STM32WBA2xxx device errata

Applicability

This document applies to the part numbers of STM32WBA2xxx devices and the device variants as stated in this page.

It gives a summary and a description of the device errata, with respect to the device datasheet and reference manual RM0521.

Deviation of the real device behavior from the intended device behavior is considered to be a device limitation. Deviation of the description in the reference manual or the datasheet from the intended device behavior is considered to be a documentation erratum. The term *"errata"* applies both to limitations and documentation errata.

Table 1. Device summary

Reference	Part numbers
STM32WBA2xxx	STM32WBA23KE, STM32WBA23CE, STM32WBA25HE, STM32WBA25CE

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32WBA2xxx	A	0x1000

1. *Refer to the device datasheet for how to identify this code on different types of package.*
2. *REV_ID[15:0] bitfield of DBGMCU_IDCODE register.*

1 Summary of device errata

The following table gives a quick reference to the STM32WBA2xxx device limitations and their status:

A	Limitation applicable. Workaround available
N	Limitation applicable. No workaround available
P	Limitation applicable. Partial workaround available
-	Limitation not applicable.
A/-, **P/-**, or **N/-**	Limitation applicable, then fixed on same silicon revision (refer to the corresponding erratum for details).

Applicability of a workaround may depend on specific conditions of target application. Adoption of a workaround may cause restrictions to target application. Workaround for a limitation is deemed partial if it only reduces the rate of occurrence and/or consequences of the limitation, or if it is fully effective for only a subset of instances on the device or in only a subset of operating modes, of the function concerned.

Table 3. Summary of device limitations

Function	Section	Limitation	Status
			Rev. A
Core	2.1.1	Access permission faults are prioritized over unaligned device memory faults	N
System	2.2.7	Higher power consumption for Stop 2 and Stop 3 modes	N
	2.2.8	LSE in bypass mode may not be functional	N
	2.2.9	Bootloader inaccessible due to incorrect factory programming of option byte	P/-
TIM	2.3.1	Unexpected PWM output when using ocref_clr	N
LPTIM	2.4.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A
	2.4.2	ARRM and CPMxOK flags are not set when APB clock is slower than kernel clock	A
	2.4.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N
RTC	2.5.1	Alarm flag may be repeatedly set when the core is stopped in debug	N
	2.5.2	RTC calendar read may return transient incorrect value when BYPSHAD = 0	A
I2C	2.6.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P
	2.6.2	Spurious bus error detection in controller mode	A
USART	2.7.1	Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1	A
	2.7.2	Received data may be corrupted upon clearing the ABREN bit	A
	2.7.3	Noise error flag set while ONEBIT is set	N
LPUART	2.8.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P
SPI	2.9.1	RDY output failure at high serial clock frequency	N
USB	2.10.1	Buffer description table update completes after CTR interrupt triggers	A

2 Description of device errata

The following sections describe the errata of the applicable devices and provide a workaround where available. They are grouped by device functions.

The applicable devices are based on Arm® Cortex® core.

arm

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

2.1 Core

Reference manual and errata notice for the Arm® Cortex®-M33 core revision r0p1 is available from http://infocenter.arm.com.

2.1.1 Access permission faults are prioritized over unaligned device memory faults

Description

A load or store which causes an unaligned access to device memory results in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault is prioritized over the UsageFault.

The failure occurs when the MPU is enabled and:

- A load/store access occurs to an address which is not aligned to the data type specified in the instruction.
- The memory access hits one region only.
- The region attributes (specified in the MAIR register) mark the location as device memory.
- The region access permissions prevent the access (that is, unprivileged or write not allowed).

The MemManage fault caused by the access permission violation is prioritized over the UNALIGNED UsageFault exception because of the memory attributes.

Workaround

None. However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.

2.2 System

2.2.1 Incorrect definition of HSERDY flag in reference manual

Description

In reference manual versions prior to RM0486 revision 4, the HSERDY flag is incorrectly described.

The correct definition is that HSERDY is set once the initial clock cycles from the HSE oscillator are detected after the HSEON bit is set. This does not guarantee clock stability.

The stability of the HSE clock depends on the external oscillator. Proper HSE operation and frequency stability of the external clock must be verified by referring to the actual startup time of the external HSE oscillator, as specified in the datasheet.

Workaround

None.

2.2.2 Incorrect definition of the RTCPRE divider formula in the reference manual

Description

In reference manual versions prior to RM0486 revision 4, the description of the RTC oscillator divider RTCPRE[5:0] and the associated formula are incorrect. The correct information is as follows:

The division ratio is non-linear and is defined as: division factor = hse_ck / (v + 1)², where v is the value of RTCPRE[5:0]:

- 0x00: hse_ck divided by 1
- 0x01: hse_ck divided by 4
- 0x02: hse_ck divided by 9
- 0x03: hse_ck divided by 16
- ...
- 0x3F: hse_ck divided by 4096

Workaround

None.

