AN14935

How to Use Low Power Compute Mode on i.MX 95 and i.MX 943

Rev. 1.0 — 16 March 2026 Application note

Document information

Information	Content
Keywords	AN14935, i.MX 95, i.MX 943, Low Power Compute
Abstract	This documentation describes how to use the Low Power Compute features supported in the System Manager.

1 Introduction

The i.MX 9 applications processor family supports a wide range of edge applications across automotive, industrial, networking, connectivity, and advanced human-machine interface markets. It features a heterogeneous compute architecture with three types of cores optimized for three types of domains. Up to six Arm Cortex-A55 cores provide high performance for the application domain. The Arm Cortex-M7 and Arm Cortex-M33 cores support two independent real-time domains. The Arm Cortex-M7 targets high-performance real-time usage, and the Cortex-M33 is intended for safety-critical, networking, and low-power usage.

Low Power Compute (LPC) mode is a system configuration in which most high-performance domains – such as the Cortex-A55 cluster and the DRAM subsystem – are powered down. This mode enables the system to continue operating with significantly reduced power consumption by executing workloads solely on Cortex-M processors. LPC mode is ideal for control loops, low-rate data processing, basic communications, or standalone subsystem operation.

This documentation describes how to use the Low Power Compute features supported in the System Manager.

2 Architecture

This section describes the SoC architecture and software architecture.

2.1 SoC architecture

This section takes the i.MX 95 SoC architecture block diagram as an example. The i.MX 943 is similar.

