SPC564A74x, SPC564A80x Errata sheet

SPC564A74x, SPC564A80x devices errata JTAG_ID = 0x2AE02041

Introduction

This errata sheet describes all the functional and electrical problems known in the revision 3.0 of the SPC564A74x and SPC564A80x devices identified with the JTAG_ID = 0x2AE02041.

All the topics covered in this document refer to *RM0029 rev 7* and *SPC564A74B4, SPC564A74L7, SPC564A80B4, SPC564A80L7 datasheet rev 8* (see *Appendix B: Reference document*).

Device identification:

- Package device marking mask identifier: CA
- JTAG_ID = 0x2AE02041
- MIDR register:
  - MAJOR_MASK: 2
  - MINOR_MASK: 0

This errata sheet applies to SPC564A74x and SPC564A80x devices in accordance with *Table 1*.

Table 1. Device summary

Part number	Package
SPC564A74B4 SPC564A80B4	PBGA324
SPC564A74L7 SPC564A80L7	LQFP176

Contents

List of tables

1 Functional problems

1.1 ERR001312: FLASH: MCR[DONE] bit may be set before high voltage operation completes when executing a suspend sequence

Description:

The program and erase sequence of the flash may be suspended to allow read and program access to the flash core. An suspend operation is initiated by setting the Erase Suspend (ESUS) bit or Program Suspend (PSUS) bit in the flash Module Configuration Register (MCR). Setting a suspend bit causes the flash module to start the sequence which places it in the suspended state. The user must then wait until the MCR[DONE] bit is set before a read or program to the flash is initiated, as the high voltage operation needs to be complete to avoid errors.

However, during normal read to the same partition, following a suspend sequence, (setting MCR bit and waiting for MCR[DONE] bit to be set) can result in read fails that will return multiple bit ECC errors. The error is due to the MCR[DONE] bit being set before the internal high voltage operation is complete.

Workaround:

Because the MCR[DONE] flag can be set too soon, a delay needs to be inserted between setting the MCR[ESUS] or MCR[PSUS] and reading the same flash partition. The minimum duration of the delay should be 40us to guarantee correct operation. The Freescale flash programming driver includes this workaround.

1.2 ERR001397: Reaction Module: Register can set if RAER is asserted

Description:

A modulation can start if the RAER bit is asserted. The REACM_CHSR bit MODACT is asserted.

Workaround:

Enable the RAER interrupt on the module initialization, asserting the REACM_CHCR bit RAEREN. The interrupt service routine must disable the channel, setting the REACM_CHCR field CHEN to zero.

1.3 ERR002382: FLASH: Flash Array Integrity Check

Description:

The Flash Array Integrity Check (AIC) which may be enabled during the flash user test (UTest) mode does not return the expected UMn[MISR] values for some flash PFCRPn[RWSC] read wait state configurations. For PFCRPn[RWSC] values of 3-6, the UMn[MISR] signature computation during AIC does not include the data read from the very last address in the selected address sequence and thus the UMn[MISR] value is not as expected. For PFCRPn[RWSC] values of 7, the UMn[MISR] signature computation during AIC will not be correct as well.

Workaround:

The Flash Array Integrity Check is correct for PFCRPn[RWSC] values of 0-2. For PFCRPn[RWSC] values of 3-6, the expected UMn[MISR] values will not include the data read from the very last address and thus the value expected should be for the data read up to the 2nd-last address in the selected address sequence. For a PFCRPn[RWSC] value of 7, the Array Integrity Check should not be used at all.

1.4 ERR002740: ETPU2: Watchdog Status Register (WDSR) may fail to update on channel timeout

Description:

The Watchdog Status Register (WDSR) contains a single watchdog status bit for each of the 32 eTPU channels per engine. When this bit is set, it indicates that the corresponding channel encountered a watchdog timeout and was aborted. Under certain conditions the corresponding bit is not set due to a watchdog timeout, and therefore no indication is available as to which channel timed out. However, the global exception is indicated correctly on a per engine basis, and the correct exception is issued to the interrupt controller and may be serviced.

