STM32G070CB/KB/RB device errata

Applicability

This document applies to the part numbers of STM32G070CB/KB/RB 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 RM0454.

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
STM32G070CB	STM32G070CBT6, STM32G070CBT6TR
STM32G070KB	STM32G070KBT6, STM32G070KBT6TR
STM32G070RB	STM32G070RBT6, STM32G070RBT6TR

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32G070CB/KB/RB	B	0x2000
	Y	0x2002

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 STM32G070CB/KB/RB 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. B	Rev. Y
System	2.2.1	Unstable LSI when it clocks RTC or CSS on LSE	P	P
	2.2.2	WUFx wakeup flag wrongly set during configuration	A	A
	2.2.3	Under Level 1 read protection, booting from main flash memory selected through PA14-BOOT0 pin is not functional	N	-
	2.2.4	DMAMUX cannot be synchronized or triggered by EXTI	N	-
	2.2.5	Overwriting with all zeros a flash memory location previously programmed with all ones fails	N	N
	2.2.6	Wakeup from Stop not effective under certain conditions	N	N
	2.2.7	PC13 signal transitions disturb LSE	N	N
	2.2.9	Corrupted content of the RTC domain due to a missed power-on reset after this domain supply voltage drop	A	A
DMA	2.3.1	DMA disable failure and error flag omission upon simultaneous transfer error and global flag clear	A	A
DMAMUX	2.4.1	SOFx not asserted when writing into DMAMUX_CFR register	N	N
	2.4.2	OFx not asserted for trigger event coinciding with last DMAMUX request	N	N
	2.4.3	OFx not asserted when writing into DMAMUX_RGCFR register	N	N
	2.4.4	Wrong input DMA request routed upon specific DMAMUX_CxCR register write coinciding with synchronization event	A	A
ADC	2.5.1	Overrun flag is not set if EOC reset coincides with new conversion end	P	P
	2.5.2	Writing ADC_CFGR1 register while ADEN bit is set resets RES[1:0] bitfield	A	A
	2.5.3	Out-of-threshold value is not detected in AWD1 Single mode	A	A
	2.5.4	ADC sampling time might be one cycle longer	N	N
	2.5.6	ADC offset may be out of specification	A	-
TIM	2.6.1	One-pulse mode trigger not detected in master-slave reset + trigger configuration	P	P
	2.6.2	Consecutive compare event missed in specific conditions	N	N
	2.6.3	Output compare clear not working with external counter reset	P	P
	2.6.4	TIM16 and TIM17 are unduly clocked by SYSCLK	N	-
RTC and TAMP	2.7.1	Calendar initialization may fail in case of consecutive INIT mode entry	A	A
I2C	2.8.1	Wrong data sampling when data setup time (tSU;DAT) is shorter than one I2C kernel clock period	P	P

Function	Section	Limitation	Status	
			Rev. B	Rev. Y
I2C	2.8.2	Spurious bus error detection in master mode	A	A
	2.8.3	Spurious master transfer upon own slave address match	P	P
USART	2.9.1	Data corruption due to noisy receive line	N	N
SPI	2.10.1	BSY bit may stay high when SPI is disabled	A	A
	2.10.2	BSY bit may stay high at the end of data transfer in slave mode	A	A

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

Table 4. Summary of device documentation errata

Function	Section	Documentation erratum
System	2.2.8	Boot select after debug interface connection
ADC	2.5.5	ADC trigger latency parameter
USART	2.9.2	USART prescaler feature missing in USART implementation section

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®-M0+ core revision r0p1 is available from http://infocenter.arm.com.

