LPC55S1x/LPC551x

Errata sheet LPC55S1x/LPC551x

Rev. 2.0 — 19 December 2023 Errata

Document information

Information	Content
Keywords	LPC55S16JBD100, LPC55S16JEV98, LPC55S16JBD64, LPC55S14JBD100, LPC55S14JBD64, LPC5516JBD100, LPC5516JEV98, LPC5516JBD64, LPC5514JBD100, LPC5514JBD64, LPC5512JBD100, LPC5512JBD64,LPC5514JEV59, LPC5516JEV59 , LPC55S14JEV59, LPC55S16JEV59
Abstract	LPC55S1x/LPC551x errata

1 Product identification

The LPC55S1x/LPC551x VFBGA98 package has the following top-side marking:

- First line: LPC55S1x/LPC551x
- Second line: JEV98
- Third line: xxxxxxxx
- Fourth line: zzzyywwxR
  - yyww: Date code with yy = year and ww = week.
  - xR: Device revision A

The LPC55S1x/LPC551x HLQFP100 package has the following top-side marking:

- First line: LPC55S1x/LPC551x
- Second line: xxxxxxxx
- Third line: zzzyywwxR
  - yyww: Date code with yy = year and ww = week.
  - xR: Device revision A

The LPC55S1x/LPC551x HTQFP64 package has the following top-side marking:

- First line: LPC55S1x/LPC551x
- Second line: JBD64
- Third line: xxxx
- Fourth line: xxxx
- Fifth line: zzzyywwxR
  - yyww: Date code with yy = year and ww = week.
  - xR: Device revision A

2 Errata overview

Table 1. Functional problems table

Functional problems	Short description	Revision identifier	Detailed description
ROM.1	ROM fails to enter ISP mode when image is corrupted with flash pages in an erased or unprogrammed state.	A	Section 3.1
USB.1	USB HS host fails when connecting to an LS device (mouse).	A	Section 3.2
USB.2	Automatic USB rate adjustment not functional when using multiple hubs.	A	Section 3.3
USB.3	For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected.	A	Section 3.4
USB.4	In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer.	A	Section 3.5
USB.5	In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacket Size of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the	A	Section 3.6

Table 1. Functional problems table*...continued*

Functional problems	Short description	Revision identifier	Detailed description
	endpoint command/status list entry for the isochronous IN endpoint is not updated.		
USB.6	In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints.	A	Section 3.7
VBAT_DCDC.1	The minimum rise time of the power supply must be 2.6 ms or slower for Tamb = -40 C, and 0.5 ms or slower for Tamb = 0 C to +105 C.	A	Section 3.8
CAN-FD.1	Bus transaction abort could occur when CAN-FD peripheral is using secure alias.	A	Section 3.9
ROM.2	ROM API can’t be used correctly to update and read monotonic counter in CFPA.VENDOR_USAGE word.	A	Section 3.10
PUF SRAM.1	PUF SRAM needs to be reset during the startup of application to prevent high deep-sleep current consumption.	A	Section 3.11
PLL.1	PLL LOCK bit is not reliable	A	Section 3.12
ROM.3	Invalid TrustZone preset data structure can prevent ROM from applying specified debug settings	A	Section 3.13

Table 2. AC/DC deviations table

AC/DC deviations	Short description	Product version(s)	Detailed description
n/a	n/a	n/a	n/a

Table 3. Errata notes

Errata notes	Short description	Revision identifier	Detailed description
n/a	n/a	n/a	n/a

3 Functional problems detail

3.1 ROM.1: ROM fails to enter ISP mode when image is corrupted with flash pages in an erased or unprogrammed state

Introduction

On the LPC55S1x/LPC551x, if the image is corrupted with flash pages in an erased or unprogrammed state, the ROM may fail to automatically enter ISP mode.

Problem

When secure boot is enabled in CMPA, and the flash memory contains an erased or unprogrammed memory page inside the memory region specified by the image size field in the image header, the device does not automatically enter into ISP mode using the fallback mechanism, as in the case of a failed boot for an invalid image. This problem occurs when the application image is only partially written or erased but a valid image header is still present in memory.