Workaround:

The application software should treat any watchdog event as a global eTPU exception and handle it in the eTPU global exception handler. Additionally, during the global exception handler the application should check the WDSR and clear any bits that may be set by writing '1' to that bit.

1.5 ERR003221: PMC: SRAM standby power low voltage detect circuit is not accurate

Description:

The power management controller (PMC) SRAM standby voltage low power detect circuit cannot reliably detect the brown-out condition if the standby supply is below 1.0 volts. The Status Register Brown Out Flag (PMC.SR[LVFSTBY]) bit may not be set during a brownout condition of the SRAM standby voltage or may be set even though no data has been lost.

Workaround:

The application software should not rely on the PMC.SR[LVFSTBY] bit to detect corrupted SRAM values.

1.6 ERR003285: SIU: MCU ID register package information not reliable on the calibration package

Description:

PKGCFG[0:1] are two external signals used in the CSP package to emulate the different production packages, the status of those signals is normally reflected in the PKG[0-4] field of the SIU MCU ID register (SIU_MIDR).
In this silicon revision there is only one configuration allowed: BGA324. Therefore in any calibration solution based on the CSP package, the PKG field will be 0b10100 (BGA324) regardless of the package.

Workaround:

The application SW must take in account that, during calibration, the information shown in the MIDR register might not reflect the package under emulation.

1.7 ERR003377: Pad Ring:Nexus pins may drive an unknown value immediately after power up but before the 1st clock edge

Description:

The Nexus Output pins (Message Data outputs 0:15 [MDO] and Message Start/End outputs 0:1 [MSEO]) may drive an unknown value (high or low) immediately after power up but before the 1st clock edge propagates through the device (instead of being weakly pulled low). This may cause high currents if the pins are tied directly to a supply/ground or any low resistance driver (when used as a general purpose input [GPI] in the application).

Workaround:

1. Do not tie the Nexus output pins directly to ground or a power supply.
2. If these pins are used as GPI, limit the current to the ability of the regulator supply to guarantee correct start up of the power supply. Each pin may draw upwards of 150mA.

If not used, the pins may be left unconnected.

1.8 ERR003378: EQADC: Pull devices on differential pins may be enabled for a short period of time during and just after POR

Description:

The programmable pull devices (up and down) on the analog differential inputs of the eQADC may randomly be enabled during the internal Power On Reset (POR) and until the 1st clock edge propagates through the device. After the first clock edge, the pull resistors will be disabled until software enables them.

Workaround:

Protect any external devices connected to the differential analog inputs. The worst case condition is with a 1.4K ohm resistor to VDDA (5K pull-up enabled) or VSSA (5K pull-down enabled). This may also cause temporary additional current requirements on the VDDA supply of each eQADC module, up to 15 mA on each eQADC if both the pull up and pull down resistors are enabled simultaneously on all of the differential analog pins.

1.9 ERR003407: FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1

Description:

FlexCAN does not transmit an expected message when the same node detects an incoming Remote Request message asking for any remote answer.

The issue happens when two specific conditions occur:

1. The Message Buffer (MB) configured for remote answer (with code "a") is the last MB. The last MB is specified by Maximum MB field in the Module Configuration Register (MCR[MAXMB]).
2. The incoming Remote Request message does not match its ID against the last MB ID.

While an incoming Remote Request message is being received, the FlexCAN also scans the transmit (Tx) MBs to select the one with the higher priority for the next bus arbitration. It

is expected that by the Intermission field it ends up with a selected candidate (winner). The coincidence of conditions (1) and (2) above creates an internal corner case that cancels the Tx winner and therefore no message will be selected for transmission in the next frame. This gives the appearance that the FlexCAN transmitter is stalled or "stops transmitting".

The problem can be detectable only if the message traffic ceases and the CAN bus enters into Idle state after the described sequence of events.

There is NO ISSUE if any of the conditions below holds:

a) The incoming message matches the remote answer MB with code "a".

b) The MB configured as remote answer with code "a" is not the last one.

c) Any MB (despite of being Tx or Rx) is reconfigured (by writing its CS field) just after the Intermission field.

d) A new incoming message sent by any external node starts just after the Intermission field.