2.2 System

2.2.1 Unstable LSI when it clocks RTC or CSS on LSE

Description

The LSI clock can become unstable (duty cycle different from 50 %) and its maximum frequency can become significantly higher than 32 kHz, when:

- LSI clocks the RTC, or it clocks the clock security system (CSS) on LSE (which holds when the LSECSSON bit set), and
- the V_{DD} power domain is reset while the backup domain is not reset, which happens:
  - upon exiting Shutdown mode
  - if V_{BAT} is separate from V_{DD} and V_{DD} goes off then on
  - if V_{BAT} is tied to V_{DD} (internally in the package for products not featuring the VBAT pin, or externally) and a short (< 1 ms) V_{DD} drop under V_{DD}(min) occurs

Workaround

Apply one of the following measures:

- Clock the RTC with LSE or HSE/32, without using the CSS on LSE
- If LSI clocks the RTC or when the LSECSSON bit is set, reset the backup domain upon each V_{DD} power up (when the BORRSTF flag is set). If V_{BAT} is separate from V_{DD}, also restore the RTC configuration, backup registers and anti-tampering configuration.

2.2.2 WUFx wakeup flag wrongly set during configuration

Description

Upon configuring a wakeup pin (WKUPx), the corresponding wakeup flag (WUFx) might spuriously go high depending on the state and configuration of the wakeup pin.

Workaround

After configuring a wakeup pin, clear its corresponding WUFx flag.

2.2.3 Under Level 1 read protection, booting from main flash memory selected through PA14-BOOT0 pin is not functional

Description

With the flash memory read protection set to level 1 and the boot mode selected through the PA14-BOOT0 pin (BOOT0 function of the pin), an attempt to boot from main flash memory can wrongly be interpreted by the read protection mechanism as an unauthorized access, preventing the user code execution. Booting from main flash memory operates correctly if selected through option bits (nBOOT_SEL and nBOOT0 both set).

Workaround

None.

2.2.4 DMAMUX cannot be synchronized or triggered by EXTI

Description

The EXTI-related DMAMUX synchronization and trigger inputs are wrongly routed to the *it_exti_per(y)* output instead of being routed to the *exti[15:0]* output lines.

The *it_exti_per(y)* signals are not usable for synchronizing and triggering DMAMUX.

Workaround

None.

2.2.5 Overwriting with all zeros a flash memory location previously programmed with all ones fails

Description

Any attempt to re-program with all zeros (0x0000 0000 0000 0000) a flash memory location previously programmed with 0xFFFF FFFF FFFF FFFF fails and the PROGERR flag of the FLASH_SR register is set.

Note: Flash memory locations in the erased state (that is, not programmed) are not affected by this failure. They can be programmed with any value.

Workaround

None.

2.2.6 Wakeup from Stop not effective under certain conditions

Description

With the HSI clock divider bitfield HSIDIV[2:0] set to a value different from 000, the device fails to enter Stop mode when SYSCLK is set to HSE clock.

With the HSI clock divider bitfield HSIDIV[2:0] set to a value different from 000, peripherals with clock request capability fail to wake the device up from Stop modes.

Workaround

None.

2.2.7 PC13 signal transitions disturb LSE

Description

The PC13 port toggling disturbs the LSE clock.

Workaround

None.

2.2.8 Boot select after debug interface connection

Description

Some revisions of the reference manual may omit the following information.

After connecting the debug interface and until the device power-down, the boot source upon reset or wakeup from a low-power mode is determined by the PA14-BOOT0 pin level before connecting the debug interface (stored by the device), as opposed to the actual PA14-BOOT0 pin level. The device power-up restores the operation of the PA14-BOOT0 pin as direct boot source selector.

This is a documentation issue rather than a device limitation.

Workaround

No application workaround is required or applicable.

2.2.9 Corrupted content of the RTC domain due to a missed power-on reset after this domain supply voltage drop

Description

The RTC domain reset may be missed upon a power-on following a power-off, if its supply voltage drops during the power-off phase hitting a window, which is few mV wide before it starts to rise again. In this critical window, the flip-flops are no longer able to safely retain the information and the RTC domain reset has not yet been triggered. This window is located in the range between 100 mV and 700 mV, with the exact position depending mainly on the device and on the temperature.

