STM32WBA5x device errata

Applicability

This document applies to the part numbers of STM32WBA5x 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 RM0493.

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
STM32WBA5x	STM32WBA52CE, STM32WBA52CG, STM32WBA52KE, STM32WBA52KG, STM32WBA54KG, STM32WBA54KE, STM32WBA54CG, STM32WBA54CE, STM32WBA55CG, STM32WBA55CE, STM32WBA55UG, STM32WBA55UE

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32WBA5x	A	0x1000
STM32WBA5x	B	0x2000

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 STM32WBA5x device limitations and their status:

A = workaround available

N = no workaround available

P = partial workaround available

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	Rev. B
Core	2.1.1	Access permission faults are prioritized over unaligned Device memory faults	N	N
	2.1.2	Incorrect SYST_CALIB register value	A	-
System	2.2.1	WWDG bus clock not enabled when hardware window watchdog is selected by WWDG_SW option bit	A	-
	2.2.2	Extra delay required before changing LSI1 division ratio	A	-
	2.2.3	SYSCLK may not reach the PLL1 frequency when step switch is enabled	A	-
	2.2.4	Spurious deactivation of HSE when HSI is selected as system clock source	N	-
	2.2.5	LSI1 oscillator wrongly enabled when HSE32 is selected as RTC kernel clock	N	-
	2.2.6	LSI1 oscillator does not automatically start when selected as RTC kernel clock	A	-
	2.2.7	When IWDG is active LSI1 oscillator is disabled by a backup domain reset	A	-
	2.2.8	A DMA transfer from the flash memory to SRAM may fail	P	-
	2.2.9	Incorrect hardware sequence of waking up from Stop mode	A	-
	2.2.10	LSE crystal oscillator may be disturbed by transitions on PC13	N	N
	2.2.11	Device-specific authentication ID is not accessible in RDP Level 0	A	A
	2.2.12	HSI16 clock cannot be stopped when used as kernel clock by ADC4	A	-
	2.2.13	LSECSSD is not cleared by clearing LSECSSON	P	-
	2.2.14	System stuck in Stop mode when HSI is selected as system clock source	A	-
	2.2.15	Accessing ICACHE after exiting from Stop 1 mode may lead to a HardFault	A	-
	2.2.16	SFI not implemented	N	-
	2.2.17	HSEPRE cannot be changed while HSE is set as system clock or PLL source	A	A
	2.2.18	RCC audio synchronization registers cannot be updated while the counter is enabled	-	A
	2.2.19	Bit LPWRRSTF of RCC_CSR can always be read	N	N
	2.2.20	Glitches on PA2 and PA7 in retention Standby mode	A	A
	2.2.21	ICACHE clock requires register RCC_AHB1ENR to have a non-zero value	A	A
Radio system	2.3.1	Bluetooth® Low Energy frequency deviation (DF1)	P	P

Function	Section	Limitation	Status	
			Rev. A	Rev. B
Radio system	2.3.2	Bluetooth® Low Energy radio system is not FCC EMC-compliant for output powers higher than +7.5 dBm	P	-
	2.3.3	Nonlinear behavior of Bluetooth® Low Energy RSSI reporting	N	N
	2.3.4	Bluetooth® Low Energy radio system is not MIC-compliant for output powers higher than +7.5 dBm	N	N
	2.3.5	HSE overconsumption for radio	-	N
GPIO	2.4.1	PA0 does not work when configured as EVENTOUT	N	-
ADC	2.5.1	ADC clock request is active after reset	A	-
TIM	2.8.1	Consecutive compare event missed in specific conditions	N	-
	2.8.2	Output compare clear not working with external counter reset	P	-
LPTIM	2.9.1	Device may remain stuck in LPTIM interrupt when entering Stop mode	A	A
	2.9.2	ARRM and CMPM flags are not set when APB clock is slower than kernel clock	P	P
	2.9.3	Interrupt status flag is cleared by hardware upon writing its corresponding bit in LPTIM_DIER register	N	N
	2.9.4	Overcapture stops working when CCxOF flag is cleared in some specific conditions	P	-
RTC and TAMP	2.10.1	Alarm flag may be repeatedly set when the core is stopped in debug	N	N
I2C	2.11.1	Wrong data sampling when data setup time ($t_{SU;DAT}$) is shorter than one I2C kernel clock period	P	P
	2.11.2	Spurious bus error detection in master mode	A	A
	2.11.3	SDA held low upon SMBus timeout expiry in slave mode	A	-
USART	2.12.1	Wrong data received in smartcard mode and 0.5 stop bit configuration	A	-
	2.12.2	Wrong data received by SPI slave receiver in autonomous mode with CPOL = 1	A	A
	2.12.3	Received data may be corrupted upon clearing the ABREN bit	A	A
	2.12.4	Noise error flag set while ONEBIT is set	N	N
LPUART	2.13.1	Possible LPUART transmitter issue when using low BRR[15:0] value	P	P
SPI	2.14.1	Truncation of SPI output signals after EOT event	A	-

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

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
TSC	2.6.1	Inhibited acquisition in short transfer phase configuration
SAES	2.7.1	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 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 will result 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 will be 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 will be 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.

