AN13794

Random Number Generation Using ROM API in LPC553x (Non-Secure) Devices

Rev. 1 — 21 August 2023 Application note

Document Information

Information	Content
Keywords	AN13794, LPC553x, random number generator (RNG), ROM API
Abstract	This application note provides an overview of how the RNG module can be used through ROM API to generate random numbers for LPC553x devices.

1 Introduction

This application note provides an overview of how the RNG module can be used through the ROM API to generate random numbers for LPC553x devices. RNG produces a sequence of numbers that are highly unpredictable and can be used to mask data or produce a key. RNG can also be used for security and cryptographic purposes. However, for applications requiring high security, NXP secure parts, such as LPC55S3x, provide a better option.

1.1 Acronyms

Table 1 defines the acronyms used in this document.

Table 1. Acronyms

Acronym	Definition
RNG	Random number generator
API	Application programming interface
DRBG	Deterministic random bit generator
TRNG	True random number generator

1.2 ROM API structure

Figure 1 shows the ROM API structure. It contains several API drivers with absolute ROM API function addresses, which can be called using function pointers. Some of the functionalities provided by the ROM API enable serial NOR flash, eFuse OTP memory read and programming, support for crypto functions, in-application programming, and so on. In LPC553x, the DRBG generates a random number. Here, the output of a TRNG is used as an entropy source to determine the seed of a DRBG. The ROM API table is located at address 0x0302FC00. The RNG API can be accessed using the NBOOT API driver located at address 0x0302FC28.

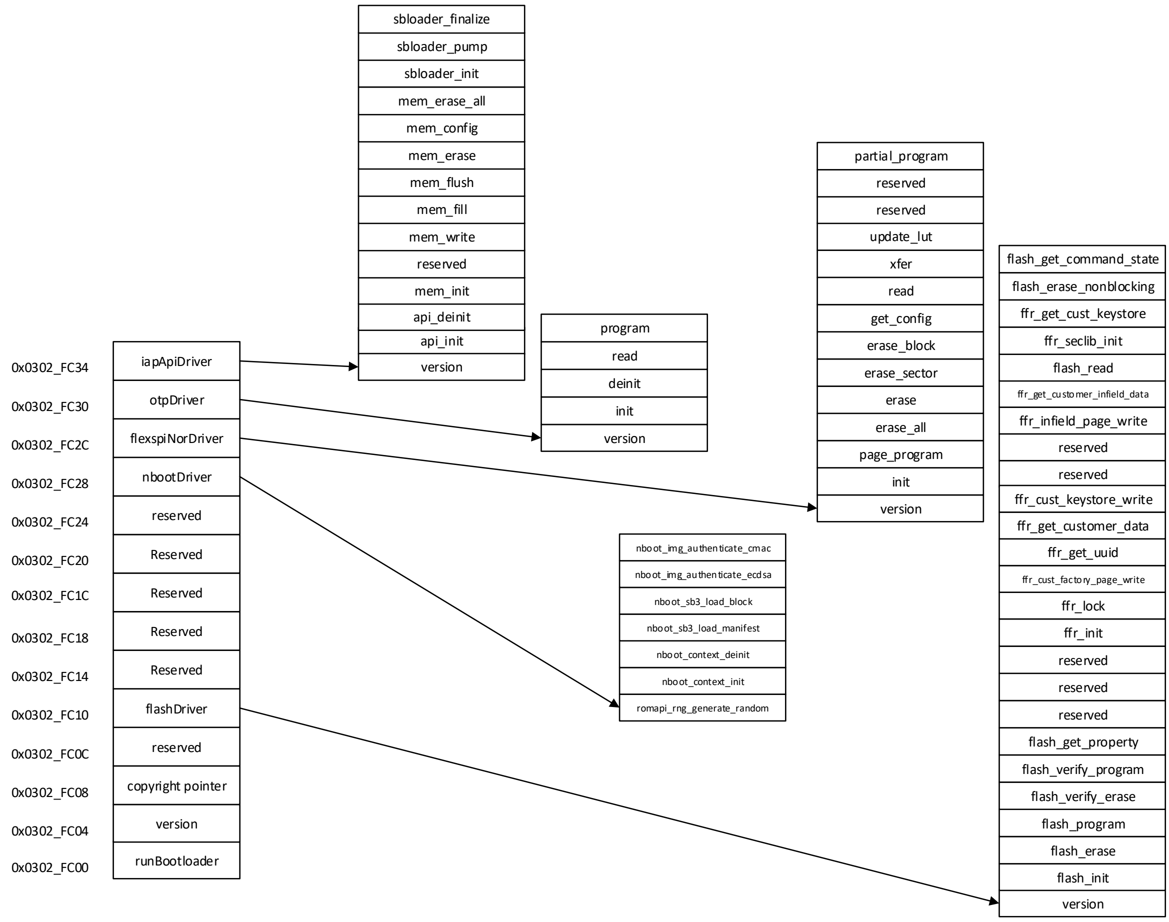


Figure 1. ROM API structure

1.3 NBOOT API driver

The main purpose of the NBOOT ROM API is to provide access to the functions used and implemented in ROM to generate random numbers and authenticate application images. Other NBOOT API functions, such as NBOOT SB3.1 API functions, can also verify the SB3.1 header (manifest) signature and decrypt individual data blocks without processing the entire SB file.

