STM32WBA6xxx device errata

Applicability

This document applies to the part numbers of STM32WBA6xxx 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 RM0515.

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
STM32WBA6xxx	STM32WBA62PG, STM32WBA62CG, STM32WBA62MG, STM32WBA63CG, STM32WBA64CG, STM32WBA65RG, STM32WBA65MG, STM32WBA65CG, STM32WBA65PG, STM32WBA62PI, STM32WBA62CI, STM32WBA62MI, STM32WBA63CI, STM32WBA64CI, STM32WBA65RI, STM32WBA65MI, STM32WBA65CI, STM32WBA65PI

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(2)	DIE_ID(1)
STM32WBA6xxx	Z	0x1001

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

1 Summary of device errata

The following table gives a quick reference to the STM32WBA6xxx device limitations and their status:

A = limitation present, workaround available

N = limitation present, no workaround available

P = limitation present, partial workaround available

"-" = limitation absent

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. Z
Core	2.1.1	Access permission faults are prioritized over unaligned device memory faults	N
System	2.2.1	LSE crystal oscillator may be disturbed by transitions on PC13	N
	2.2.2	Device-specific authentication ID is not accessible in RDP Level 0	A
	2.2.3	HSEPRE cannot be changed while HSE is set as system clock or PLL source	A
	2.2.4	RCC audio synchronization registers cannot be updated while the counter is enabled	A
	2.2.5	Bit LPWRRSTF of RCC_CSR can always be read	N
	2.2.6	Glitches on PA2 and PA7 in retention Standby mode	A
	2.2.7	ICACHE clock requires register RCC_AHB1ENR to have a non-zero value	A
	2.2.8	RTC clocked by LSI stops working when a reset is triggered from the NRST pad	A
	2.2.9	Fast-mode Plus cannot be activated using SYSCFG_CFGR1 register	N
	2.2.10	Longer HSE32 stabilization time when the clock is stopped for a time between 2 and 5 ms	A
	2.2.11	No security gating is applied to MCO on PA8 when AF0 is selected	N
	2.2.12	Reset can cause the system to get stuck in Standby mode or STOP 2 LPMODE	N
	2.2.13	RCC_BDCR1 register reset behavior not correctly implemented	N
	2.2.14	Privilege and secure access to RCC_CCIPR2 register ASSEL and OTGHSSEL bits not correctly implemented	N
	2.2.15	VREFBUF control by VRS[2:0] not operational	A
	2.2.16	PD9 leakage issue in USB mode	N
	2.2.17	System cannot enter LPMODE if RCC CFGR2.HPRE is not equal to 0	A
Radio system	2.3.1	Bluetooth® LE frequency deviation	P
	2.3.2	Nonlinear behavior of Bluetooth® LE RSSI reporting	N
LPTIM	2.5.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A
	2.5.2	ARRM and CMPM flags are not set when APB clock is slower than kernel clock	A
	2.5.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N
RTC	2.6.1	Alarm flag may be repeatedly set when the core is stopped in debug	N
I2C	2.7.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P

Function	Section	Limitation	Status Rev. Z
I2C	2.7.2	Spurious bus error detection in controller mode	A
USART	2.8.1	Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1	A
	2.8.2	Received data may be corrupted upon clearing the ABREN bit	A
	2.8.3	Noise error flag set while ONEBIT is set	N
LPUART	2.9.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P
SPI	2.10.1	RDY output failure at high serial clock frequency	N

The following table gives a quick reference to the documentation errata.

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
SAES	2.4.1	Data transfer from TAMP_BKPxR to key registers must be done only in ascending order when KEYSEL[2:0] is set to 010 or 100

2 Description of device errata

The following sections describe the errata of the applicable devices with Arm® core and provide workarounds if available. They are grouped by device functions.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2.1 Core

Reference manual and errata notice for the Arm® Cortex®-M33 core revision r0p 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 LSE crystal oscillator may be disturbed by transitions on PC13

Description

On UFQFPN packages, the LSE crystal oscillator clock frequency can be incorrect when PC13 is toggling in input or output (for example when used for RTC_OUT1).