2.1.2 Incorrect SYST_CALIB register value

Description

The Arm® Cortex®-M33 SysTick calibration value defined in the SYST_CALIB register is incorrect. For example, a 1 MHz reference clock does not give a 10 ms clock period.

Workaround

Set the SysTick counter by software according to the system clock frequency.

2.2 System

2.2.1 WWDG bus clock not enabled when hardware window watchdog is selected by WWDG_SW option bit

Description

Upon power-on with the WWDG_SW option bit at 0, the device hardware selects the window watchdog WWDG and it is expected to unconditionally enable the WWDG bus clock, through setting the WWDGEN bit of the RCC_APB1ENR1 register.

However, until the software performs a write access to an RCC register, the WWDGEN bit remains low and the WWDG bus clock disabled.

Workaround

Write one of the RCC registers. For example, write 0x0000 0000 to RCC_CIER.

Note: The instant of writing defines the start of the WWDG countdown.

2.2.2 Extra delay required before changing LSI1 division ratio

Description

The LSI1 clock division (by 1 or by 128) can normally be changed through the LSI1PREDIV bit of the RCC_BDCR1 register when the LSI1 oscillator is stopped, that is, whenever the LSI1ON and LSI1RDY bits are both low.

However, if the LSI1PREDIV bit value is changed immediately after the LSI1RDY bit goes low, the new value of the LSI1PREDIV bit is ignored and the LSI1 frequency remains unchanged upon turning the LSI1 oscillator back on.

Workaround

After the LSI1RDY bit goes low, leave at least a half LSI1 clock cycle (at 32 KHz) of extra delay before changing the LSI1PREDIV bit value.

2.2.3 SYSCLK may not reach the PLL1 frequency when step switch is enabled

Description

The step switch of the SYSCLK to PLL1 is controlled through the PLL1RCLKPRE and PLL1RCLKPRESTEP bits. When enabled, then instead of instantly switching to the PLL1 frequency, the SYSCLK frequency increases progressively, in two or three intermediate frequency steps.

It occurs that at the end of the switch process, SYSCLK does not reach the PLL1 frequency but remains at one of the intermediate frequency steps (for example 44.45 MHz if PLLR is set to 100 MHz and a two-step division is applied).

Workaround

Set the PLL1RCLKPRE bit of RCC_PLL1CFGR twice in the following cases:

- at each startup (after system reset)
- before switching the system clock source from PLL1 to any other source
- when PLL1R uses the system clock before entering Stop mode

2.2.4 Spurious deactivation of HSE when HSI is selected as system clock source

Description

With HSE active, switching the system clock source from any other source to HSI, spuriously deactivates HSE (HSEON = 0 in the RCC_CR register) unless it has been requested by STRADIOCLKON of RCC_RADIOENR.

Workaround

None.

2.2.5 LSI1 oscillator wrongly enabled when HSE32 is selected as RTC kernel clock

Description

When HSE32 is selected as RTC kernel clock source, LSI1 is forced active and cannot be disabled.

Workaround

None.

2.2.6 LSI1 oscillator does not automatically start when selected as RTC kernel clock

Description

Selecting LSI1 as RTC kernel clock source (RTCSEL[1:0] = 10 in RCC_BDCR1 register) fails to automatically start the LSI1 oscillator.

Workaround

Enable the LSI1 oscillator (set the LSI1ON bit of the RCC_BDCR1 register) before selecting LSI1 as RTC kernel clock.

2.2.7 When IWDG is active LSI1 oscillator is disabled by a backup domain reset

Description

As long as IWDG is active, the device hardware is expected to keep the LSI1 oscillator active and the LSI1RDY flag high until the next system reset.

However, the backup domain reset (by setting the BDRST bit of the RCC_BDCR1 register) unduly stops the LSI1 oscillator and clears the LSI1RDY flag of RCC_BDCR1.

Workaround

Enable again the LSI1 oscillator after a backup domain reset.

2.2.8 A DMA transfer from the flash memory to SRAM may fail

Description

A DMA transfer from flash memory to SRAM may fail. The failure rate depends on the source and destination ports, on the setting of the flash memory wait states, and on the system clock frequency.