Section 1.3.1 describes the API for generating a random number.

1.3.1 romapi_rng_generate_random API

This ROM API function is used to generate a random number with specified length.

Prototype:

```
romapi_status_t (*romapi_rng_generate_random)(uint8_t *output, size_t
 outputByteLen);
```

Parameters:

- `output [in]`: Pointer to random number buffer

- `outputByteLen [in]`: Length of generated random number in bytes. The length has to be in the range from 1 to 2^{16}

Return values:

- `kStatus_NBOOT_InvalidArgument`: Invalid input parameters (input pointers point to NULL or length is invalid)
- `kStatus_NBOOT_Success`: Operation successfully finished
- `kStatus_NBOOT_Fail`: Error occurred during operation

2 Setup and SDK example

There is a software package available to download with this application note. The package contains the demo project for MCUXpresso, Keil, and IAR platforms that demonstrates the use of the ROM API to generate random numbers. The projects can be imported into their respective IDEs and similar steps mentioned in the below sections can be followed for each IDE.

A respective SDK package for each IDE is required for building and flashing the project onto the MCU. The SDK packages for the different platforms are available at https://mcuxpresso.nxp.com/en/welcome.

2.1 Hardware setup

The LPCXpresso55S36 development board is used for the hardware setup, which is connected to the host computer through the J1 debug probe. The J1 debug probe uses the MCU-LINK VCOM output, which acts as a USB-to-serial bridge to the host computer and provides the CMSIS-DAP debug interface.

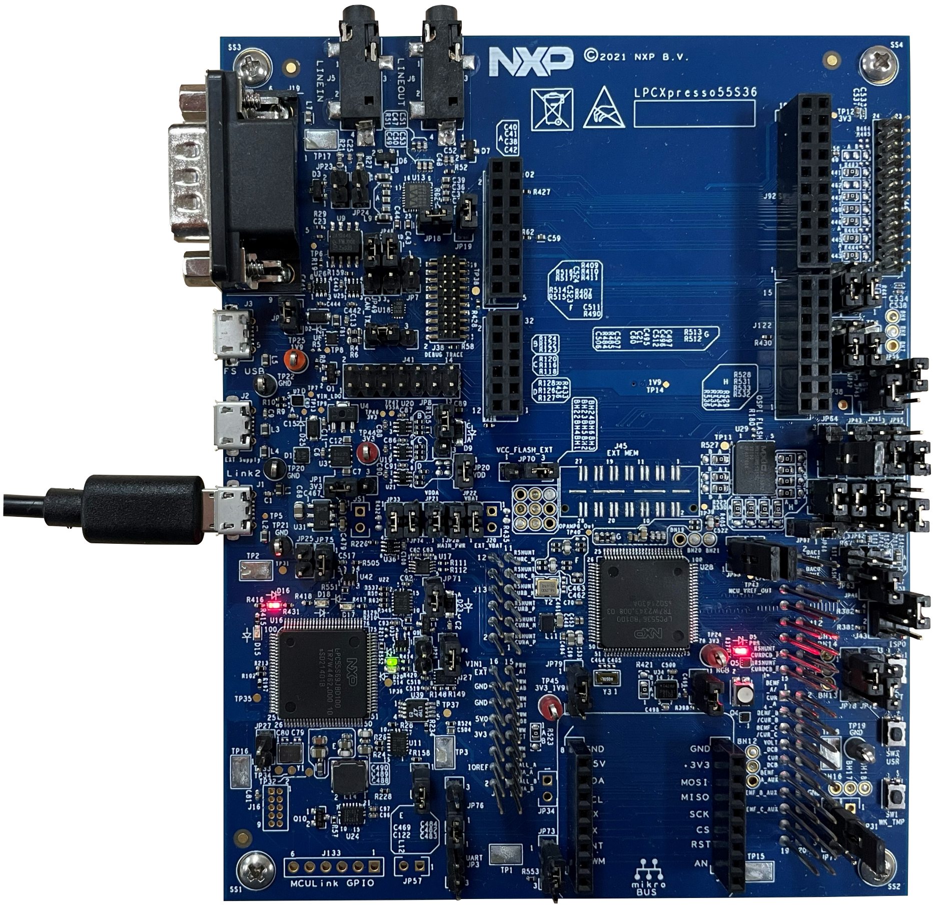


Figure 2. LPCXpresso55S36 development board; USB Micro-B cable connected to J1 debug probe