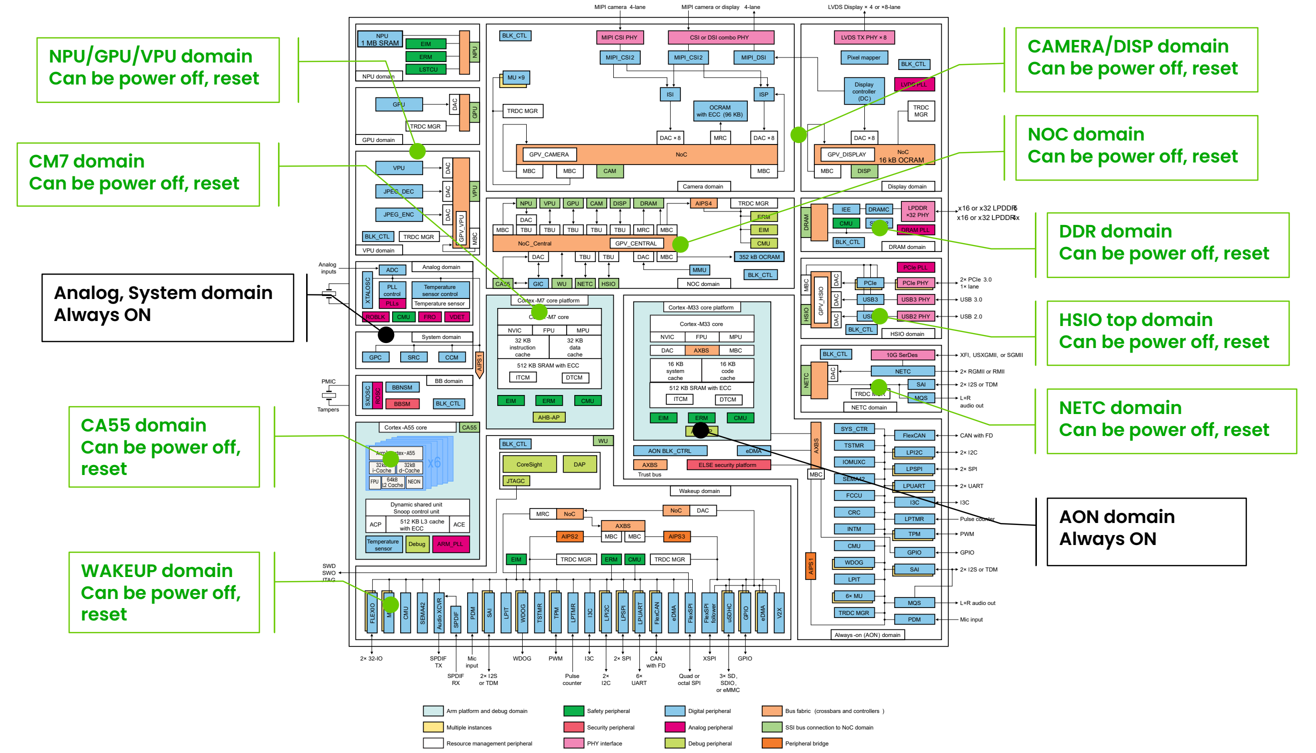

Figure 1. i.MX 95 SoC architecture

2.1.1 Domains

The i.MX 95/943 SoC is composed of many domains. Most of the domains has dedicated power switches, which can be turned on and off separately.

Table 1. Domains

Domain	Description	Power off?
Always-On (AON) domain	Known as AONMIX. The AON domain includes the Cortex-M33 processor (runs System Manager), the EdgeLock Enclave, and a set of peripherals.	NO
Wakeup (WU) domain	Known as WAKEUPMIX. The Wakeup domain includes an assortment of peripherals that support audio, memory, timer, communication, and motor drive (i.MX 943) applications.	YES
Battery-Backup (BB) domain	Known as BBSMMIX. The BBSM serves as the nonvolatile security logic and storage for the ELE. This domain also controls the 32 kHz oscillator source for the chip.	NO
Clock, Reset, and Power (CCMSRCGPC) domain	SRC controls the reset and booting of the chip. The CCM clock root generation logic generates various clock roots and delivers them to each module. GPC controls the power mode of the chip and manages the power gating in low-power modes and the power-up and power-down sequences.	NO
Analog domain	Known as ANAMIX. The Analog domain includes the 24 MHz OSC that generates the reference clock to the chip. The oscillators, PLLs, and pre-dividers generate clock sources with fixed or variable frequency.	YES
Cortex-M7 platform domain	Known as M7MIX. The Cortex-M7 domain consists of the Cortex-M7 platform, which targets real-time and safety applications.	YES
Cortex-A55 platform domain	The Cortex-A55 domain consists of the six Cortex-A55 cores to make up the application cluster.	YES
Network-on-Chip (NoC) domain	Known as NOCMIX. The NoC domain provides interconnectivity between several domains within the chip.	YES
NET Controller (NETC) domain	Known as NETCMIX. The NETC domain includes Ethernet, time-sensitive networking functions. i.MX 943 also includes Ethernet switch and M33_Sync core in this domain.	YES
DRAM domain	The DRAM and PHY DRAM controllers support LPDDR5 and LPDDR4x DDR SDRAMs.	YES
High-Speed I/O (HSIO) domain	The PCIe and USB controllers of the HSIO domain enable high-speed communication.	YES
Neural Processing Unit (NPU) domain	Known as NPUMIX. The NPU domain consists of the NPU complex, which provides hardware acceleration for ML applications.	YES
Graphics Processing Unit (GPU) domain **(i.MX 95 only)**	Known as GPUMIX. The GPU domain features a GPU graphics acceleration platform for rendering complex graphics data and performing general-purpose processing tasks.	YES
Video Processing Unit (VPU) domain **(i.MX 95 only)**	Known as VPUMIX. The VPU domain includes a high-performing video-processing unit.	YES
Camera domain **(i.MX 95 only)**	Known as CAMERAMIX. The modules in the Camera domain provide an interface to CSI-2-compliant sensors, perform image signal processing on the camera stream, and provide an interface to DSI-compliant display devices.	YES

Table 1. Domains*...continued*

Domain	Description	Power off?
Display domain	Known as DISPMIX. The i.MX 95 Display domain contains two display engines for DSI/LVDS interfaces and 2D blit operations. The i.MX 943 Display domain contains DCIF controller and PXP 2D engine.	YES

Note: *YES = can power off*

2.1.2 SoC driving mode

i.MX 95/943 supports different driving modes: Over Drive (OD), Nominal Drive (ND), Low Drive (LD) mode. Each driving mode has its own typical voltage for the core power supply: VDD_SOC.