This missed reset results in unpredictable values of the RTC domain registers. This may cause a spurious behavior (such as driving the LSCO output pin on PA2 or influencing RTC functions).

Workaround

Apply one of the following measures:

- In the application, let the V_{DD} and V_{BAT} supply voltages fall to a level below 100 mV for more than 200 ms before a new power-on.
- If the above workaround cannot be applied, and the boot follows a power-on reset, erase the RTC domain by software.
  When the application is using shutdown mode, user needs to discriminate between the power-on reset or an exit from a shutdown mode.
  For this purpose, at least one backup register must have been previously programmed with a BKP_REG_VAL value with 16 bits set and 16 bits cleared.
  Robustness of this workaround can be significantly improved by using a CRC rather than registers. The registers are subject to backup domain reset.
  The workaround consists of calculating the CRC of the backup registers: RCC_BDCR and RTC registers, excluding bits modified by HW.
  The CRC result can be stored in the backup register instead of a fixed value. This value needs to be updated for each modification of values covered by CRC, such as by using CRC peripheral.
  At the very beginning of the boot code, insert the following software sequence:
  1. Check the BORRSTF flag of the RCC_CSR register. If set, the reset is caused by a power on, or is exiting from shutdown mode.
  2. If BORRSTF flag is true, and the shutdown mode is used in the application, check that the backup register value is different from BKP_REG_VAL. When tamper detection is enabled,check that no tamper flag is set. If both conditions are met then the reset is caused by a power-on.
  3. If the reset is caused by a power-on, apply the following sequence:
     a. Enable the PWR clock in the RCC, by setting the PWREN bit.
     b. Enable the RTC domain access in the PWR, by setting the DBP bit.
     c. Reset the RTC domain, by:
        i. Writing 0x0001 0000 in the RCC_BDCR register, which sets the BDRST bit and clears other register bits that might not be reset.
        ii. reading the RCC_BDCR register, to make the reset time long enough
        iii. writing 0x0000 0000 in the RCC_BDCR register, to clear the BDRST bit
     d. Clear the BORRSTF flag by setting the RMVF bit of the RCC_CSR register.

2.3 DMA

2.3.1 DMA disable failure and error flag omission upon simultaneous transfer error and global flag clear

Description

Upon a data transfer error in a DMA channel x, both the specific TEIFx and the global GIFx flags are raised and the channel x is normally automatically disabled. However, if in the same clock cycle the software clears the GIFx flag (by setting the CGIFx bit of the DMA_IFCR register), the automatic channel disable fails and the TEIFx flag is not raised.

This issue does not occur with ST's HAL software that does not use and clear the GIFx flag when the channel is active.

Workaround

Do not clear GIFx flags when the channel is active. Instead, use HTIFx, TCIFx, and TEIFx specific event flags and their corresponding clear bits.

2.4 DMAMUX

2.4.1 SOFx not asserted when writing into DMAMUX_CFR register

Description

The SOFx flag of the DMAMUX_CSR status register is not asserted if overrun from another DMAMUX channel occurs when the software writes into the DMAMUX_CFR register.

This can happen when multiple DMA channels operate in synchronization mode, and when overrun can occur from more than one channel. As the SOFx flag clear requires a write into the DMAMUX_CFR register (to set the corresponding CSOFx bit), overrun occurring from another DMAMUX channel operating during that write operation fails to raise its corresponding SOFx flag.

Workaround

None. Avoid the use of synchronization mode for concurrent DMAMUX channels, if at least two of them potentially generate synchronization overrun.

2.4.2 OFx not asserted for trigger event coinciding with last DMAMUX request

Description

In the DMAMUX request generator, a trigger event detected in a critical instant of the last-generated DMAMUX request being served by the DMA controller does not assert the corresponding trigger overrun flag OFx. The critical instant is the clock cycle at the very end of the trigger overrun condition.