The external clock input (LSE bypass) is not impacted by this limitation.

The WLCSP and UFBGA packages are not impacted by this limitation.

Workaround

None.

Avoid toggling PC13 when LSE is used on UFQFPN packages.

2.2.2 Device-specific authentication ID is not accessible in RDP Level 0

Description

The AUTH_ID bitfield of the DBGMCU_DBG_AUTH_DEVICE register is not accessible in RDP Level 0. The read value is always 0. Therefore, this bitfield cannot be used to discriminate between different devices.

Workaround

Increase the RDP to Level 1 before reading the device-specific authentication ID. Then, decrease the RDP back to Level 0.

2.2.3 HSEPRE cannot be changed while HSE is set as system clock or PLL source

Description

The clock divider may produce glitches if changed while HSE is running.

Workaround

Set the system clock temporarily to HSI, then change the HSEPRE setting of the divider.

2.2.4 RCC audio synchronization registers cannot be updated while the counter is enabled

Description

The compare register ASCOR cannot be used when the synchronization has started.

Workaround

For writing, CEN should be set to zero leading to a reset of registers, including the free running counter.

2.2.5 Bit LPWRRSTF of RCC_CSR can always be read

Description

Bit LPWRRSTF of RCC_CSR can be always read regardless of the privilege level set in the RCC_PRIVCFGR register.

Workaround

None

2.2.6 Glitches on PA2 and PA7 in retention Standby mode

Description

PA2 and PA7 correspond with ADC4_IN7 and ADC4_IN2. When coming into or coming out of Standby mode with retention, these signals may show glitches.

Workaround

Discard the IO retention feature, and force the pull-down on these two signals.

2.2.7 ICACHE clock requires register RCC_AHB1ENR to have a non-zero value

Description

ICACHE can be configured to perform an address remap to SRAM.
The ICACHE clock is disabled when all the bits of the RCC_AHB1ENR register are at zero.
A deadlock can occur when:

- code and data are stored in SRAM2,
- ICACHE is configured to remap the addresses 0x0A00/0x0E00 to 0x2000/0x3000,
- all the register RCC_AHB1ENR bits are at zero, and
- the CPU attempts to reboot from 0x0A00/0x0E00.

The device does not boot.

Workaround

Ensure that at least one bit of the RCC_AHB1ENR register is set.

2.2.8 RTC clocked by LSI stops working when a reset is triggered from the NRST pad

Description

An external trigger from the NRST pad resets the LSI1ON, LSI1PREDIV, and RADIOSTSEL bits of the RCC_BDCR1 register, causing RTC and TAMP to stop functioning.

Workaround

Use as the clock for RTC.

2.2.9 Fast-mode Plus cannot be activated using SYSCFG_CFGR1 register

Description

The activation of the Fast-mode Plus mode on GPIO PB3, PA15, PA7, or PA6 by setting the corresponding bit in the SYSCFG_CFGR1 register is ineffective.

Workaround

None.

2.2.10 Longer HSE32 stabilization time when the clock is stopped for a time between 2 and 5 ms

Description

When the HSE32 is restarted after having been off within a window of time between 2 and 5 ms, it may be not ready even if the HSERDY bit is set in the RCC_CR register. This may result in a CPU hard fault, wrong timer counting, or an incorrect behavior of the other peripherals that use the HSE32 clock.

In this case, the HSE32 oscillator stabilization time t_{STAB} may be increased by 1 ms (360 µs typical).