Note: *VDD_ARM on i.MX 95 can also support OD/ND/LD driving mode. Because VDD_ARM is powered off in low-power compute mode, this document focuses on the VDD_SOC only.*

Table 2. Voltage table for VDD_SOC

	Over drive mode	Nominal drive mode	Low drive mode
VDD_SOC	0.9 V	0.85 V	0.8 V

Reducing core voltage can certainly save power, but it also lowers the performance level of each domain. The table below shows the maximum clock frequency for each domain under different driving modes.

In the following tables, **Park mode** refers to a software-defined low-drive mode that uses an extremely low, fixed 24 MHz clock from the OSC24 source, with all PLLs turned off to save power.

Table 3. i.MX 95 performance level table (Max. frequency)

Domain	Over drive mode (MHz)	Nominal drive mode (MHz)	Low drive mode (MHz)	Park mode (MHz)
CM33	333	250	166	24
CM7	800	666	400	24
HSIO	500	400	250	24
NPU	1000	800	500	24
GPU	100	800	500	24
VPU	666	500	333	24
Camera	800	666	400	24
Display	800	666	400	24
NOC	800	666	400	24
WAKEUP	400	333	200	24

Table 4. i.MX 943 performance level table (Max. frequency)

Domain	Over drive mode (MHz)	Nominal drive mode (MHz)	Low drive mode (MHz)	Park mode (MHz)
CM33	333	250	166	24
CM33_Sync	266	250	142	24
CM7 (0/1)	800	666	400	24

Table 4. i.MX 943 performance level table (Max. frequency)*...continued*

Domain	Over drive mode (MHz)	Nominal drive mode (MHz)	Low drive mode (MHz)	Park mode (MHz)
HSIO	500	400	250	24
NPU	1000	800	500	24
Display	400	333	200	24
NOC	500	400	250	24
WAKEUP	400	333	200	24

To balance power and performance, users can choose the lowest driving mode, which lowers the core supply voltage while still meeting performance requirements within the appropriate frequency range.