2.2 Software setup

For the below provided application example, MCUXpresso IDE v11.8 is used (available for download at https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE). An SDK for LPC5536 is also required for building and debugging the code. SDK_2.14.0 is used for the below example. The output can be displayed on either the Terminal window in MCUXpresso IDE or a terminal application, such as, Tera Term. While using Tera Term or any other terminal application, the below settings must be used:

- 115,200 baud rate
- No parity
- 8 data bits
- 1 stop bits

2.3 SDK example

The below example code shows how to generate a random number by using the ROM API.

```
/*
 * Copyright 2016-2022 NXP
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
```

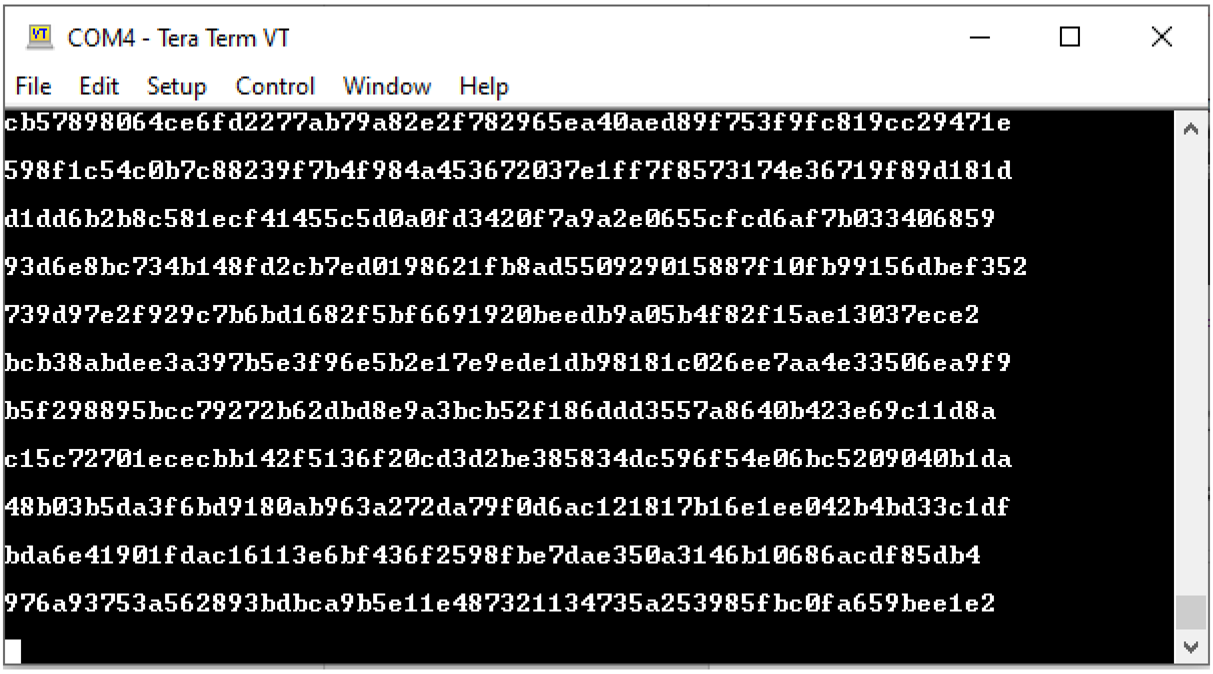
```
 *
 * o Redistributions of source code must retain the above copyright notice, this list
 *   of conditions and the following disclaimer.
 *
 * o Redistributions in binary form must reproduce the above copyright notice, this
 *   list of conditions and the following disclaimer in the documentation and/or
 *   other materials provided with the distribution.
 *
 * o Neither the name of NXP Semiconductor, Inc. nor the names of its
 *   contributors may be used to endorse or promote products derived from this
 *   software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
#include "fsl_device_registers.h"
#include "fsl_debug_console.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "board.h"
#include "fsl_nboot.h"
/*******************************************************************************
 * Prototypes
 ******************************************************************************/
/*! @brief Interface for the NBOOT API driver.*/
typedef struct
{
romapi_status_t (*romapi_rng_generate_random)(uint8_t *output, size_t outputByteLen);
nboot_status_t (*nboot_context_init)(nboot_context_t *context);
nboot_status_t (*nboot_context_deinit)(nboot_context_t *context);
nboot_status_protected_t (*nboot_sb3_load_manifest)(nboot_context_t *context,
uint32_t *manifest,
nboot_sb3_load_manifest_parms_t *parms);
nboot_status_protected_t (*nboot_sb3_load_block)(nboot_context_t *context,
uint32_t *block);
nboot_status_protected_t (*nboot_img_authenticate_ecdsa)(nboot_context_t *context,
uint8_t imageStartAddress[],
nboot_bool_t *isSignatureVerified,
nboot_img_auth_ecdsa_parms_t *parms);
nboot_status_protected_t (*nboot_img_authenticate_cmac)(nboot_context_t *context,
uint8_t imageStartAddress[],
nboot_bool_t *isSignatureVerified,
nboot_img_authenticate_cmac_parms_t *parms);
} nboot_interface_t;
/*******************************************************************************
 * Definitions
 ******************************************************************************/
/* Locating the random number generation API using the ROM API structure */
#define ROM_API_TREE *((uint32_t *)(0x1302fc00)+10)
#define NBOOT_API_TREE ((nboot_interface_t *) ROM_API_TREE)
/*******************************************************************************
 * Code
 ******************************************************************************/
/*!
 * @brief Main function
 */
int main(void)
{
 char ch;
 uint8_t rndBuffer[32];
 static nboot_context_t contextt;
 /* Init board hardware. */
 /* attach main clock divide to FLEXCOMM0 (debug console) */
 CLOCK_SetClkDiv(kCLOCK_DivFlexcom0Clk, 0u, false);
 CLOCK_SetClkDiv(kCLOCK_DivFlexcom0Clk, 1u, true);
 CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
 BOARD_InitPins();
 BOARD_BootClockPLL150M();
 BOARD_InitDebugConsole();
 /* Initialization of nboot context data structure */
 nboot_status_t status_nboot = NBOOT_API_TREE->nboot_context_init(&contextt);
/* Initialization of nboot context data structure */
 nboot_status_t status_nboot = NBOOT_API_TREE->nboot_context_init(&contextt);
 for(int x=0;x<11;x++){
 /* Generate random number with specified length */
 romapi_status_t status_romapi = NBOOT_API_TREE->romapi_rng_generate_random(&rndBuffer[0], 32);
 /* If operation is successful, print the random number generated */
 if(status_romapi == kStatus_NBOOT_Success){
```

```
 //PRINTF("Generated Random Number: \r\n");
 for(int i=0;i<32;i++)
 PRINTF("%x", rndBuffer[i]);
 PRINTF("\r\n");
 }
 /* Print error message if invalid arguments */
 if(status_romapi == kStatus_NBOOT_InvalidArgument){
 PRINTF("Invalid Arguments\n");
 }
 /* Print error message if operation failed */
 if(status_romapi == kStatus_NBOOT_Fail){
 PRINTF("Random Number Generation Failed!\n");
 }
 PRINTF("\r\n");
 }
 while (1)
 {
 ch = GETCHAR();
 PUTCHAR(ch);
 }
}
```