2.2.3 Incorrect description of the OTP16 BootROM_CONFIG_7 in reference manual

Description

In reference manual versions prior to RM0486 revision 4, the detailed descriptions of bit 0: disable_traces, bit 1: disable_hse_freq_detect, and bit 2: disable_hse_bypass_detect in OTP16 were incorrect and used inverted logic.

Workaround

None.

2.2.4 Incorrect HSE minimum values in OTP16 BootROM_CONFIG_7 and in RCC section of the reference manual

Description

In reference manual versions prior to RM0486 revision 4, all references to HSE values of 8, 10, 12, or 14 are incorrect.

The minimum HSE value on STM32N6 is 16 MHz.

Workaround

None.

2.2.5 Missing default value for rng_htcr_value in OPT19 BootROM_CONFIG_10

Description

In reference manual versions prior to RM0486 revision 4, the default value of the rng_htcr_value field in OPT19 – BootROM_CONFIG_10 is not documented correctly.

The correct default value is: rng_htcr_value = 6. This value corresponds to an rng_htcr register value of 0xAAC7 for the associated bitfield configuration.

This issue affects documentation only. The silicon behavior is correct.

Workaround

None.

2.2.6 Incorrect operational status of PVM monitoring for V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, USB33, and V_{DDCORE} in reference manual or datasheet

Description

In reference manual versions prior to applicable revision, the Programmable Voltage Monitor (PVM) monitoring for the following supplies is incorrectly described as operational:

- V_{DDIO2}
- V_{DDIO3}
- V_{DDIO4}
- V_{DDIO5}
- USB33 VAM
- V_{DDCORE}

In these versions of the reference manual, the PVM functionality is indicated as available and functional for the above power domains. However, PVM monitoring is not operational for these supplies.

PVM events, flags, or interrupts related to the listed voltage domains are not generated. Do not use software that relies on PVM status for these supplies for voltage supervision or safety monitoring.

Workaround

None.

2.2.7 Higher power consumption for Stop 2 and Stop 3 modes

Description

Using either Stop 2 or Stop 3 mode can result in higher power consumption of approximately 50 µA in typical conditions, and 500 µA at high temperature.

Workaround

Do not use Stop 2 nor Stop 3 mode. Only use Stop 1 mode.

2.2.8 LSE in bypass mode may not be functional

Description

LSE in bypass mode fails on around 1% of produced parts. The failure rate increases with colder temperature.

Workaround

None.

2.2.9 Bootloader inaccessible due to incorrect factory programming of option byte

This issue affects STM32WBA2xxx devices with date code earlier than week 14 of 2026. Devices manufactured from week 14 of 2026 (date code 614) onward are not affected.

Description

On some devices, the NSBOOTADD1 option byte is incorrectly programmed during production. As a result, the bootloader address is not correctly defined, and the device cannot boot from the bootloader. Consequently, the bootloader cannot be used to program the device.

Workaround

If JTAG/SW is accessible, follow these steps to reprogram the NSBOOTADD1 option byte at address `0x4002 2048` so that it points to the system bootloader address `0x0BF8 5000`:

1. Power the device on.
2. Connect to the STM32WBA2 device via JTAG/SW under reset.

3. Reprogram the user option byte NSBOOTADD1 to the address `0x0BF8 5000`.
4. Power the device off.

2.3 TIM

2.3.1 Unexpected PWM output when using ocref_clr

Description

In combined PWM mode 1, asymmetric PWM mode 1, or asymmetric PWM mode 2, using ocref_clr can cause the tim_ocxrefc output to be unexpectedly re-enabled or disabled. This behavior depends on the timing of when ocref_clr is activated and deactivated.

Workaround

None.

2.4 LPTIM

2.4.1 Device may remain stuck in LPTIM interrupt when entering Stop mode

Description

This limitation occurs when disabling the low-power timer (LPTIM).

When the user application clears the ENABLE bit in the LPTIM_CR register within a small time window around one LPTIM interrupt occurrence, then the LPTIM interrupt signal used to wake up the device from Stop mode may be frozen in active state. Consequently, when trying to enter Stop mode, this limitation prevents the device from entering low-power mode and the firmware remains stuck in the LPTIM interrupt routine.

This limitation applies to all Stop modes and to all instances of the LPTIM. Note that the occurrence of this issue is very low.