2.2 Software architecture

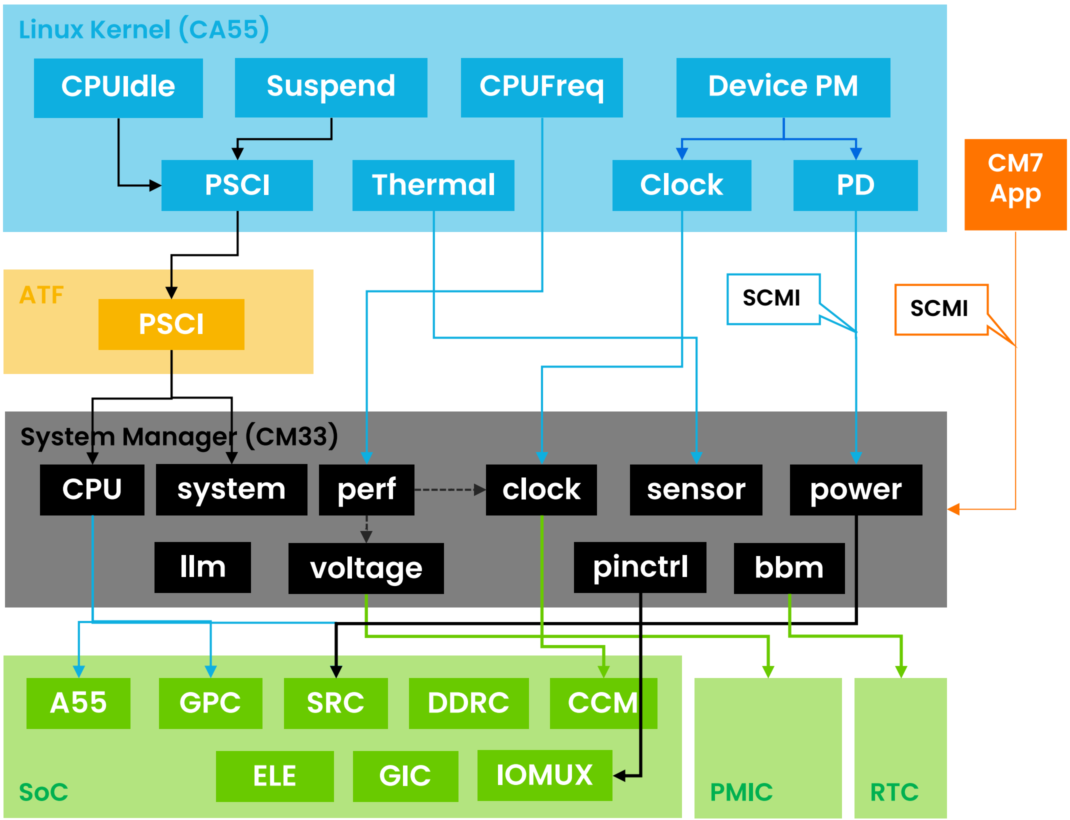

Figure 2. Software architecture

As shown in the software architecture block diagram above, the System Manager provides a mechanism to manage the power, performance, clock, reset, and pin control by providing System Control and Management Interface (SCMI defined by Arm) to other cores. The software running on Cortex-A55/M7/M33 can determine how to configure and use different power modes.

In the default BSP, the Linux power-management and device subsystems communicate with the System Manager directly by SCMI or through ATF to implement features of CPU idle, Suspend-to-RAM, CPU frequency, clock management, and power-domain. The CM7 SDK power mode switch demo manages the CM7 power status using the SCMI API.

Most high-performance and multimedia domains have their power status tied to the Cortex-A55 domain. This means when Linux is suspended to RAM (with the Cortex-A55 domain powered off), Linux requests the System Manager to turn off the domains, such as DRAM, VPU, GPU, NPU, Display, Camera, and HSIO. Those domains remain off during suspend. Other domains like WAKEUP, NOC, NETC, CM7 can be managed by the CM7 software.

The Low Power Compute feature is designed for the use cases where Linux is suspended to RAM (with the Cortex-A55 domain powered off). In this mode, the CM7 core may remain idle or execute tasks that do not require high performance, such as math computing, low speed peripherals communication like FlexCAN and SPI.

3 Use case example

Two representative use cases demonstrate the operational benefits of LPC mode:

- **CM7 computing in TCM**
  The Cortex-M7 runs entirely from TCM, with all non-essential domains powered down and the DDR device kept in retention mode. This configuration supports deterministic, low-latency compute tasks such as FFT processing. Enable LPC under this use case, which can save around 400 mW on the total SoC power consumption.
- **NETC switch standalone mode**
  i.MX 943 NETC switch standalone mode, in which the NETC domain and CM33S (`CM33_Sync`) core remain powered on while the Cortex-A55, DRAM, NOC, and WAKEUP domains are turned off. This mode allows TSN packet forwarding in an ultra-low-power system state. Enable LPC under this use case, which can save around 480 mW on the total SoC power consumption.
  This example code can be found as the `netc_switch_standalone` demo in formal SDK release.

4 How to

To request and enter low power compute mode, perform the following steps:

1. Place the Linux system into Suspend-to-RAM mode.
   Run `echo mem > /sys/power/state` in the Linux console.
2. The software on CM7 or CM33S (i.MX 943 only) requests for:
   a. System sleep with performance-level configurations.
   b. Power off domains, which would not be used.
   c. Core run under low-power compute mode.
3. System Manager enters IDLE (automatically or by entering `idle` on the interface).

This section focuses on Step 2 for the Cortex-M core software.

5 SCMI APIs

To support Low Power Compute mode, there are several SCMI APIs defined based on the Arm basic SCMI protocol (`SCMI_PROTOCOL_SYS`) and NXP vendor extended protocol (`SCMI_PROTOCOL_CPU`).

Table 5. Summary table

API	Purpose	Typical use in LPC
`SCMI_SystemPowerStateSet`	Set the system sleep mode with performance level (drive mode).	Select low drive mode.

Table 5. Summary table*...continued*

API	Purpose	Typical use in LPC
`SCMI_CpuPdLpmConfigSet`	Power off/on specific domains.	Power off NOC/WAKEUP domain, etc.
`SCMI_CpuIrqWakeSet`	Set/Clear GPC wake IRQs.	Clear pending IRQs before entering LPC.
`SCMI_CpuSleepModeSet`	Set CPU low-power state.	Keep CM33/CM7 in RUN with LPC.