Workaround

To avoid incorrect results, invalidate the internal cache before starting a data transfer from flash memory to SRAM. To do this, apply the following sequence:

1. Set up a DMA channel to an unused flash memory location, and start reading from this channel:

```
HAL_DMA_Start(&hdma_dummy, (uint32_t)SRCBuffer_in_FLASH, (uint32_t)destBuffer_dummy, (BUFF
ER_SIZE * 2));
```

2. Then start transferring the data to the desired location. Ensure that the dummy transfer starts earlier and ends later than the desired transfer, so that the dummy transfer is active for the entire duration of the desired data transfer:

```
HAL_DMA_Start(&hdma_real, (uint32_t)SRCBuffer_in_FLASH, (uint32_t)destBuffer, BUFFER_SIZE)
;
```

2.2.9 Incorrect hardware sequence of waking up from Stop mode

Description

The hardware wake-up sequence may take some time to correctly restart the device: the STOPF flag of the PWR_SR register and the clock settings are correct 25 SYSCLK clock cycles after the wake-up from Stop mode.

Workaround

After WFI, wait for at least 25 SYSCLK clock cycles before reading or modifying any register.

2.2.10 LSE crystal oscillator may be disturbed by transitions on PC13

Description

On LQFP and 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 LQFP and UFQFPN packages.

2.2.11 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.

2.2.12 HSI16 clock cannot be stopped when used as kernel clock by ADC4

Description

Once selected as kernel clock source for the ADC4 peripheral, the HSI16 clock can no longer be disabled using the HSION bit of the RCC_CR register. This is due to the fact that ADC4 requests its kernel clock by default after reset. When the device enters Stop 0 or Stop 1 mode, the kernel clock HSI16 remains enabled, which leads to an overconsumption in this mode.

Workaround

Before disabling the HSI16 clock, configure ADCSEL[2:0] to HCLK:

- If ADC4 is not used during Stop mode:
  1. Before entering Stop 0 or Stop 1 mode, configure the ADCSEL[2:0] bitfield to HCLK.
  2. Upon Stop mode exit, restore the kernel clock (HSI16) using the ADCSEL[2:0] bitfield.
- If ADC4 is used during Stop mode:
  Before entering Stop 0 or Stop 1 mode, simply configure the ADCSEL[2:0] bitfield to use HSI16 as kernel clock. ADC4 requests its kernel clock when it needs it.

2.2.13 LSECSSD is not cleared by clearing LSECSSON

Description

LSECSSD flag set in of the RCC_BDCR1 register indicates that a clock security issue occurred. The software must then clear the LSECSSON bit, which in turns clears LSECSSD. However, until a backup domain reset occurs, the LSECSSD flag remains set.

Workaround

Set the BDRST bit of RCC_BDCR1 to reset LSECSSD.

2.2.14 System stuck in Stop mode when HSI is selected as system clock source

Description

When HSI is selected as system clock source and the device is in Stop mode, the device may not react correctly to a wake-up signal, and the system may remain in Stop mode.

Workaround

Use HSE as system clock source when entering Stop mode.

2.2.15 Accessing ICACHE after exiting from Stop 1 mode may lead to a HardFault

Description

ICACHE RAM is switched off (ICRAMPDS set in PWR_CR2 register) when the device enters Stop 1 mode. Attempting to access ICACHE just after exiting from Stop 1 exit may lead to a HardFault error.

Workaround

Disable ICACHE memory before entering Stop 1 mode and enable it again after exiting Stop 1 mode.

2.2.16 SFI not implemented

Description

The secure firmware install (SFI) is not available.

Workaround

None.

2.2.17 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.

2.2.18 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.

2.2.19 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

2.2.20 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.

2.2.21 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.

2.3 Radio system

2.3.1 Bluetooth® Low Energy frequency deviation (DF1)

Description

Channel 15 and 31 frequencies are out of specifications.

Workaround

Discard channels 15 and 31 by issuing an HCI update channel map command HCI_LE_SET_HOST_CHANNEL_CLASSIFICATION.

2.3.2 Bluetooth® Low Energy radio system is not FCC EMC-compliant for output powers higher than +7.5 dBm

Description

The Bluetooth® Low Energy radio system is expected to be EMC-compliant for output powers up to +10 dBm. However, when the output power is higher than +7.5 dBm, the band-edge test on channel 39 fails the FCC certification Part 15.247.

Workaround

Use the Cube firmware version V1.1.0 or later to enable the band-edge test on channel 39, to pass the FCC certification.