Workaround

In order to disable a low power timer (LPTIMx) peripheral, do not clear its ENABLE bit in its respective LPTIM_CR register. Instead, reset the whole LPTIMx peripheral via the RCC controller by setting and resetting its respective LPTIMxRST bit in the relevant RCC register.

2.4.2 ARRM and CPMxOK flags are not set when APB clock is slower than kernel clock

Description

When LPTIM is configured in one shot mode and APB clock is lower than kernel clock, there is a chance that ARRM and CPMxOK flags are not set at the end of the counting cycle defined by the repetition value REP[7:0]. This issue can only occur when the repetition counter is configured with an odd repetition value.

Workaround

To avoid this issue, the following formula must be respected:

{ARR, CCR} ≥ KER_CLK / (2* APB_CLK),

where APB_CLK is the LPTIM APB clock frequency, and KER_CLK is the LPTIM kernel clock frequency. ARR and CCR are expressed in decimal value.

Example: The following example illustrates a configuration where the issue can occur:

- APB clock source (MSI) = 1 MHz, kernel clock source (HSI) = 16 MHz
- The repetition counter is set with REP[7:0] = 0x3 (odd value)

The above example is subject to issues, unless the user respects:

{CCR, ARR} ≥ 16 MHz / (2 * 1 MHz)

→ ARR must be ≥ 8 and CCR must be ≥ 8

Note: REP set to 0x3 means that effective repetition is REP+1 (= 4) but the user must consider the parity of the value loaded in the LPTIM_RCR register (=3, odd) to assess the risk of issue.

2.4.3 Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register

Description

When any interrupt bit of the LPTIM_DIER register is modified, the corresponding flag of the LPTIM_ISR register is cleared by hardware.

Workaround

None.

2.5 RTC

2.5.1 Alarm flag may be repeatedly set when the core is stopped in debug

Description

When the core is stopped in debug mode, the clock is supplied to subsecond RTC alarm downcounter even when the device is configured to stop the RTC in debug.

As a consequence, when the subsecond counter is used for alarm condition (the MASKSS[3:0] bitfield of the RTC_ALRMASSR and/or RTC_ALRMBSSR register set to a non-zero value) and the alarm condition is met just before entering a breakpoint or printf, the ALRAF and/or ALRBF flag of the RTC_SR register is repeatedly set by hardware during the breakpoint or printf, which makes any attempt to clear the flag(s) ineffective.

Workaround

None.

2.5.2 RTC calendar read may return transient incorrect value when BYPSHAD = 0

Description

When the BYPSHAD control bit in the RTC_CR register is cleared, reading the RTC calendar registers (RTC_SSR, RTC_TR, and RTC_DR) may seldom produce incorrect values. This issue occurs because internal timing can cause the asynchronous RTC calendar registers to be copied into their shadow registers during a transition of the asynchronous registers.

Since this copying process occurs every RTC kernel clock cycle, any erroneous value persists for only one RTC kernel clock cycle. The likelihood of this failure is very low and depends on several factors, including:

- RTC software configuration, such as the RTC kernel clock source, asynchronous prescaler factor, and calibration settings
- APB clock frequency
- Operating voltage and temperature
- The timing of the register read operation

Additionally, process variations may cause timing differences between samples, which can also influence the probability of this failure.

Workaround

To avoid this condition and ensure consistent calendar readings:

- Set the BYPSHAD control bit to 1 in the RTC_CR register to bypass shadow registers.
- Read the RTC calendar registers twice.
- Compare the two readings. If the values differ, repeat the read and compare sequence until two consecutive readings match.

This software method ensures a reliable calendar value under all operating conditions. For further details, see section *Reading the calendar – When the BYPSHAD control bit is set in the RTC_CR register (bypass shadow registers)* in the reference manual.

2.6 I2C

2.6.1 Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period

Description

The I^2C-bus specification and user manual specify a minimum data setup time ($t_{SU;DAT}$) as:

- 250 ns in Standard mode
- 100 ns in Fast mode
- 50 ns in Fast mode Plus

The device does not correctly sample the I^2C-bus SDA line when $t_{SU;DAT}$ is smaller than one I2C kernel clock (I^2C-bus peripheral clock) period: the previous SDA value is sampled instead of the current one. This can result in a wrong receipt of target address, data byte, or acknowledge bit.