5.1 SCMI_SystemPowerStateSet

```
int32_t SCMI_SystemPowerStateSet(uint32_t channel, uint32_t flags, uint32_t
 systemState);
```

Description:

This function sends a command to set the system power state through the System Control and Management Interface (SCMI) protocol. It allows the caller to request various system power transitions, such as shutdown, reset, suspend, and sleep operations with performance level settings.

Protocol: `SCMI_PROTOCOL_SYS (0x12)`

Message ID: `SCMI_MSG_SYSTEM_POWER_STATE_SET (0x3)`

Parameters:

- `channel`: SCMI channel identifier used for communication, usually **0**
- `flags`: Graceful request
  - `0x1`: Graceful request
  - `0x0`: Non graceful request
- `systemState`: Desired system sleep mode settings

Table 6. System sleep mode settings

Command Msg ID	Sleep mode performance level	Sleep mode reserved	Sleep flags
Bits [31:24]	Bits [23:20]	Bits [19:16]	Bits [15:0]
0xC0	0x0 - Park mode 0x1 - LowDrive mode 0x2 - NominalDrive mode 0x3 - OverDrive mode	Reserved	0x1 – Keep SM active 0x2 - Keep FRO active 0x4 - Keep SYSCTR active 0x8 - No PMIC_STBY assertion 0x10 - Keep OSC24M active

Returned Value:

- **0**: Success. Power state command accepted.
- Others: Failure.

5.2 SCMI_CpuPdLpmConfigSet

```
int32_t SCMI_CpuPdLpmConfigSet(uint32_t channel, uint32_t cpuId, uint32_t
 numConfigs, const scmi_pd_lpm_config_t *pdConfigs);
```

Description:

This function configures a list of power domains Low Power Mode settings through the System Control and Management Interface (SCMI) protocol. It allows the caller to request power off or keep power of a set of domains under suspend or LP Compute mode.

Protocol: `SCMI_PROTOCOL_CPU (0x82)`

Message ID: `SCMI_MSG_CPU_PD_LPM_CONFIG_SET (0xA)`

Parameters:

- `channel`: SCMI channel identifier used for communication, usually **0**
- `cpuId`: CPU identifier to configure
- `numConfigs`: number of power domain LPM configuration structures in `pdConfigs`
- `pdConfigs`: LPM configuration structure array

```
typedef struct
{
uint32_t domainId;     /* domain id for this configure */
uint32_t lpmSetting;  /* lpm settings */
uint32_t retMask;      /* domain memory retention */
} scmi_pd_lpm_config_t;

lpmSetting:
```

  - 0: Off
  - 1: On when the CPU is RUN
  - 2: On when the CPU is RUN/WAIT, off in STOP/SUSPEND
  - 3: On when the CPU is RUN/WAIT/STOP, off in SUSPEND
  - 4: On always

Returned Value:

- **0**: Success. LPM configuration request successfully,
- Others: Failure,

5.3 SCMI_CpuIrqWakeSet

```
int32_t SCMI_CpuIrqWakeSet(uint32_t channel, uint32_t cpuId, uint32_t maskIdx,
 uint32_t numMask, const uint32_t *mask);
```

Description:

This function configures which interrupts can wake up a CPU from low-power or sleep states through the System Control and Management Interface (SCMI) protocol. It can be used to clear all the wakeup sources in GPC for the CPU before it goes to low power compute mode.

Protocol: `SCMI_PROTOCOL_CPU (0x82)`

Message ID: `SCMI_MSG_CPU_IRQ_WAKE_SET (0x8)`

Parameters:

- `channel`: SCMI channel identifier used for communication, usually **0**
- `cpuId`: CPU identifier to configure
- `maskIdx`: GPC IRQ wakeup mask register start index
- `numMask`: how many mask registers need to be configured
- `mask`: mask value array

Returned Value:

- **0**: Success. IRQ wake configuration applied.
- Others: Failure.

5.4 SCMI_CpuSleepModeSet

```
int32_t SCMI_CpuSleepModeSet(uint32_t channel, uint32_t cpuId, uint32_t flags,
 uint32_t sleepMode);
```

Description:

This function configures the CPU sleep mode through the System Control and Management Interface (SCMI) protocol. It allows the caller to request specific low-power states for the CPU core.