Work-around

Perform a mass-erase to remove the incomplete and corrupted image using one of the following methods:

- Execute the erase command using Debug Mailbox. The device will enter directly into ISP mode after exiting the mailbox.
- Enter into ISP mode using the Debug Mailbox command and use the flash-erase command.
- Reset the device and enter into ISP mode using the ISP pin. Use the flash-erase command to erase the corrupted (incomplete) image.

3.2 USB.1: HS host fails when connecting with the LS device (mouse)

Introduction

The USB1 high-speed controller is available on select LPC55S1x/LPC551x devices and provides a plug-and-play connection of peripheral devices to a host with three different data speeds:

- high-speed with a data rate of 480Mbps.
- full-speed with a data rate of 12 Mbps.
- low-speed with a data rate of 1.5 Mbps.

Many portable devices can benefit from the ability to communicate with each other over the USB interface without intervention of a host PC.

Problem

USB HS host fails when connecting with an LS device (mouse).

Work-around

To support Full-Speed and Low-Speed applications, it is recommended to use the USB0 Full-Speed port and the USB1 High-speed port for Device or Host. In addition, should an application require support of Low-Speed USB devices with a USB High-Speed Host, this can be accomplished by inserting a USB Hub between the USB1 High-speed port and external USB devices.

3.3 USB. 2: Automatic USB rate adjustment is not functional when using multiple hubs

Introduction:

Full-speed and low-speed signaling uses bit stuffing throughout the packet without exception. If the receiver sees seven consecutive ones anywhere in the packet, then a bit stuffing error has occurred, and the packet should be ignored.

The time interval just before an End of Packet (EOP) is a special case. The last data bit before the EOP can become stretched by hub switching skews. This is known as dribble and can lead to a situation where dribble introduces a sixth bit that does not require a bit stuff. Therefore, the receiver must accept a packet where there are up to six full bit times at the port with no transitions prior to the EOP.

Problem:

The LPC55S1x/LPC551x devices use the start of an EOP for frequency measurements. This is not functional when going through multiple hubs that introduce a dribble bit because of hub switching skews. For this reason, the start of the EOP cannot be used for frequency measurements for automatic USB rate adjustment (by setting USBCLKADJ in the FRO192M_CTRL register). The problem does not occur when a single hub is used.

Work-around:

Use the FRO calibration library provided in technical note TNxxxxx. This library allows the application to have a crystal-less USB device operation in full-speed mode.

3.4 USB.3: For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected

Introduction

See the USB2.0 specification for details regarding the USB High-speed Detection Handshake protocol.

Problem

As a high-speed device, when certain full-speed hubs are connected, the USB device does not detect the HOST KJ sequence correctly and, as a result, does not recognize the speed of the connected host. In this case, the USB device can act erratically due to the wrong speed detection.

Work-around

There are two workarounds:

1. The software work-around below can be implemented in usb_dev_hid_mouse where API is called "USB_DeviceHsPhyChirpIssueWorkaround()". In event handler in USB_DeviceCallback(),
   - On "kUSB_DeviceEventBusReset" event, USB_DeviceHsPhyChirpIssueWorkaround() should be called to identify the speed of the host connected to. If full-speed host is connected or "isConnectedToFsHostFlag" is set , FORCE_FS (bit 21) of DEVCMDSTAT register should be set to force the device operating in full-speed mode.
   - On "kUSB_DeviceEventDetach" event, FORCE_FS (bit 21) of DEVCMDSTAT register should be cleared.
2. The software workaround below is available in tech note (TN00071) In event handler in USB_DeviceCallback(),
   - On "kUSB_DeviceEventAttach" event, set PHY_RX register trip-level voltage to the highest. USBPHY->RX &= ~(USBPHY_RX_ENVADJ_MASK);USBPHY->RX |= 2;.
   - On "kUSB_DeviceEventBusReset" event, check the DEVCMDSTAT[SPEED] to determine the connected bus speed. (SPEED are bits 22 and 23). If DEVCMDSTAT[SPEED]=FS, FORCE_FS (bit 21) of DEVCMDSTAT should be set to force the device operating in full-speed mode.
   - On "kUSB_DeviceEventGetDeviceDescriptor" event, or first SETUP packet has arrived, Set the USBPHY_RX[ENVADJ] field back to default 0. Otherwise, USBPHY_RX[ENVADJ] field will remains as 2 unless a disconnect event occurs.
   - On "kUSB_DeviceEventDetach" event, Clear FORCE_FS (bit 21) of DEVCMDSTAT register to zero. Reset USBPHY_RX[ENVADJ] field back to default 0.