Additionally, upon the following trigger event, one single DMA request is issued by the DMAMUX request generator, regardless of the programmed number of DMA requests to generate.

The failure only occurs if the number of requests to generate is set to more than two (GNBREQ[4:0] > 00001).

Workaround

Make the trigger period longer than the duration required for serving the programmed number of DMA requests, so as to avoid the trigger overrun condition from occurring on the very last DMA data transfer.

2.4.3 OFx not asserted when writing into DMAMUX_RGCFR register

Description

The OFx flag of the DMAMUX_RGSR status register is not asserted if an overrun from another DMAMUX request generator channel occurs when the software writes into the DMAMUX_RGCFR register. This can happen when multiple DMA channels operate with the DMAMUX request generator, and when an overrun can occur from more than one request generator channel. As the OFx flag clear requires a write into the DMAMUX_RGCFR register (to set the corresponding COFx bit), an overrun occurring in another DMAMUX channel operating with another request generator channel during that write operation fails to raise the corresponding OFx flag.

Workaround

None. Avoid the use of request generator mode for concurrent DMAMUX channels, if at least two channels are potentially generating a request generator overrun.

2.4.4 Wrong input DMA request routed upon specific DMAMUX_CxCR register write coinciding with synchronization event

Description

If a write access into the DMAMUX_CxCR register having the SE bit at zero and SPOL[1:0] bitfield at a value other than 00:

- sets the SE bit (enables synchronization),
- modifies the values of the DMAREQ_ID[5:0] and SYNC_ID[4:0] bitfields, and
- does not modify the SPOL[1:0] bitfield,

and if a synchronization event occurs on the previously selected synchronization input exactly two AHB clock cycles before this DMAMUX_CxCR write, then the input DMA request selected by the DMAREQ_ID[5:0] value before that write is routed.

Workaround

Ensure that the SPOL[1:0] bitfield is at 00 whenever the SE bit is 0. When enabling synchronization by setting the SE bit, always set the SPOL[1:0] bitfield to a value other than 00 with the same write operation into the DMAMUX_CxCR register.

2.5 ADC

2.5.1 Overrun flag is not set if EOC reset coincides with new conversion end

Description

If the EOC flag is cleared by an ADC_DR register read operation or by software during the same APB cycle in which the data from a new conversion are written in the ADC_DR register, the overrun event duly occurs (which results in the loss of either current or new data) but the overrun flag (OVR) may stay low.

Workaround

Clear the EOC flag, by performing an ADC_DR read operation or by software within less than one ADC conversion cycle period from the last conversion cycle end, in order to avoid the coincidence with the end of the new conversion cycle.

2.5.2 Writing ADC_CFGR1 register while ADEN bit is set resets RES[1:0] bitfield

Description

Modifying the ADC_CFGR1 register while ADC is enabled (ADEN set in ADC_CR) resets RES[1:0] to 00 whatever the bitfield previous value.

Workaround

Apply the following sequence:

1. Set ADDIS to disable the ADC, and wait until ADEN is cleared.
2. Program the ADC_CFGR1 register according to the application requirements.
3. Set ADEN bit.

2.5.3 Out-of-threshold value is not detected in AWD1 Single mode

Description

AWD1 analog watchdog does not detect that the result of a converted channel has reached the programmed threshold when the ADC operates in Single mode, performs a sequence of conversions, and one of the converted channels other than the first one is monitored by the AWD1 analog watchdog.

Workaround

Apply one of the following measures:

- Use a conversion sequence of one single channel.
- Configure the monitored channel as the first one of the sequence.

2.5.4 ADC sampling time might be one cycle longer

Description

For sampling time set to 1.5 or 3.5 cycles, the sampling in a single ADC conversion or in the first conversion of a sequence takes one extra cycle.

Workaround

None.

2.5.5 ADC trigger latency parameter

Description

Some datasheet revisions state incorrectly ADC trigger latency values for CKMODE set to 01, 10, and 11, as 2.75, 2.63, and 3 ADC clock cycles at maximum, respectively.