Protocol: `SCMI_PROTOCOL_CPU (0x82)`

Message ID: `SCMI_MSG_CPU_SLEEP_MODE_SET (0x7)`

Parameters:

- `channel`: SCMI channel identifier used for communication, usually 0
- `cpuId`: CPU identifier to configure
- `flags`
  - `0x1`: switch wakeup from GPC to GIC for A55 (For A55 CPU only)
  - `0x2`: wake A55 cores during A55 platform wakeup (For A55 CPU only)
  - `0x4`: enable LP compute during system sleep (For Cortex-M CPU)
- `sleepMode`: Desired CPU sleep mode
  - `0x0`: RUN
  - `0x1`: WAIT
  - `0x2`: STOP
  - `0x3`: SUSPEND

Returned Value:

- **0**: Success. Sleep mode command accepted.
- Others: Failure.

6 Program flow

This section takes the i.MX 943 NETC switch standalone use case as an example, which can be found in the SDK demo.

6.1 Entering LPC

To enter LPC, perform the following steps:

1. Specify the performance level required during system sleep.
   For example, set the low drive mode performance level during system sleep.
   ```
   SCMI_SystemPowerStateSet(SCMI_A2P, SCMI_SYS_FLAGS_GRACEFUL(1U),

   SCMI_SYS_STATE_MODE | SCMI_SYS_SLEEP_MODE(0x10) | SCMI_SYS_SLEEP_FLAGS(0));
   ```
2. Specify which power domains need to be on or off during system sleep.

For example, keep the NETC domain and memory on, and power off the NOC and WAKEUP domains.

```
lpmConfig[0].domainId = SYSTEM_POWER_PLATFORM_MIX_SLICE_IDX_NETC;
lpmConfig[0].lpmSetting = SCMI_CPU_LPM_SETTING_ON_ALWAYS;
lpmConfig[0].retMask = 1U << SYSTEM_POWER_PLATFORM_MEM_SLICE_IDX_NETC;
lpmConfig[1].domainId = POWER_MIX_SLICE_IDX_NOC;
lpmConfig[1].lpmSetting = SCMI_CPU_LPM_SETTING_ON_RUN_WAIT_STOP;
lpmConfig[1].retMask = 0U;
lpmConfig[2].domainId = POWER_MIX_SLICE_IDX_WAKEUP;
lpmConfig[2].lpmSetting = SCMI_CPU_LPM_SETTING_ON_RUN_WAIT_STOP;
lpmConfig[2].retMask = 0U;
SCMI_CpuPdLpmConfigSet(SCMI_A2P, SYSTEM_PLATFORM_M33S_ID, 3U, lpmConfig);
```

3. Clear all GPC wake sources because the CPU remains active during system sleep.

```
SCMI_CpuIrqWakeSet(SCMI_A2P, SYSTEM_PLATFORM_M33S_ID, 0U, 15, wakeMask);
```

Note: *`wakeMask` is an array of 15 items, with all values of `0xFFFFFFFF`.*

4. Request LP COMPUTE when setting the CPU sleep mode to **RUN**.

```
SCMI_CpuSleepModeSet(SCMI_A2P, SYSTEM_PLATFORM_M33S_ID,
SCMI_CPU_FLAGS_LP_COMPUTE(1U), SCMI_CPU_SLEEP_RUN);
```

6.2 Exiting LPC

Exiting LPC is simple. Call `SCMI_CpuSleepModeSet` to draw the CM33 or CM7 out of LPC mode.

```
SCMI_CpuSleepModeSet(SCMI_A2P, SYSTEM_PLATFORM_M33S_ID,
SCMI_CPU_FLAGS_LP_COMPUTE(0U), SCMI_CPU_SLEEP_RUN);
```

7 Conclusion

Low Power Compute mode provides an effective means to reduce system power consumption on the i.MX 95/943 platform while maintaining essential real-time or networking functionality. By applying SCMI-based control mechanisms, software can precisely manage domain power states and processor execution modes to meet application requirements with optimal efficiency.

8 Note about the source code in the document

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

9 Revision history

Table 7 summarizes the revisions to this document.

Table 7. Revision history

Document ID	Release date	Description
AN14935 v.1.0	16 March 2026	Initial public release

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

Contents

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