3.5 USB.4: In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer

Introduction

The LPC55S1x/LPC551x device family include a USB high-speed interface (USB1) that can operate in device mode at high-speed. The NBytes value represents the number of bytes that can be received in the buffer.

Problem

The LPC55S1x/LPC551x USB device controller writes extra bytes to the receive data buffer if the size of the transfer is not a multiple of 8 bytes since the USB device controller always writes 8 bytes. For example, if the transfer length is 1 bytes, 7 extra bytes will be written to the receive data buffer. If the transfer length is 7 bytes, 1 extra bytes will be written to the receive data buffer.

Work-around

Reserve an additional, intermediary buffer along with the buffer used by the application for USB data. After the USB data transfer into the intermediary buffer has been completed, use memcpy to move the data from the intermediary buffer into the application buffer, skipping the extraneous extra byte. This software work-around is implemented on the SDK software platform.

3.6 USB.5: In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated

Introduction

The LPC55S1x/LPC551x device family include a USB high-speed interface (USB1) that can operate in device mode at high-speed. The isochronous IN endpoint supports a MaxPacketSize of 1024 bytes.

Problem

When device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated.

Work-around

Restrict the isochronous IN endpoint MaxPacketSize to 1023 bytes in device descriptor.

3.7 USB.6: In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints

Introduction

The LPC55S1x/LPC551x device family include a USB high-speed interface which can operate in host mode. Up to three high-speed transactions are allowed in a single micro-frame to support high-bandwidth endpoints. This mode is enabled by setting the Mult (Multiple) field in the Proprietary Transfer Descriptor (PTD) and is used to indicate to the host controller the number of transactions that should be executed per micro-frame. The allowed bit settings are:

00b	Reserved. A zero in this field yields undefined results.
01b	One transaction to be issued for this endpoint per micro-frame.
10b	Two transactions to be issued for this endpoint per micro-frame.
11b	Three transactions to be issued for this endpoint per micro-frame.

Problem

For High-bandwidth mode, using multiple packets (MULT = 10b or 11b) in a frame causes unreliable operation. Only one transaction (MULT = 01b) can be issued per micro-frame.

Work-around

There is no software workaround. Only one transaction can be issued per micro-frame.

3.8 VBAT_DCDC.1: The minimum rise time of the power supply must be 2.6 ms or slower for Tamb = -40 C, and 0.5 ms or slower for Tamb = 0 C to +105 C

Introduction

The datasheet specifies no power-up requirements for the power supply on the VBAT_DCDC pin.

Problem

The device might not always start-up if the minimum rise time of the power supply ramp is 2.6 ms or faster for Tamb = -40 C, and 0.5 ms or faster for Tamb = 0 C to +105 C.

Work-around

None.

3.9 CAN-FD.1: Bus transaction abort could occur when CAN-FD peripheral is using secure alias.

Introduction

Unlike CM33, for other AHB masters (CAN-FD, USB-FS, DMA), the security level of transaction is fixed based on the level assigned for the master in SEC_AHB->MASTER_SEC_LEVEL register. So, if application needs to restrict the CAN-FD to secure, following steps are required:

- Set the security level of CAN-FD to secure-user (0x2) or secure privilege (0x3) in SEC_AHB->MASTER_SEC_LEVEL register.
- Assign secure-user or secure-privilege level for CAN-FD register space in SEC_AHB-> SEC_CTRL_AHB_PORT8_SLAVE1 Register.
- Assign secure-user or secure-privilege level for message RAM.

Example: If 16 KB of SRAM 2 (0x2000_C000) bank is used for CAN message RAM. Then set rules in SEC_AHB-> SEC_CTRL_RAM2_MEM_RULE0 register to secure-user (0x2) or secure privilege (0x3).