Workaround

Increase the I2C kernel clock frequency to get I2C kernel clock period within the transmitter minimum data setup time. Alternatively, increase transmitter's minimum data setup time. If the transmitter setup time minimum value corresponds to the minimum value provided in the I^2C-bus standard, the minimum I2CCLK frequencies are as follows:

- In Standard mode, if the transmitter minimum setup time is 250 ns, the I2CCLK frequency must be at least 4 MHz.
- In Fast mode, if the transmitter minimum setup time is 100 ns, the I2CCLK frequency must be at least 10 MHz.
- In Fast-mode Plus, if the transmitter minimum setup time is 50 ns, the I2CCLK frequency must be at least 20 MHz.

2.6.2 Spurious bus error detection in controller mode

Description

In controller mode, a bus error can be detected spuriously, with the consequence of setting the BERR flag of the I2C_SR register and generating bus error interrupt if such interrupt is enabled. Detection of bus error has no effect on the I^2C-bus transfer in controller mode and any such transfer continues normally.

Workaround

If a bus error interrupt is generated in controller mode, the BERR flag must be cleared by software. No other action is required and the ongoing transfer can be handled normally.

2.7 USART

2.7.1 Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1

Description

The SPI slave receiver device receives wrong data when all the following conditions are met:

- The USART is used in SPI master transmitter mode
- The autonomous mode is used
- The CPOL bit of the USART_CR2 register is set

Workaround

When the autonomous mode is used, do not set the CPOL bit in USART_CR2.

2.7.2 Received data may be corrupted upon clearing the ABREN bit

Description

The USART receiver may miss data or receive corrupted data when the auto baud rate feature is disabled by software (ABREN bit cleared in the USART_CR2 register) after an auto baud rate detection, while a reception is ongoing.

Workaround

Do not clear the ABREN bit.

2.7.3 Noise error flag set while ONEBIT is set

Description

When the ONEBIT bit is set in the USART_CR3 register (one sample bit method is used), the noise error (NE) flag must remain cleared. Instead, this flag is set upon noise detection on the START bit.

Workaround

None.

Note: Having noise on the START bit is contradictory with the fact that the one sample bit method is used in a noise free environment.

2.8 LPUART

2.8.1 Possible LPUART transmitter issue when using low BRR[15:0] value

Description

The LPUART transmitter bit length sequence is not reset between consecutive bytes, which could result in a jitter that cannot be handled by the receiver device. As a result, depending on the receiver device bit sampling sequence, a desynchronization between the LPUART transmitter and the receiver device may occur resulting in data corruption on the receiver side.

This happens when the ratio between the LPUART kernel clock and the baud rate programmed in the LPUART_BRR register (BRR[15:0]) is not an integer, and is in the three to four range. A typical example is when the 32.768 kHz clock is used as kernel clock and the baud rate is equal to 9600 baud, resulting in a ratio of 3.41.

Workaround

Apply one of the following measures:

- On the transmitter side, increase the ratio between the LPUART kernel clock and the baud rate. To do so:
  - Increase the LPUART kernel clock frequency, or
  - Decrease the baud rate.
- On the receiver side, generate the baud rate by using a higher frequency and applying oversampling techniques if supported.

2.9 SPI

2.9.1 RDY output failure at high serial clock frequency

Description

When acting as slave with RDY alternate function enabled through setting the RDIOM bit of the SPI_CFG2 register, the device may fail to indicate its *Not ready* status in time through the RDY output signal to suspend communication. This may then lead to data overrun and/or underrun on the device side. The failure occurs when the serial clock frequency exceeds:

- Twice the APB clock frequency, with data sizes from 8 to 15 bits
- Six times the APB clock frequency, with data sizes from 16 to 23 bits
- Fourteen times the APB clock frequency, with data sizes from 24 to 32 bits

Workaround

None.

2.10 USB

2.10.1 Buffer description table update completes after CTR interrupt triggers

Description

During OUT transfers, the correct transfer interrupt (CTR) is triggered a little before the last USB SRAM accesses have completed. If the software responds quickly to the interrupt, the full buffer contents may not be correct.

Workaround

Software should ensure that a small delay is included before accessing the SRAM contents. This delay should be 800 ns in Full Speed mode and 6.4 µs in Low Speed mode.