Workaround

Apply the following measure in applications where the HSE32 clock is stopped for more than 2 ms and less than 5 ms:

1. Before stopping the HSE32 clock, deselect HSE32 for all the peripherals that use this clock as kernel clock:
   - SYSCLK via the SW[1:0] bitfield of the RCC_CFGR1 register
   - PLL1 via the PLL1SRC[1:0] bitfield of the RCC_PLL1CFGR register
   - RTC and TAMP via the RTCSEL[1:0] bitfield in the RCC_BDCR1 register
   - ADC4 via the ADCSEL[2:0] bitfield of the RCC_CCIPR3 register
   - 2.4 GHz RADIO sleep clock via the RADIOSTSEL[1:0] bitfield of the RCC_BDCR1 register.
   - 2.4 GHz RADIO baseband kernel clock shall be disabled via the BBCLKEN bitfield of the RCC_RADIOENR register.
2. Wait until HSERDY is set, then wait for an additional time of 200 µs before using HSE32 again for the SYSCLK, PLL1, RTC, TAMP, ADC4, 2.4 GHz RADIO sleep clock, or 2.4 GHz RADIO baseband clock.
3. Keep the HSE clock security disabled, by clearing the HSECSSON bit in the RCC_CR register.

The above measure does not prevent the 2.4 GHz RADIO to occasionally miss some packets when t_{STAB} is higher than 360 µs and smaller than 1 ms). To prevent this issue for occurring, the 2.4 GHz RADIO wake-up can be anticipated by 600 µs.

2.2.11 No security gating is applied to MCO on PA8 when AF0 is selected

Description

When GPIO alternate function 0 is selected (AF0), the MCO is output on PA8. Setting the SYSCLKSEC bit of the RCC_SECCFGR register should make this output secure. Instead, the MCO is still output on PA8.

Workaround

None.

2.2.12 Reset can cause the system to get stuck in Standby mode or STOP 2 LPMODE

Description

If NRESET is activated after a few nanoseconds of an internal standby or stop 2 request, the system may be unable to exit Standby or Stop 2 mode, in which case a POR is necessary.

Workaround

None.

2.2.13 RCC_BDCR1 register reset behavior not correctly implemented

Description

The RCC_BDCR1 register resets with Backup domain power-on reset except for the BDRST bit, which resets on POR (power-on reset).

In the current implementation, some bits of the RCC_DBCR1 register are reset with padreset or poreset.

Workaround

None.

2.2.14 Privilege and secure access to RCC_CCIPR2 register ASSEL and OTGHSSEL bits not correctly implemented

Description

The configuration bits of the clock of one IPx can be accessed if the access is performed with the same or higher security of the IPx. The access can be secured by GTZC_TZSC registers IPxSEC bits. When in securemode, a nonsecure read/write access is RAZ/WI (Read As Zero/Write Ignored). This bit can be protected against unprivileged access when in secure mode using the RCC_SPRIV register, or when in nonsecure mode using the RCC_NSPRIV register.

However, the secure/privilege information for the SAI and OTGHS has not been correctly connected in the ASSEL and OTGHSSEL bitfields of the RCC_CCIPR2 register:

- ASSEL access depends on the OTGHS security/privilege attribute.
- OTGHSSEL[1] access depends on the SAI security/privilege attribute.
- OTGHSSEL[0] does not depend on any security/privilege attribute.

Workaround

None.

2.2.15 VREFBUF control by VRS[2:0] not operational

Description

Upon reset and upon writing the VRS[2:0] bitfield of the VREFBUF_CSR register, the TRIM[5:0] bitfield of the VREFBUF_CCR register is expected to automatically be filled with the corresponding factory VREFBUF calibration value. However, the TRIM[5:0] takes the value 0x3F upon reset and writing the VRS[2:0] bitfield has no effect on it.

Workaround

Upon reset and upon changing the VRS[2:0] value, copy (by software) the appropriate VREFBUFx_TRIM[5:0] calibration into the bitfield TRIM[5:0] of the VREFBUF_CCR register.