2.3.3 Nonlinear behavior of Bluetooth® Low Energy RSSI reporting

Description

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

Workaround

None.

2.3.4 Bluetooth® Low Energy radio system is not MIC-compliant for output powers higher than +7.5 dBm

Description

The Bluetooth® Low Energy radio system is expected to be MIC-compliant for output powers up to +10 dBm. However, when the output power is higher than +7.5 dBm, the spurious emission test on channel 0 passes the MIC certification for 1 Mbps, but fails for 2 Mbps with 10 dBm. Both rates pass with 7.5dBm.

Workaround

None

2.3.5 HSE overconsumption for radio

Description

The power consumption when using HSE clock is 200 uA more than indicated in the datasheet.

Workaround

None.

2.4 GPIO

2.4.1 PA0 does not work when configured as EVENTOUT

Description

The PA0 GPIO configured as alternate function AF15 (EVENTOUT) does not signal any events.

Workaround

None.

2.5 ADC

2.5.1 ADC clock request is active after reset

Description

After reset (system startup), the ADC requests the kernel clock even if it is not configured nor activated. This creates additional current consumption.

Workaround

Apply the following sequence at system startup:

1. Set ADEN bit of ADC_CR to enable the ADC, and wait a minimum time of two ADC_CLK cycles.
2. Set the ADDIS bit of ADC_CR, then wait until ADEN is cleared (it takes six ADC_CLK cycles).
3. Clear ADVREGEN bit of ADC_CR to disable the ADC voltage regulator, previously set by setting ADEN bit.

2.6 TSC

2.6.1 Inhibited acquisition in short transfer phase configuration

Description

Some revisions of the reference manual may omit the information that the following configurations of the TSC_CR register are forbidden:

- The PGPSC[2:0] bitfield set to 000 and the CTPL[3:0] bitfield to 0000 or 0001
- The PGPSC[2:0] bitfield set to 001 and the CTPL[3:0] bitfield to 0000

Failure to respect this restriction leads to an inhibition of the acquisition.

This is a documentation inaccuracy issue rather than a product limitation.

Workaround

No application workaround is required.

2.7 SAES

2.7.1 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.

2.8 TIM

2.8.1 Consecutive compare event missed in specific conditions

Description

Every match of the counter (CNT) value with the compare register (CCR) value is expected to trigger a compare event. However, if such matches occur in two consecutive counter clock cycles (as consequence of the CCR value change between the two cycles), the second compare event is missed for the following CCR value changes:

- in edge-aligned mode, from ARR to 0:
  - first compare event: CNT = CCR = ARR
  - second (missed) compare event: CNT = CCR = 0

- in center-aligned mode while up-counting, from ARR-1 to ARR (possibly a new ARR value if the period is also changed) at the crest (that is, when TIMx_RCR = 0):
  - first compare event: CNT = CCR = (ARR-1)
  - second (missed) compare event: CNT = CCR = ARR
- in center-aligned mode while down-counting, from 1 to 0 at the valley (that is, when TIMx_RCR = 0):
  - first compare event: CNT = CCR = 1
  - second (missed) compare event: CNT = CCR = 0

This typically corresponds to an abrupt change of compare value aiming at creating a timer clock single-cycle-wide pulse in toggle mode.

As a consequence:

- In toggle mode, the output only toggles once per counter period (squared waveform), whereas it is expected to toggle twice within two consecutive counter cycles (and so exhibit a short pulse per counter period).
- In center mode, the compare interrupt flag does note rise and the interrupt is not generated.

Note: The timer output operates as expected in modes other than the toggle mode.

Workaround

None.

2.8.2 Output compare clear not working with external counter reset

Description

The output compare clear event (ocref_clr) is not correctly generated when the timer is configured in the following slave modes: Reset mode, Combined reset + trigger mode, and Combined gated + reset mode.

The PWM output remains inactive during one extra PWM cycle if the following sequence occurs:

1. The output is cleared by the ocref_clr event.
2. The timer reset occurs before the programmed compare event.

Workaround

Apply one of the following measures:

- Use BKIN (or BKIN2 if available) input for clearing the output, selecting the Automatic output enable mode (AOE = 1).
- Mask the timer reset during the PWM ON time to prevent it from occurring before the compare event (for example with a spare timer compare channel open-drain output connected with the reset signal, pulling the timer reset line down).

2.9 LPTIM

2.9.1 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.

2.9.2 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
- Repetition counter is set with REP[7:0] = 0x3 (odd value)

The above example is subject to issue, 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 LPTIM_RCR register (=3, odd) to assess the risk of issue.

2.9.3 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.

