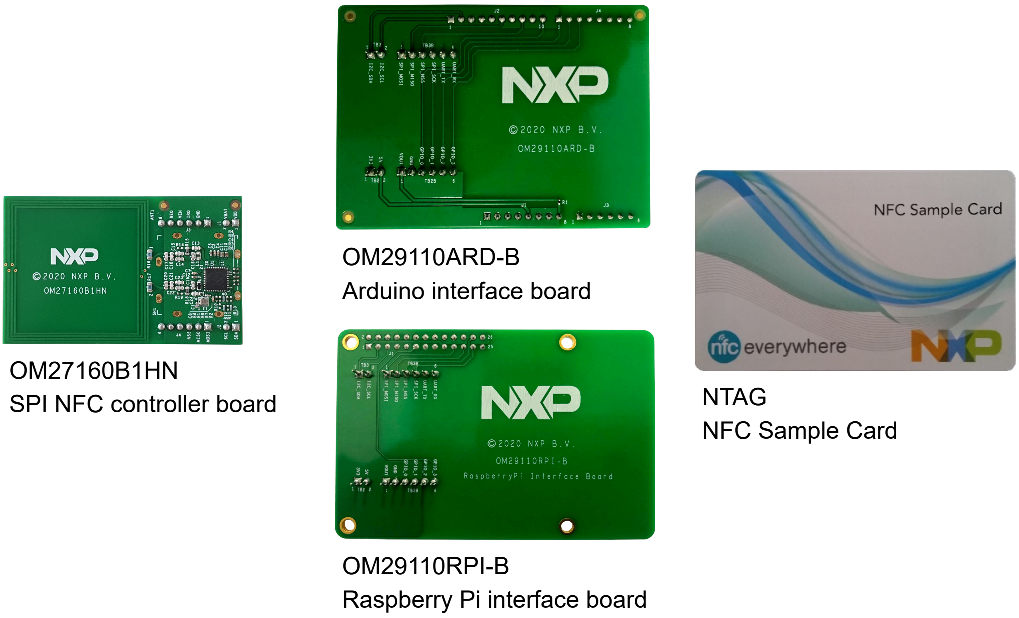
OM27160A1HN
I^2C NFC controller board

OM29110ARD-B
Arduino interface board

OM29110RPI-B
Raspberry Pi interface board

NTAG
NFC Sample Card

Figure 2. OM27160A1EVK I^2C dev kit overview



OM27160B1HN
SPI NFC controller board

OM29110ARD-B
Arduino interface board

OM29110RPI-B
Raspberry Pi interface board

NTAG
NFC Sample Card

Figure 3. OM27160B1EVK SPI dev kit overview

1.2 Linux driver support

PN7160 NFC controller is supported under GNU/Linux systems using the NXP Linux libnfc-nci software stack (see ref.[2]).

1.3 Android driver support

The PN7160 NFC controller is supported from the official Android Open Source Project (refer to ref.[3] for more details) with the addition of dedicated patches (see ref.[4]).

1.4 RTOS and Null OS support

Since implementing NFC Forum NCI 2.0 standardized API, the PN7160 NFC controller can be easily integrated into system based on RTOS or even without OS.

Code examples are given in the form of MCUXpresso projects (dedicated to specific NXP's MCUs) and can easily be ported to any other system (refer to ref.[5]).

2 Quick startup with Raspberry Pi interface board

2.1 Required items

- Raspberry Pi ref.[1] running Raspberry Pi OS.

2.2 Hardware setup

Assemble the PN7160 NFC controller board (OM27160A1HN or OM27160B1HN) with the Raspberry Pi interface board (OM29110RPI).

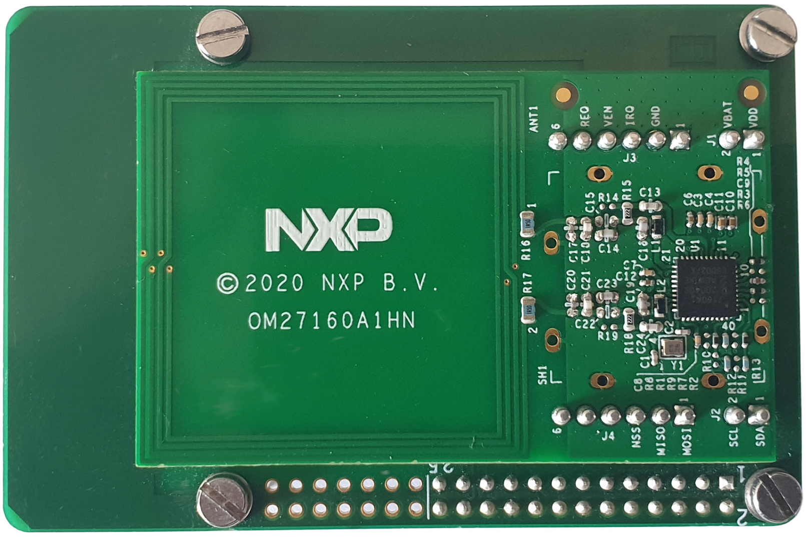


Figure 4. OM27160 RPI configuration

Then stack together the boards with the Raspberry Pi board.