Workaround:

Do not configure the last MB as a Remote Answer (with code "a").

1.10 ERR004480: eQADC: Differential conversions with 4x gain may halt command processing

Description:

If the four times amplifier is enabled for a differential analog-to-digital conversion in the Enhanced Queued Analog to Digital Converter (eQADC) and the ADC clock prescaler is set to divide by 12 or greater, then the ADC will stop processing commands if a conversion command is executed immediately after a differential, gain 4x conversion.

Workaround:

1. Do not use a prescaler divide factor greater than or equal to 12 (11 can be used on devices that support odd prescalers).
2. Insert a dummy write command to any internal ADC register after every 4x conversion command.

Note 1: If the command FIFO preemption feature is used and it is possible to preempt a FIFO which contains the 4x conversion + dummy write workaround, then the preempting command FIFO must be loaded FIRST with a dummy write command and then the desired preempting conversion command in order to avoid the possibility of following a 4x conversion command with another conversion command in the same ADC.

Note 2: The level sensitive triggers (when in Low/High Level Gated External Trigger, Single/Continuous Scan modes) can interrupt the command sequence at any point in time, potentially breaking the safe sequence 4x conversion command -> dummy write command.

Note 3: When using an odd prescaler (ADCx_CLK_ODD = 1), the duty cycle setting (ADCxCLK_DTY) must be kept at the default setting of 0.

1.11 ERR004532: REACM: OCDF flag is set during hold-off time

Description:

The Reaction Modules (REACM) Open Circuit Detection Flag (OCDF) indicates that an Open Circuit was detected during the holdoff time phase of a threshold-holdoff modulation. This flag indication is incorrect because the OCDF flag is meant to indicates ON periods larger than a configured value. In this case, OCDF is setting during the OFF phase of the modulation.

Workaround:

When a modulation sequence has a phase that uses Threshold-holdoff modulation mode, the OCDF flag cannot be used reliably. In is this case, one can write the Modulation Range Pulse Width Register [REACM_RANGEPWD] register to '0x000' which will disable the check of "Open Circuit" functionality.

1.12 ERR005037: CRC: CRC-32 (Ethernet) and CRC-16 (CCITT) operation do not match industry standards.

Description:

CRC-32 (Ethernet) and CRC-16 (CCITT) do not calculate CRCs according to industry standards. For CRC-32, the CRC engine expects the CRC Input to be byte swapped. For CRC-16, the CRC engine expects the CRC Input to be bit reversed.

Workaround:

The input data has to be bit reversed and entered into the CRC Input Register (CRC_INP) for CRC-16 calculations and byte reversed for CRC-32 calculations to get the correct CRC signature.

For CRC-32, given input data of 0xABCDEF98, the user software must enter 0x98EFCDAB in CRC_INP register to get the correct result for CRC-32. Also, the Inversion (INV) and Swap selection (SWAP) bits of the CRC Configuration Register (CRC_CFG) are to be set to 1 for CRC-32 polynomial calculations along with the byte swaps.

For CRC-16, given input data of 0xABDEF98, the user software must enter 0x19F7B3D5 into the CRC_INP register. INV and SWAP bits are programmed to zero for CRC-16 operation.

1.13 ERR005086: eQADC: unexpected result may be pushed when Immediate Conversion Command is enabled

Description:

In the enhanced Queued Analog to Digital Converter (eQADC), when the Immediate Conversion Command is enabled (ICEAn=1) in the eQADC_MCR (Module Configuration Register), if a conversion from Command First-In-First Out (CFIFO0, conv0) is requested concurrently with the end-of-conversion from another, lower priority conversion (convx), the result of the convx may be lost or duplicated causing an unexpected number of results in the FIFO (too few or too many).

Workaround:

Workaround 1: Do not use the abort feature (ICEAn=0).

Workaround 2: Arrange the timing of the CFIFO0 trigger such that it does not assert the trigger at the end of another, lower priority conversion.