2.2.16 PD9 leakage issue in USB mode

Description

When the USB internal switches are activated, PD9 becomes a dedicated pad for USB (VBUS pad). In this USB scenario, a leakage current consumption of approximately 35 µA occurs if an external pull-up or voltage (equal to V_{DDUSB}) is applied to PD9.

VBUS is used in device mode, not in host mode. This unexpected extraconsumption is not an issue for external devices. PD9 functions correctly in GPIO mode.

Workaround

None.

2.2.17 System cannot enter LPMODE if RCC CFGR2.HPRE is not equal to 0

Description

The AHB4 clock can be configured to be equal to or a divided version of SYSCLK using the HPRE bitield of the RCC_CFGR2 register.

RCC and PWRCTRL FSMs function with SYSCLK, but PWRCTRL FSM is gated by the synchronization phase of SYSCLK and CK AHB4. The system cannot enter LPMODE if SYSCLK and CK AHB4 have different frequency (which is set by the HPRE bits of the RCC_CFGR2 register). As a consequence:

- The system stays in Run mode.
- Clocks are stopped like in LPMODE.

Workaround

Clear the HPRE bitfield of the RCC_CFGR2 register to remove the CK AHB4 division factor before requesting entry into the LPMODE.

2.3 Radio system

2.3.1 Bluetooth® LE frequency deviation

Description

The channel 15 and 31 frequencies are slightly out of specification, which can create communication failures.

The fact that the Bluetooth® LE stack handles retries upon failure and the Bluetooth® LE protocol uses channel hopping reduces the impact of such failures.

Note: The product still meets the Bluetooth® LE certification requirements.

Workaround

When the device plays a critical role in a Bluetooth® LE application, avoid using these channels by issuing the HCI update channel map command HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION.

2.3.2 Nonlinear behavior of Bluetooth® LE RSSI reporting

Description

The RSSI is linear only in the range [-32 dBm; -70 dBm].

Workaround

None.

2.4 SAES

2.4.1 Data transfer from TAMP_BKPxR to key registers must be done only in ascending order when KEYSEL[2:0] is set to 010 or 100

Description

The KEYSEL[2:0] bitfield of the SAES_CR register defines the source of the key information to use in the SAES cryptographic core:

- When KEYSEL[2:0] is set to 010, the boot hardware key (BHK), stored in tamper-resistant secure backup registers, is entirely transferred into the key registers upon a secure application performing a single read of all TAMP_BKPxR registers (x = 0 to 3 for KEYSIZE = 0, x = 0 to 7 for KEYSIZE = 1).
- When KEYSEL[2:0] is set to 100, the XOR combination of DHUK and BHK is entirely transferred into the key registers upon a secure application performing a single read of all TAMP_BKPxR registers (x = 0 to 3 for KEYSIZE = 0, x = 0 to 7 for KEYSIZE = 1).

Some revisions of the reference manual may wrongly specify that the read operation can be performed either in ascending or descending order, while it must be performed always in **ascending** order.

This is a documentation issue rather than a product limitation.

Workaround

No application workaround is required, provided that the read operation to the TAMP_BKPxR registers is always done in ascending order.

2.5 LPTIM

2.5.1 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.

2.5.2 ARRM and CMPM 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 CMPM 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, CMP} ≥ 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 CMP 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:

{CMP, ARR} ≥ 16 MHz / (2 * 1 MHz)

→ ARR must be ≥ 8 and CMP 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.

2.5.3 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.

2.6 RTC

2.6.1 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.

2.7 I2C

2.7.1 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.

2.7.2 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.

2.8 USART

2.8.1 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.

2.8.2 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.

2.8.3 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.

2.9 LPUART

2.9.1 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.

2.10 SPI

2.10.1 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.

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 5. **Document revision history**

Date	Version	Changes
14-Feb-2025	1	Initial release.
04-Jul-2025	2	Modified errata: • VREFBUF control by VRS[2:0] not operational • Bluetooth® LE frequency deviation

Contents