The correct specification is 6.5, 12.5, and 3.5 PCLK cycles typical, respectively.

This is a documentation issue rather than a product limitation.

Workaround

No application workaround is applicable.

.

2.5.6 ADC offset may be out of specification

Description

When the V_{REF+} voltage is lower than 3.0 V, the ADC hardware calibration may not fully compensate for the offset error caused by the diffusion process variation. When this occurs, the CALFACT register value is either 0x00 or 0x7F and the EO parameter is out of the maximum stated in the device data sheet. To reflect this, the following specification substitutes the one in the device datasheet:

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EO	Offset error	$2\ V < V_{DDA} = V_{REF+} < 3\ V$ $f_{ADC} = 35\ MHz$, $f_s \leq 2.5\ Msps$	-	1.5	31	LSB

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EO	Offset error	T_A = entire range				LSB
		1.65 V < V_{DDA} = V_{REF+} < 3 V f_{ADC} = 35 MHz, f_s ≤ 2.2 Msps f_{ADC} = 16 MHz; f_s ≤ 1.1 Msps T_A = entire range	-	1.5	50	
ET	Total unadjusted error	2 V < V_{DDA} = V_{REF+} < 3 V f_{ADC} = 35 MHz, f_s ≤ 2.5 Msps T_A = entire range	-	3	33	
		1.65 V < V_{DDA} = V_{REF+} < 3 V f_{ADC} = 35 MHz, f_s ≤ 2.2 Msps f_{ADC} = 16 MHz; f_s ≤ 1.1 Msps T_A = entire range	-	3	52	

Workaround

Apply one of the following measures:

- Use V_{DDA} and V_{REF+} higher than 3V.
- Use the ADC for measuring a difference between two DC voltages or for measuring AC voltages with a head room to V_{REF+} and V_{SSA} greater than the voltage corresponding to EO(max).
- In the hardware application, make V_{REF+} and V_{SSA} available on the ADC GPIO inputs. Then if CALFACT register value indicates 0x00 or 0x7F after the hardware calibration, compensate A/D conversions by software for the ADC offset remaining after the hardware compensation as follows:
  - For CALFACT equal to 0x00:
    1. With the ADC, measure the V_{REF+} voltage through the GPIO channel. Subtract the A/D conversion result from the full-scale value plus one (4096 for a 12-bit ADC), to obtain the offset value. For example, if the A/D conversion result is 4093, the offset value is 3 (3 LSBs).
    2. Add the offset value to any further A/D conversion result. For example, if the A/D conversion result is 1550, the result after this compensation is 1553.
  - For CALFACT equal to 0x7F:
    1. With the ADC, measure the V_{SSA} voltage through the GPIO channel, to obtain the offset value, for example 4 (4 LSBs).
    2. Subtract the offset value from any further A/D conversion result. For example, if the A/D conversion result is 1550, the result after this compensation is 1546.

Note: For better accuracy of the remaining offset measurement, it is recommended to use an average of multiple (for example eight) A/D conversions.

Note: The compensation for the offset remaining after the hardware calibration reduces the effective ADC conversion range.

2.6 TIM

2.6.1 One-pulse mode trigger not detected in master-slave reset + trigger configuration

Description

The failure occurs when several timers configured in one-pulse mode are cascaded, and the master timer is configured in combined reset + trigger mode with the MSM bit set:

OPM = 1 in TIMx_CR1, SMS[3:0] = 1000 and MSM = 1 in TIMx_SMCR.

The MSM delays the reaction of the master timer to the trigger event, so as to have the slave timers cycle-accurately synchronized.

If the trigger arrives when the counter value is equal to the period value set in the TIMx_ARR register, the one-pulse mode of the master timer does not work and no pulse is generated on the output.