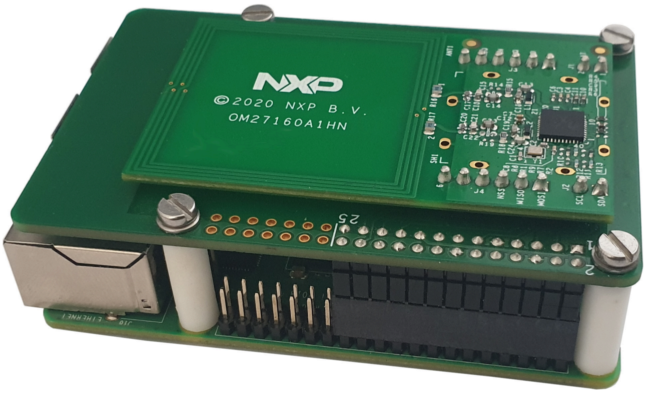


Figure 5. OM27160 evaluation kit and Raspberry Pi board stacked together

2.3 Software setup

Use Raspberry Pi OS (https://www.raspberrypi.org/software/operating-systems/). Guidelines to set up Linux environment on raspberry pi can be found here: https://www.raspberrypi.org/documentation/installation/installing-images/).

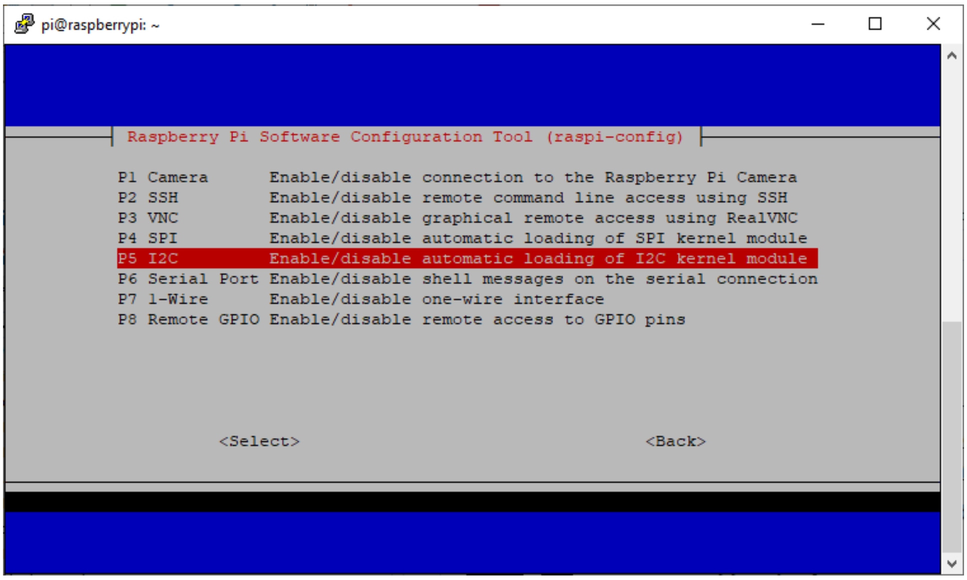
Below is the step-by-step procedure run from the Raspberry Pi to add software support for PN7160:

2.3.1 Enable I^{2}C interface (only for OM27160A1EVK)

1. Run command:

```
sudo raspi-config
```

2. Use the down arrow to select "`Interface Options`"
3. Arrow down to "`P5 I2C`"
4. Select "`yes`" when it asks you to enable I2C
5. Use the right arrow to select the `<Finish>` button



```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins

          <Select>                          <Back>
```

Figure 6. Enabling I^{2}C interface on Raspberry Pi OS

To verify the I^{2}C interface is enabled, enter the following command `ls /dev/i2c*`.

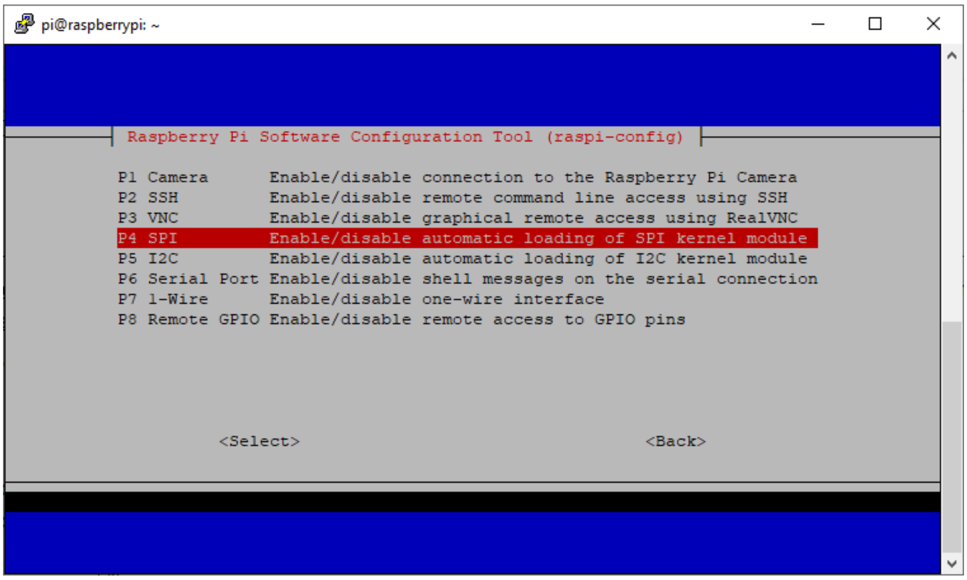
The Pi should respond with "`/dev/i2c-1`" which represents the user-mode I^{2}C interface to which is connected the PN7160.