Problem

The shared memory used by CAN-FD controller and CPU should be accessible using secure alias with address bit 28 set (example 0x3000_C000). However, when CAN-FD makes a bus transaction using secure alias (address bit 28 set), the transaction is aborted.

Work-around

- When CPU is accessing the CAN-FD register or message RAM it should always use secure alias i.e., 0x3000_C000 for message RAM manipulation. .
- For any structure the CAN-FD peripheral uses to fetch or write, memory should be set to use 0x2000_C000 in order for bus transaction to work. CAN-FD software driver should set "Message RAM base address register (MRBA, offset 0x200)" with physical address of RAM instead of secure alias.

3.10 ROM.2: ROM API can’t be used correctly to update and read monotonic counter in CFPA.VENDOR_USAGE word

Introduction

Customer Field Programmable Area (CFPA) of Protected Flash Region (PFR) contains VENDOR_USAGE word. The lower 16-bits of the VENDOR_USAGE word implement a monotonic counter which should contain current value or higher value when new version of CFPA page is written. Upper 16-bits of the VENDOR_USAGE word should contain inverse value of aforesaid monotonic counter.

Problem

In the ROM, 16-bit monotonic counter is implemented by upper 16-bits of the VENDOR_USAGE word while lower 16-bits contain inverse value of monotonic counter i.e Monotonic Counter and its inverse value are swapped erroneously in the ROM. Due this error, ROM APIs do not access VENDOR_USAGE monotonic counter correctly.

Work-around

User should increment and store Monotonic Counter value in upper 16-bits of VENDOR_USAGE word while inverse value of the monotonic counter should be stored in the lower 16-bits of the VENDOR_USAGE word.

3.11 PUF SRAM.1: PUF SRAM needs to be reset during the startup of application to prevent high deep-sleep current consumption

Introduction

The LPC55S1x family offers SRAM PUF feature where the PUF provides a unique key per device. By default, the SRAM PUF block is disabled on the LPC551x devices.

Problem

The SRAM PUF block is enabled on the LPC551x devices resulting in higher deep-sleep current. The PUF SRAM block needs to be enabled and reset in order to achieve the deep-sleep current specification.

Work-around

On the LPC551x devices, following software workaround must be applied in the SystemInit function (SDK source file - "system_LPC55xx.c") to prevent high deep-sleep mode current consumption:

Enable the PUF Clock to access necessary registers.

Reset the PUF.

Enabled the PUF SRAM.

Disable the PUF clock.

```
/* Following code is to reset PUF to remove over consumption */
/* Enable PUF register clock to access register */
SYSCON->AHBCLKCTRLSET[2] = SYSCON_AHBCLKCTRL2_PUF_MASK;
/* Release PUF reset */
SYSCON->PRESETCTRLCLR[2] = SYSCON_PRESETCTRL2_PUF_RST_MASK ;
/* Enable PUF SRAM */
#define PUF_SRAM_CTRL_CFG (*((volatile uint32_t*)(0x4003B000u + 0x300u)))
#define PUF_SRAM_CTRL_INT_STATUS (*((volatile uint32_t*)(0x4003B000u + 0x3E0u)))
PUF_SRAM_CTRL_CFG |= 0x01 | 0x04;
/* Disable PUF register clock. */
// Delaying the line of code below until the PUF State Machine execution is completed:
// Shutting down the clock to early will prevent the state machine from reaching the end.
// => Wait for status bit in PUF Controller Registers before stop PUF clock.
while(!(PUF_SRAM_CTRL_INT_STATUS & 0x1));
SYSCON->AHBCLKCTRLCLR[2] = SYSCON_AHBCLKCTRL2_PUF_MASK ;
```

Remark: The SRAM PUF block should not be used on the LPC551x devices and is not guaranteed to function. This feature is only available on the LPC55S1x devices.

3.12 PLL.1: PLL LOCK bit is not reliable

Introduction

On the LPC55S1x/LPC551x devices, PLLxSTAT register of PLLs contains a LOCK detector status bit (bit 0 of PLLxSTAT register).