Workaround

None. However, unless a cycle-level synchronization is mandatory, it is advised to keep the MSM bit reset, in which case the problem is not present. The MSM = 0 configuration also allows decreasing the timer latency to external trigger events.

2.6.2 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.

2.6.3 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).

2.6.4 TIM16 and TIM17 are unduly clocked by SYSCLK

Description

The timers TIM16 and TIM17 are unduly clocked by SYSCLK instead of being clocked by the timer clock TIMPCLK. As a consequence, they do not reflect AHB and APB prescaler settings.

The TIM16 and TIM17 are fully functional as long as the SYSCLK-to-PCLK frequency ratio remains smaller than or equal to four.

Workaround

None.

2.7 RTC and TAMP

2.7.1 Calendar initialization may fail in case of consecutive INIT mode entry

Description

If the INIT bit of the RTC_ICSR register is set between one and two RTCCLK cycles after being cleared, the INITF flag is set immediately instead of waiting for synchronization delay (which should be between one and two RTCCLK cycles), and the initialization of registers may fail. Depending on the INIT bit clearing and setting instants versus the RTCCLK edges, it can happen that, after being immediately set, the INITF flag is cleared during one RTCCLK period then set again. As writes to calendar registers are ignored when INITF is low, a write occurring during this critical period might result in the corruption of one or more calendar registers.

Workaround

After existing the initialization mode, clear the BYPSHAD bit (if set) then wait for RSF to rise, before entering the initialization mode again.

Note: It is recommended to write all registers in a single initialization session to avoid accumulating synchronization delays.

2.8 I2C

2.8.1 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.

2.8.2 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.

2.8.3 Spurious master transfer upon own slave address match

Description

When the device is configured to operate at the same time as master and slave (in a multi- master I^2C-bus application), a spurious master transfer may occur under the following condition:

- Another master on the bus is in process of sending the slave address of the device (the bus is busy).
- The device initiates a master transfer by bit set before the slave address match event (the ADDR flag set in the I2C_ISR register) occurs.
- After the ADDR flag is set:
  - the device does not write I2C_CR2 before clearing the ADDR flag, or
  - the device writes I2C_CR2 earlier than three I2C kernel clock cycles before clearing the ADDR flag

In these circumstances, even though the START bit is automatically cleared by the circuitry handling the ADDR flag, the device spuriously proceeds to the master transfer as soon as the bus becomes free. The transfer configuration depends on the content of the I2C_CR2 register when the master transfer starts. Moreover, if the I2C_CR2 is written less than three kernel clocks before the ADDR flag is cleared, the I2C peripheral may fall into an unpredictable state.

Workaround

Upon the address match event (ADDR flag set), apply the following sequence.

Normal mode (SBC = 0):

1. Set the ADDRCF bit.
2. Before Stop condition occurs on the bus, write I2C_CR2 with the START bit low.

Slave byte control mode (SBC = 1):

1. Write I2C_CR2 with the slave transfer configuration and the START bit low.
2. Wait for longer than three I2C kernel clock cycles.
3. Set the ADDRCF bit.
4. Before Stop condition occurs on the bus, write I2C_CR2 again with its current value.

The time for the software application to write the I2C_CR2 register before the Stop condition is limited, as the clock stretching (if enabled), is aborted when clearing the ADDR flag.

Polling the BUSY flag before requesting the master transfer is not a reliable workaround as the bus may become busy between the BUSY flag check and the write into the I2C_CR2 register with the START bit set.

2.9 USART

2.9.1 Data corruption due to noisy receive line

Description

In UART mode with oversampling by 8 or 16 and with 1 or 2 stop bits, the received data may be corrupted if a glitch to zero shorter than the half-bit occurs on the receive line within the second half of the stop bit.

Workaround

None.

2.9.2 USART prescaler feature missing in USART implementation section

Description

Some reference manual revisions may omit the information that the USART prescaler is not present in all USART instances. This information is provided in the USART implementation section of the corresponding reference manual.