Workaround 3: Detect the extra or missing conversion result by checking the EQADC_CFTCRx (EQADC CFIFO Transfer Counter Register x). This register records how many commands were issued, so it can be used to check that the expected number of results have been received.

1.14 ERR005640: ETPU2: Watchdog timeout may fail in busy length mode

Description:

When the Enhanced Time Processing Unit (eTPU) watchdog is programmed for busy length mode (eTPU Watchdog Timer Register (ETPU_WDTR) Watchdog Mode field (WDM) = 3), a watchdog timeout will not be detected if all of the conditions below are met:

1. The watchdog timeout occurs at the time slot transition, at the first instruction of a thread, or at the thread gap. (a thread gap is a 1 microcycle period between threads that service the same channel).
2. The thread has only one instruction.
3. The eTPU goes idle right after the timed-out thread, or after consecutive single-instruction threads.

Workaround:

Insert a NOP instruction in threads which have only one instruction.

1.15 ERR005642: ETPU2: Limitations of forced instructions executed via the debug interface

Description:

The following limitations apply to forced instructions executed through the Nexus debug interface on the Enhanced Time Processing Unit (ETPU):

1. When a branch or dispatch call instruction with the pipeline flush enabled (field FLS=0) is forced (through the debug port), the Return Address Register (RAR) is updated with the current program counter (PC) value, instead of PC value + 1.
2. The Channel Interrupt and Data Transfer Requests (CIRC) instruction field is not operational.

Workaround:

Workaround for limitation #1 (branch or dispatch call instruction):

Increment the PC value stored in the RAR by executing a forced Arithmetic Logic Unit (ALU) instruction after the execution of the branch or dispatch call instruction.

Workaround for limitation #2 (CIRC):

To force an interrupt or DMA request from the debugger:

1. Program a Shared Code Memory (SCM) location with an instruction that issues the interrupt and/or DMA request. Note: Save the original value at the SCM location.
2. Save the address of the next instruction to be executed.
3. Force a jump with flush to the instruction position.
4. Single-step the execution.
5. Restore the saved value to the SCM location (saved in step 1).
6. Force a jump with flush to the address of the next instruction to be executed (saved in step 2).

Note: This workaround cannot be executed when the eTPU is in HALT_IDLE state.

1.16 ERR006026: DSPI: Incorrect SPI Frame Generated in Combined Serial Interface Configuration

Description:

In the Combined Serial Interface (CSI) configuration of the Deserial Serial Peripheral Interface (DSPI) where data frames are periodically being sent (Deserial Serial Interface, DSI), a Serial Peripheral Interface (SPI) frame may be transmitted with incorrect framing.

The incorrect frame may occur in this configuration if the user application writes SPI data to the DSPI Push TX FIFO Register (DSPI_PUSHR) during the last two peripheral clock cycles of the Delay-after-Transfer (DT) phase. In this case, the SPI frame is corrupted.

Workaround:

Workaround 1: Perform SPI FIFO writes after halting the DSPI.

To prevent writing to the FIFO during the last two clock cycles of DT, perform the following steps every time a SPI frame is required to be transmitted:

Step 1: Halt the DSPI by setting the HALT control bit in the Module Configuration Register (DSPI_MCR[HALT]).

Step 2: Poll the Status Register's Transmit and Receive Status bit (DSPI_SR[TXRXS]) to ensure the DSPI has entered the HALT state and completed any in-progress transmission. Alternatively, if continuous polling is undesirable in the application, wait for a fixed time interval such as 35 baud clocks to ensure completion of any in-progress transmission and then check once for DSPI_SR[TXRXS].

Step 3: Perform the write to DSPI_PUSHR for the SPI frame.

Step 4: Clear bit DSPI_MCR[HALT] to bring the DSPI out of the HALT state and return to normal operation.

Workaround 2: Do not use the CSI configuration. Use the DSPI in either DSI-only mode or SPI-only mode.