Important security notice

The STMicroelectronics group of companies (ST) places a high value on product security, which is why the ST product(s) identified in this documentation may be certified by various security certification bodies and/or may implement our own security measures as set forth herein. However, no level of security certification and/or built-in security measures can guarantee that ST products are resistant to all forms of attacks. As such, it is the responsibility of each of ST's customers to determine if the level of security provided in an ST product meets the customer needs both in relation to the ST product alone, as well as when combined with other components and/or software for the customer end product or application. In particular, take note that:

- ST products may have been certified by one or more security certification bodies, such as Platform Security Architecture (www.psacertified.org) and/or Security Evaluation standard for IoT Platforms (www.trustcb.com). For details concerning whether the ST product(s) referenced herein have received security certification along with the level and current status of such certification, either visit the relevant certification standards website or go to the relevant product page on www.st.com for the most up to date information. As the status and/or level of security certification for an ST product can change from time to time, customers should re-check security certification status/level as needed. If an ST product is not shown to be certified under a particular security standard, customers should not assume it is certified.
- Certification bodies have the right to evaluate, grant and revoke security certification in relation to ST products. These certification bodies are therefore independently responsible for granting or revoking security certification for an ST product, and ST does not take any responsibility for mistakes, evaluations, assessments, testing, or other activity carried out by the certification body with respect to any ST product.
- Industry-based cryptographic algorithms (such as AES, DES, or MD5) and other open standard technologies which may be used in conjunction with an ST product are based on standards which were not developed by ST. ST does not take responsibility for any flaws in such cryptographic algorithms or open technologies or for any methods which have been or may be developed to bypass, decrypt or crack such algorithms or technologies.
- While robust security testing may be done, no level of certification can absolutely guarantee protections against all attacks, including, for example, against advanced attacks which have not been tested for, against new or unidentified forms of attack, or against any form of attack when using an ST product outside of its specification or intended use, or in conjunction with other components or software which are used by customer to create their end product or application. ST is not responsible for resistance against such attacks. As such, regardless of the incorporated security features and/or any information or support that may be provided by ST, each customer is solely responsible for determining if the level of attacks tested for meets their needs, both in relation to the ST product alone and when incorporated into a customer end product or application.
- All security features of ST products (inclusive of any hardware, software, documentation, and the like), including but not limited to any enhanced security features added by ST, are provided on an "AS IS" BASIS. AS SUCH, TO THE EXTENT PERMITTED BY APPLICABLE LAW, ST DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, unless the applicable written and signed contract terms specifically provide otherwise.

Revision history

Table 4. **Document revision history**

Date	Version	Changes
05-Jan-2026	1	Initial release
19-Jan-2026	2	Minor changes.
20-Feb-2026	3	First public release. Modified: • Higher power consumption for Stop 2 and Stop 3 modes • LSE in bypass mode may not be functional
19-May-2026	4	Added: Bootloader inaccessible due to incorrect factory programming of option byte

Contents

1 **Summary of device errata** 2
2 **Description of device errata** 3
2.1 Core 3
2.1.1 Access permission faults are prioritized over unaligned device memory faults 3
2.2 System 3
2.2.1 Incorrect definition of HSERDY flag in reference manual 3
2.2.2 Incorrect definition of the RTCPRE divider formula in the reference manual 4
2.2.3 Incorrect description of the OTP16 BootROM_CONFIG_7 in reference manual 4
2.2.4 Incorrect HSE minimum values in OTP16 BootROM_CONFIG_7 and in RCC section of the reference manual 4
2.2.5 Missing default value for rng_htcr_value in OPT19 BootROM_CONFIG_10 4
2.2.6 Incorrect operational status of PVM monitoring for V_{DDIO2}, V_{DDIO3}, V_{DDIO4}, V_{DDIO5}, USB33, and V_{DDCORE} in reference manual or datasheet 5
2.2.7 Higher power consumption for Stop 2 and Stop 3 modes 5
2.2.8 LSE in bypass mode may not be functional. 5
2.2.9 Bootloader inaccessible due to incorrect factory programming of option byte 5
2.3 TIM 6
2.3.1 Unexpected PWM output when using ocref_clr 6
2.4 LPTIM 6
2.4.1 Device may remain stuck in LPTIM interrupt when entering Stop mode 6
2.4.2 ARRM and CPMxOK flags are not set when APB clock is slower than kernel clock 6
2.4.3 Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register 7
2.5 RTC. 7
2.5.1 Alarm flag may be repeatedly set when the core is stopped in debug 7
2.5.2 RTC calendar read may return transient incorrect value when BYPSHAD = 0 7
2.6 I2C 8
2.6.1 Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period. 8
2.6.2 Spurious bus error detection in controller mode 8
2.7 USART 8
2.7.1 Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1 8
2.7.2 Received data may be corrupted upon clearing the ABREN bit. 9
2.7.3 Noise error flag set while ONEBIT is set 9
2.8 LPUART 9
2.8.1 Possible LPUART transmitter issue when using low BRR[15:0] value. 9
2.9 SPI 10

2.9.1 RDY output failure at high serial clock frequency 10
2.10 USB. 10
2.10.1 Buffer description table update completes after CTR interrupt triggers 10
Important security notice 11
Revision history 12

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.