2.9.4 Overcapture stops working when CCxOF flag is cleared in some specific conditions

Description

The overcapture stops working if the CCxOF flag in the LPTIMx_ISR register is cleared simultaneously with a new input capture event detection, that is, when an input capture pulse is active.

As a result, the LPTIM does not detect anymore overcapture events, no interrupt is generated, and the CCxOF flag is not set.

Workaround

Disable the corresponding input capture channel (the CCxE bit of the LPTIM_CCMRx register cleared) immediately after clearing the CCxOF flag, then enable the channel again after a delay that must be equal or greater than the value of (PRESC * 3) kernel clock cycles, PRESC[2:0] being the clock decimal division factor (1, 2, 4,..128).

2.10 RTC and TAMP

2.10.1 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.

2.11 I2C

2.11.1 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 slave 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.

2.11.2 Spurious bus error detection in master mode

Description

In master 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 master mode and any such transfer continues normally.

Workaround

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

2.11.3 SDA held low upon SMBus timeout expiry in slave mode

Description

For the slave mode, the SMBus specification defines $t_{TIMEOUT}$ (detect clock low timeout) and $t_{LOW:SEXT}$ (cumulative clock low extend time) timeouts. When one of them expires while the I2C peripheral in slave mode drives SDA low to acknowledge either its address or a data transmitted by the master, the device is expected to report such an expiry and release the SDA line.

However, although the device duly reports the timeout expiry, it fails to release SDA. This stalls the I^2C bus and prevents the master from generating RESTART or STOP condition.

Workaround

When a timeout is reported in slave mode (TIMEOUT bit of the I2C_ISR register is set), apply this sequence:

1. Wait until the frame is expected to end.
2. Read the STOPF bit of the I2C_ISR register. If it is low, reset the I2C kernel by clearing the PE bit of the I2C_CR1 register.
3. Wait for at least three APB clock cycles before enabling again the I2C peripheral.

2.12 USART

2.12.1 Wrong data received in smartcard mode and 0.5 stop bit configuration

Description

The USART receiver reads wrong data in smartcard mode and 0.5 stop bit configuration.

Workaround

Use the 1.5 stop bit configuration.

2.12.2 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.

2.12.3 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.

2.12.4 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.

2.13 LPUART

2.13.1 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.

2.14 SPI

2.14.1 Truncation of SPI output signals after EOT event

Description

After an EOT event signaling the end of a non-zero transfer size transaction (TSIZE > 0) upon sampling the last data bit, the software may disable the SPI peripheral. As expected, disabling SPI deactivates the SPI outputs (SCK, MOSI and SS when the SPI operates as a master, MISO when as a slave), by making them float or statically output their by-default levels, according to the AFCNTR bit of the SPI_CFG2 register.

With fast software execution (high PCLK frequency) and slow SPI (low SCK frequency), the SPI disable occurring too fast may result in truncating the SPI output signals. For example, the device operating as a master then generates an asymmetric last SCK pulse (with CPHA = 0), which may prevent the correct last data bit reception by the other node involved in the communication.

Workaround

Apply one of the following measures or their combination:

- Add a delay between the EOT event and SPI disable action.
- Decrease the ratio between PCLK and SCK frequencies.

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
03-Mar-2023	1	Initial release.
20-Jun-2023	2	Added part numbers STM32WBA52KE and STM32WBA52KG. Corrected core type and revision. Modified errata: • Bluetooth® Low Energy radio system is not EMC-compliant for output powers higher than +7.5 dBm , with the workaround re-qualified from N to P • Consecutive compare event missed in specific conditions • Possible LPUART transmitter issue when using low BRR[15:0] value
07-Mars-2024	3	Added part numbers STM32WBA54KG, STM32WBA54KE, STM32WBA54CG, STM32WBA54CE, STM32WBA55CG, STM32WBA55CE, STM32WBA55UG, and STM32WBA55UE Corrected core revision. Added errata: • HSEPRE cannot be changed while HSE is set as system clock or PLL source • RCC audio synchronization registers cannot be updated while the counter is enabled • Bluetooth® Low Energy radio system is not MIC-compliant for output powers higher than +7.5 dBm Modified errata: • Access permission faults are prioritized over unaligned Device memory faults, with the workaround re-qualified from *N* to *P* • LSE crystal oscillator may be disturbed by transitions on PC13 • Bluetooth® Low Energy radio system is not FCC EMC-compliant for output powers higher than +7.5 dBm • Bit LPWRRSTF of RCC_CSR can always be read • Glitches on PA2 and PA7 in retention Standby mode • HSE overconsumption for radio • ICACHE clock requires register RCC_AHB1ENR to have a non-zero value

Contents