Workaround 3: Use the DSPI's Transfer Complete Flag (TCF) interrupt to reduce worst-case wait time of Workaround 1

Step 1: When a SPI frame is required to be sent, halt the DSPI as in Step 1 of Workaround 1 above.

Step 2: Enable the TCF interrupt by setting the DSPI DMA/Interrupt Request Select and Enable Register's Transmission Complete Request Enable bit (DSPI_RSER[TCF_RE])

Step 3: In the TCF interrupt service routine, clear the interrupt status (DSPI_SR[TCF]) and the interrupt request enable (DSPI_RSER[TCF_RE]). Confirm that DSPI is halted by checking DSPI_SR[TXRXS] and then write data to DSPI_PUSHR for the SPI frame. Finally, clear bit DSPI_MCR[HALT] to bring the DSPI out of the HALT state and return to normal operation.

1.17 ERR006726: NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled

Description:

The Nexus auxiliary message clock (MCKO) may be gated one clock period early when the MCKO frequency is programmed as SYS_CLK/8 in the Nexus Port Controller Port Configuration Register (NPC_PCR[MCKO_DIV]=111) and the MCKO gating function is enabled (NPC_PCR[MCKO_GT]=1). In this case, the last MCKO received by the tool prior to the gating will correspond to the END_MESSAGE state. The tool will not receive an MCKO to indicate the transition to the IDLE state, even though the NPC will transition to the IDLE state internally. Upon re-enabling of MCKO, the first MCKO edge will drive the Message Start/End Output (MSEO=11) and move the tool's state to IDLE.

Workaround:

Expect to receive the MCKO edge corresponding to the IDLE state upon re-enabling of MCKO after MCKO has been gated.

1.18 ERR006967: eDMA: Possible misbehavior of a preempted channel when using continuous link mode

Description:

When using Direct Memory Access (DMA) continuous link mode Control Register Continuous Link Mode (DMA_CR[CLM]) = 1) with a high priority channel linking to itself, if the high priority channel preempts a lower priority channel on the cycle before its last read/write sequence, the counters for the preempted channel (the lower priority channel) are corrupted. When the preempted channel is restored, it continues to transfer data past its "done" point (that is the byte transfer counter wraps past zero and it transfers more data than indicated by the byte transfer count (NBYTES)) instead of performing a single read/write sequence and retiring.

The preempting channel (the higher priority channel) will execute as expected.

Workaround:

Disable continuous link mode (DMA_CR[CLM]=0) if a high priority channel is using minor loop channel linking to itself and preemption is enabled. The second activation of the preempting channel will experience the normal startup latency (one read/write sequence + startup) instead of the shortened latency (startup only) provided by continuous link mode.

1.19 ERR007322: FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state

Description:

Under normal operation, when FlexCAN enters in Bus Off state, a Bus Off Interrupt is issued to the CPU if the Bus Off Mask bit (CTRL[BOFF_MSK]) in the Control Register is set. In consequence, the CPU services the interrupt and clears the ESR[BOFF_INT] flag in the Error and Status Register to turn off the Bus Off Interrupt.
In continuation, if the CPU performs a soft reset after servicing the bus off interrupt request, by either requesting a global soft reset or by asserting the MCR[SOFT_RST] bit in the Module Configuration Register, once MCR[SOFT_RST] bit transitions from 1 to 0 to acknowledge the soft reset completion, the ESR[BOFF_INT] flag (and therefore the Bus Off Interrupt) is re-asserted.
The defect under consideration is the erroneous value of Bus Off flag after soft reset under the scenario described in the previous paragraph.

The Fault Confinement State (ESR[FLT_CONF] bit field in the Error and Status Register) changes from 0b11 to 0b00 by the soft reset, but gets back to 0b11 again for a short period, resuming after certain time to the expected Error Active state (0b00). However, this late correct state does not reflect the correct ESR[BOFF_INT] flag which stays in a wrong value and in consequence may trigger a new interrupt service.