3 Output

The output is displayed on the Tera Term application using the UART COM4 port. After the successful build and flashing of the code onto the MCU, the output is displayed, as shown in Figure 3. 11 strings of random numbers are printed, with each number printed in a new line and having a length of 32 bytes. Hex format is used for displaying the generated random numbers.

Note: *All the random numbers generated are different, ensuring the proper working of the RNG module.*



```
cb57898064ce6fd2277ab79a82e2f782965ea40aed89f753f9fc819cc29471e
598f1c54c0b7c88239f7b4f984a453672037e1ff7f8573174e36719f89d181d
d1dd6b2b8c581ecf41455c5d0a0fd3420f7a9a2e0655cfcd6af7b033406859
93d6e8bc734b148fd2cb7ed0198621fb8ad550929015887f10fb99156dbef352
739d97e2f929c7b6bd1682f5bf6691920beedb9a05b4f82f15ae13037ece2
bcb38abdee3a397b5e3f96e5b2e17e9ede1db98181c026ee7aa4e33506ea9f9
b5f298895bcc79272b62dbd8e9a3bcb52f186ddd3557a8640b423e69c11d8a
c15c72701ececbb142f5136f20cd3d2be385834dc596f54e06bc5209040b1da
48b03b5da3f6bd9180ab963a272da79f0d6ac121817b16e1ee042b4bd33c1df
bda6e41901fdac16113e6bf436f2598fbe7dae350a3146b10686acdf85db4
976a93753a562893bdbca9b5e11e487321134735a253985fbc0fa659bee1e2
```

Figure 3. Generated output displayed on Tera Term

4 References

The following document is referred:

- *LPC553x Reference Manual* (document LPC553xRM)

5 Note about the source code in the document

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

6 Revision history

Table 2 summarizes the revisions to this document.

Table 2. Revision history

Revision number	Release date	Description
1	21 August 2023	Initial public release

7 Legal information

7.1 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.

7.2 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, 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.

7.3 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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Contents

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

For more information, please visit: http://www.nxp.com

Date of release: 21 August 2023
Document identifier: AN13794