When the LOCK detector status bit is set to 1, the PLL is considered to be locked and stable.

The PLL LOCK signal is specified to work for Fref range from 100 kHz to 20 MHz. When the Fref is below 100 kHz or above 20 MHz, software should use a 6 ms time interval to insure the PLL will be stable.

Problem

On the LPC55S1x/LPC551x, the PLL status LOCK bit is not always reliable in the ranges specified and as a result, the PLL doesn´t initialize correctly.

Work-around

For Fref ≥ 20 MHz:

Software must wait at least (500us + 400/Fref) (Fref in Hz result in s) to ensure the PLL is stable.

For Fref < 20 MHz:

- If the PLL lock detector status bit is 1 before the wait time duration ((500us + 400/Fref)) is completed, the PLL is stable.
- If the PLL lock detector status bit is 0 but the wait time duration ((500us + 400/Fref)) is completed, the PLL is stable.

Software workaround is implemented in SDK 2.14 clock driver version 2.3.7.

Remark: This errata does not apply for spread spectrum mode.

3.13 ROM.3: Invalid TrustZone preset data structure can prevent ROM from applying specified debug settings

Introduction

On LPC55S1x devices, secure firmware images can optionally include TrustZone preset data that allows the core to configure TrustZone related registers while executing ROM code before jumping to the application code.

Problem

If the TrustZone preset data is invalid, in some cases this can prevent the ROM execution from proceeding further. Consequently, the application will not boot, and the ROM will not apply debug configurations (DBGEN, NIDEN, SPIDEN, etc.) settings.

Work-around

All application firmware images in this device should use secure boot. Also, the TrustZone preset data should be included within the authentication area to prevent the ROM from attempting to apply corrupted and/or maliciously modified TrustZone preset data settings.

4 AC/DC deviations detail

No known errata.

5 Errata notes detail

No known errata.

6 Revision history

Table 4. Revision history

Rev	Date	Description
2.0	20231219	Corrected Revision identifier field for PLL.1 in Table 1. Added Section 3.13 "ROM.3: Invalid TrustZone preset data structure can prevent ROM from applying specified debug settings"
1.9	20230524	Added Section 3.12 "PLL.1: PLL LOCK bit is not reliable"
1.8	20221219	Added Section 3.11 "PUF SRAM.1: PUF SRAM needs to be reset during the startup of application to prevent high deep-sleep current consumption"
1.7	20220513	Added ROM.2: Section 3.10 "ROM.2: ROM API can't be used correctly to update and read monotonic counter in CFPA.VENDOR_USAGE word"

Table 4. Revision history*...continued*

Rev	Date	Description
1.6	20211028	Added CAN-FD.1 note in Section 3.9 "CAN-FD.1: Bus transaction abort could occur when CAN-FD peripheral is using secure alias"
1.5	20210810	Added VBAT_DCDC.1: Section 3.8 "VBAT_DCDC.1: The minimum rise time of the power supply must be 2.6 ms or slower for Tamb = -40 C, and 0.5 ms or slower for Tamb = 0 C to +105 C"
1.4	20210423	Added USB.5, Section 3.6 "USB.5: In USB high-speed device mode, when device isochronous IN endpoint sends a packet of MaxPacketSize of 1024 bytes in response to IN token from host, the isochronous IN endpoint interrupt is not set and the endpoint command/status list entry for the isochronous IN endpoint is not updated"
		Added USB.6, Section 3.7 "USB.6: In USB high-speed host mode, only one transaction per micro-frame is allowed for isochronous IN endpoints"
1.3	20210225	Added USB.4, Section 3.5 USB.4: In USB high-speed device mode, device writes extra byte(s) to the buffer if the NBytes is not multiple of 8 for OUT transfer" Corrected Typo, Revision identifier as A for USB.3 in Table 1.
1.2	20201214	Includes Section 3.4 "USB.3: For the USB high-speed device controller, the detection handshaking fails when certain full-speed hubs are connected"
1.1	20200827	Adds Section 5.1 "CAN-FD peripheral cannot access secure alias address"
1.0	20191204	Initial version.

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 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.

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: 19 December 2023
Document identifier: ES_LPC55S1x/LPC551x