Workaround:

To prevent the occurrence of the erroneous Bus Off flag (and eventual Bus Off Interrupt) the following soft reset procedure must be used:

1. Clear CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).
2. Set MCR[SOFT_RST] bit in the Module Configuration Register.
3. Poll MCR[SOFT_RST] bit in the Module Configuration Register until this bit is cleared.
4. Wait for 4 peripheral clocks.
5. Poll ESR[FLTCONF] bit in the Error and Status Register until this field is equal to 0b00.
6. Write "1" to clear the ESR[BOFF_INT] bit in the Error and Status Register.
7. Set CTRL[BOFF_MSK] bit in the Control Register (optional step in case the Bus Off Interrupt is enabled).

1.20 ERR007352: DSPI: reserved bits in slave CTAR are writable

Description:

When the Deserial/Serial Peripheral Interface (DSPI) module is operating in slave mode (the Master [MSTR] bit of the DSPI Module Configuration Register [DSPIx_MCR] is cleared), bits 10 to 31 (31 = least significant bit) of the Clock and Transfer Attributes Registers (DSPIx_CTARx) should be read only (and always read 0). However, these bits are writable, but setting any of these bits to a 1 does not change the operation of the module.

Workaround:

There are two possible workarounds.

Workaround 1: Always write zeros to the reserved bits of the DSPIx_CTARn_SLAVE (when operating in slave mode).

Workaround 2: Mask the reserved bits of DSPIx_CTARn_SLAVE when reading the register in slave mode.

Appendix A Defect across silicon version

Table 2. Defects across silicon version

Defect ID	Title	Cut 2.0	Cut 3.0
ERR001312	FLASH: MCR[DONE] bit may be set before high voltage operation completes when executing a suspend sequence	x	x
ERR001397	Reaction Module: Register can set if RAER is asserted	x	x
ERR002382	FLASH: Flash Array Integrity Check	x	x
ERR002740	ETPU2: Watchdog Status Register (WDSR) may fail to update on channel timeout	x	x
ERR003221	PMC: SRAM standby power low voltage detect circuit is not accurate	x	x
ERR003285	SIU: MCU ID register package information not reliable on the calibration package	x	x
ERR003377	Pad Ring:Nexus pins may drive an unknown value immediately after power up but before the 1st clock edge	x	x
ERR003378	EQADC: Pull devices on differential pins may be enabled for a short period of time during and just after POR	x	x
ERR003407	FlexCAN: CAN Transmitter Stall in case of no Remote Frame in response to Tx packet with RTR=1	x	x
ERR003659	FLASH: Resuming after a suspend during an Erase may prevent the erase from completing.	x	
ERR004480	eQADC: Differential conversions with 4x gain may halt command processing	x	x
ERR004532	REACM: OCDF flag is set during hold-off time	x	x
ERR005037	CRC: CRC-32 (Ethernet) and CRC-16 (CCITT) operation do not match industry standards.	x	x
ERR005086	eQADC: unexpected result may be pushed when Immediate Conversion Command is enabled	x	x
ERR005640	ETPU2: Watchdog timeout may fail in busy length mode	x	x
ERR005642	ETPU2: Limitations of forced instructions executed via the debug interface	x	x
ERR006026	DSPI: Incorrect SPI Frame Generated in Combined Serial Interface Configuration	x	x
ERR006726	NPC: MCKO clock may be gated one clock period early when MCKO frequency is programmed as SYS_CLK/8.and gating is enabled	x	x
ERR006967	eDMA: Possible misbehavior of a preempted channel when using continuous link mode	x	x
ERR007322	FlexCAN: Bus Off Interrupt bit is erroneously asserted when soft reset is performed while FlexCAN is in Bus Off state	x	x
ERR007352	DSPI: reserved bits in slave CTAR are writable	x	x

Appendix B Reference document

- *SPC564A74xx, SPC564A80xx 32-bit MCU family built on the embedded Power Architecture*® (RM0029, Doc ID 15177)
- *32-bit MCU family built on the embedded Power Architecture*® (SPC564A74B4, SPC564A74L7, SPC564A80B4, SPC564804L7datasheet, Doc ID 15399)

Revision history

Table 3. Document revision history

Date	Revision	Changes
11-Jul-2014	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com