This is a documentation issue rather than a product limitation.

Workaround

No application workaround is required or applicable.

2.10 SPI

2.10.1 BSY bit may stay high when SPI is disabled

Description

The BSY flag may remain high upon disabling the SPI while operating in:

- master transmit mode and the TXE flag is low (data register full).
- master receive-only mode (simplex receive or half-duplex bidirectional receive phase) and an SCK strobing edge has not occurred since the transition of the RXNE flag from low to high.
- slave mode and NSS signal is removed during the communication.

Workaround

When the SPI operates in:

- master transmit mode, disable the SPI when TXE = 1 and BSY = 0.
- master receive-only mode, ignore the BSY flag.
- slave mode, do not remove the NSS signal during the communication.

2.10.2 BSY bit may stay high at the end of data transfer in slave mode

Description

BSY flag may sporadically remain high at the end of a data transfer in slave mode. This occurs upon coincidence of internal CPU clock and external SCK clock provided by master.

In such an event, if the software only relies on BSY flag to detect the end of SPI slave data transaction (for example to enter low-power mode or to change data line direction in half-duplex bidirectional mode), the detection fails.

As a conclusion, the BSY flag is unreliable for detecting the end of data transactions.

Workaround

Depending on SPI operating mode, use the following means for detecting the end of transaction:

- When NSS hardware management is applied and NSS signal is provided by master, use NSS flag.
- In SPI receiving mode, use the corresponding RXNE event flag.
- In SPI transmit-only mode, use the BSY flag in conjunction with a timeout expiry event. Set the timeout such as to exceed the expected duration of the last data frame and start it upon TXE event that occurs with the second bit of the last data frame. The end of the transaction corresponds to either the BSY flag becoming low or the timeout expiry, whichever happens first.

Prefer one of the first two measures to the third as they are simpler and less constraining.

Alternatively, apply the following sequence to ensure reliable operation of the BSY flag in SPI transmit mode:

1. Write last data to data register.
2. Poll the TXE flag until it becomes high, which occurs with the second bit of the data frame transfer.
3. Disable SPI by clearing the SPE bit mandatorily before the end of the frame transfer.
4. Poll the BSY bit until it becomes low, which signals the end of transfer.

Note: The alternative method can only be used with relatively fast CPU speeds versus relatively slow SPI clocks or/and long last data frames. The faster is the software execution, the shorter can be the duration of the last data frame.

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-Nov-2018	1	Initial release.
16-Jan-2020	2	Added errata: • DMAMUX cannot be synchronized or triggered by EXTI • Overwriting with all zeros a flash memory location previously programmed with all ones fails • Overrun flag is not set if EOC reset coincides with new conversion end • Writing ADC_CFGR1 register while ADEN bit is set resets RES[1:0] bitfield • Out-of-threshold value is not detected in AWD1 Single mode • TIM16 and TIM17 are unduly clocked by SYSCLK • Data corruption due to noisy receive line • USART prescaler feature missing in USART implementation section Added Table 4. Summary of device documentation errata.
01-Feb-2021	3	Added errata: • Wakeup from Stop not effective under certain conditions • PC13 signal transitions disturb LSE • ADC sampling time might be one cycle longer • ADC trigger latency parameter • Consecutive compare event missed in specific conditions • Output compare clear not working with external counter reset
24-Oct-2022	4	Added: • Section Important security notice • Table 1. Device summary • Section 2.2.8 Boot select after debug interface connection • Section 2.2.9 Corrupted content of the RTC domain due to a missed power-on reset after this domain supply voltage drop Update: • Table 2. Device variants • Section 2.2.5 Overwriting with all zeros a flash memory location previously programmed with all ones fails • Section 2.5.2 Writing ADC_CFGR1 register while ADEN bit is set resets RES[1:0] bitfield • Section 2.5.6 ADC offset may be out of specification Removed the GPIO and LPTIM section headings.

Contents