2.3.2 Enable SPI interface (only for OM27160B1EVK)

1. Run command:

```
sudo raspi-config
```

2. Use the down arrow to select "`Interface Options`"
3. Arrow down to "`P4 SPI`"
4. Select "`yes`" when it asks you to enable SPI
5. Use the right arrow to select the `<Finish>` button



```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins

          <Select>                                <Back>
```

Figure 7. Enabling SPI interface on Raspberry Pi OS

To verify the SPI interface is enabled, enter the following command `ls /dev/spi*`.

The Pi should respond with "`/dev/spi0.0`" which represents the user-mode SPI interface to which is connected the PN7160.

2.3.3 Install necessary tools

Execute the command:

```
sudo apt-get install autoconf automake libtool git
```

2.3.4 Clone Linux libnfc-nci library repository

Execute the command:

```
git clone https://github.com/NXPNFCLinux/linux_libnfc-nci.git -b NCI2.0_PN7160
```

2.3.5 Configure the library

Execute the commands:

```
cd linux_libnfc-nci
./bootstrap
./configure
```

2.3.6 Set the library to map I^{2}C interface (only for OM27160A1EVK)

Edit `linux_libnfc-nci/conf/libnfc-nxp.conf` file to update `NXP_TRANSPORT` and `NXP_NFC_DEV_NODE` settings as shown below:

```
###############################################################################
# TRANSPORT Type
# 0x00 - I2C /SPI for noraml nxpnfc driver
# 0x01 - Not Used, kept to align with Android code
# 0x02 - ALT_I2C
# 0x03 - ALT_SPI
NXP_TRANSPORT=0x02

###############################################################################
# NXP HW Device Node information
NXP_NFC_DEV_NODE="/dev/i2c-1"
```

2.3.7 Set the library to map SPI interface (only for OM27160B1EVK)

Edit `linux_libnfc-nci/conf/libnfc-nxp.conf` file to update `NXP_TRANSPORT` and `NXP_NFC_DEV_NODE` settings as shown below:

```
###############################################################################
# TRANSPORT Type
# 0x00 - I2C /SPI for noraml nxpnfc driver
# 0x01 - Not Used, kept to align with Android code
# 0x02 - ALT_I2C
# 0x03 - ALT_SPI
NXP_TRANSPORT=0x03

###############################################################################
# NXP HW Device Node information
NXP_NFC_DEV_NODE="/dev/spidev0.0"
```

2.3.8 Build and install the library

Execute the commands:

```
make
sudo make install
export LD_LIBRARY_PATH=/usr/local/lib
```

To make this last setting permanent, run the following command:

```
echo "export LD_LIBRARY_PATH=/usr/local/lib" >> .bashrc
```

2.3.9 Run the demo application (built and installed together with the library during previous step)

To simply display all data collected from remote NFC device (Peer, reader/writer or card), run the demo application in "poll mode" executing the command:

```
nfcDemoApp poll
```

For more details about the demo application modes execute command:

```
nfcDemoApp --help
```

For more detailed information about the demo application, but also for additional example applications, refer to ref.[2].

3 Quick startup with Arduino interface board

Refer to ref.[5] for details about PN7160 demonstration using OM27160 ARD configuration on Arduino compatible NXP's MCU boards.

4 References

[1] The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It is capable of doing everything you would expect a desktop computer to do, from browsing the Internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. For more information, visit https://www.raspberrypi.org/

[2] AN13287 - PN7160 Linux porting guide: https://www.nxp.com/doc/AN13287

[3] Android is an open source software stack for a wide range of mobile devices and a corresponding open source project led by Google.
For more information, visit https://source.android.com/

[4] AN13189 - PN7160 Android porting guide: https://www.nxp.com/doc/AN13189

[5] AN13288 - PN7160 NXP-NCI2.0 MCUXpresso examples guide: https://www.nxp.com/doc/AN13288

5 Note about the source code in the document

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

6 Revision history

Table 1. Revision history

Document ID	Release date	Description
AN12991 v.1.2	5 September 2025	Editoral changes (typos, etc.). Document adapted to latest NXP guidelines. • Section 2.1 "Required items": updated the term "raspbian" to "Raspberry Pi OS". • Section 2.3 "Software setup": corrected typo, updated the term "raspbian" to "Raspberry Pi OS". • Section 4 "References": updated. • Section 5 "Note about the source code in the document": added.
AN12991 v.1.1	13 September 2021	• Security status changed to "Company public", no content change.
AN12991 v.1.0	25 August 2021	• Initial version.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

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