STM32H743xI Errata sheet

STM32H743xI rev Y device limitations

Applicability

This document applies to the part numbers of STM32H743xI devices listed in *Table 1* and their variants shown in *Table 2*.

Section 1 gives a summary and *Section 2* a description of workarounds for device limitations, with respect to the device datasheet and reference manual RM0433.

Table 1. Device summary

Reference	Part numbers
STM32H743xI	STM32H743VI, STM32H743ZI, STM32H743II, STM32H743BI, STM32H743XI, STM32H743AI

Table 2. Device variants

Reference	Silicon revision codes	
	Device marking(1)	REV_ID(2)
STM32H743xI	Y	0x1003

1. Refer to the device data sheet for how to identify this code on different types of package.
2. REV_ID[15:0] bit field of DBGMCU_IDC register. Refer to the reference manual.

Contents

1 Summary of device limitations

The following table gives a quick references to all documented device limitations of STM32H743xI 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	Next Silicon revision
			Rev. Y	
System	2.2.1	Timer system breaks do not work	N	-
	2.2.2	Clock recovery system synchronization with USB SOF does not work	A	-
	2.2.3	SysTick external clock is not HCLK/8	A	-
	2.2.4	Clock switching does not work when LSE failure is detected by CSS	A	-
	2.2.5	Option byte loading can be done with the user wait-state configuration	A	-
	2.2.6	Flash BusFault address register might not be valid when an ECC double error occurs	A	-
	2.2.7	48 MHz RC oscillator user calibration values are lost after system reset	A	-
	2.2.8	RTC stopped when a system reset occurs while the LSI is used as a clock source	A	-
	2.2.9	Flash memory write sequence error flag PGSERR is not set	A	-
	2.2.10	Flash ECC address register might not be updated	N	-
	2.2.11	Accessing the system memory might stall the system when Flash memory banks are swapped	A	-
	2.2.12	Write flags are not swapped when Flash memory banks are swapped	A	-
	2.2.13	PCROP-protected areas in Flash memory might be unprotected	A	-
	2.2.14	USB OTG_FS PHY drive limit on DP/DM pins	N	-
	2.2.15	Reading from AXI SRAM might lead to data read corruption	A	-
	2.2.16	Conversion overlap might impact the ADC accuracy	A	-
	2.2.17	ADC resolution limited by LSE activity	A	-
	2.2.18	ADC maximum sampling rate when VDDA is lower than 2 V	A	-
	2.2.19	ADC maximum resolution when VDDA is higher than 3.3 V	A	-
	2.2.20	LSE oscillator driving capability selection bits are swapped	A	-
	2.2.21	HRTIM internal synchronization does not work	N	-
FMC	2.3.1	Dummy read cycles inserted when reading synchronous memories	N	N
	2.3.2	Wrong data read from a busy NAND Flash memory	A	A
	2.3.3	Missed clocks with continuous clock feature enabled	A	-
QUADSPI	2.4.1	First nibble of data is not written after a dummy phase	A	-
I2C	2.5.1	Wrong data sampling when data set-up time (tSU;DAT) is smaller than one i2c_ker_ck period	A	A
	2.5.2	NOSTRETCH setting might impact Master mode	A	A
	2.5.3	START bit not cleared when first part of the 10-bit address is not acknowledged	A	A
USART	2.6.1	Underrun flag is set when the USART is used in SPI slave receive mode	A	A

Table 3. Summary of device limitations (continued)

Function	Section	Limitation	Status	Next Silicon revision
			Rev. Y	
SPI	*2.7.1*	*Wrong DMA receive requests might be generated in Half-duplex mode*	A	-
SDMMC	*2.8.1*	*Busy not detected when a write operation suspended during busy phase resumes*	A	-
	2.8.2	*Wrong data line 2 generation between two blocks during DDR transfer with Read wait mode enabled*	A	-
	2.8.3	*Unwanted overrun detection when an AHB error is reported whereas all bytes have been received*	A	-
	2.8.4	*Consecutive multiple block transfers can induce incorrect data length*	A	-
	2.8.5	*Clock stop reported during Read wait mode sequence*	A	-
FDCAN	*2.9.1*	*Writing FDCAN_TTTS during initialization corrupts FDCAN_TTTMC*	A	A
	2.9.2	*Wrong data might be read from Message RAM by the CPU when using two FDCANs*	A	-
HDMI-CEC	*2.10.1*	*Unexpected switch to receive mode without automatic transmission retry and notification*	A	A
	2.10.2	*CEC header not received due to unjustified Rx-Overrun detection*	A	A

2 Description of device limitations

The following sections describe device limitations and provide workarounds if available. They are grouped by device functions.

2.1 Arm® 32-bit Cortex®-M7

There are not limitations related to the Arm® Cortex®-M7 (r1p1) core.

Refer to the following Arm® documents:

- Arm® processor Cortex®-M7 (AT610) and Cortex®-M7 with FPU (AT611) software developer errata notice.
- Arm® embedded trace macrocell CoreSight ETM-M7 (TM975) software developer errata notice.

2.2 System

2.2.1 Timer system breaks do not work

Description

System break sources (processor LOCKUP output, PVD detection, RAM ECC error, Flash ECC error or clock security system detection) do not generate a break event on TIM1, TIM8 and HRTIM.

Workaround

None

2.2.2 Clock recovery system synchronization with USB SOF does not work

Description

The clock recovery system (CRS) synchronization by USB start-of-frame signal (SOF) does not work.

Workaround

When available, use the LSE oscillator as synchronization source.

2.2.3 SysTick external clock is not HCLK/8

Description

The SysTick external clock is the system clock, instead of the system clock divided by 8 (HCLK/8).

Workaround

Use the system clock (HCLK) as external clock and multiply the reload value by 8 in STK_LOAD register (take care that the maximum value is 2^{24}-1).

2.2.4 Clock switching does not work when LSE failure is detected by CSS

Description

When a failure on the LSE oscillator is detected by a clock security system (CSS), the backup domain clock source cannot be changed.

Workaround

When a clock security system detects a LSE failure, reset the backup domain and select a functional clock source.

2.2.5 Option byte loading can be done with the user wait-state configuration

Description

After an option byte change, the option byte loading is performed with the user wait-state configuration instead of the default configuration.

Workaround

When performing option byte loading (modification), configure the correct number of wait-states or use the default value (7 wait states).

2.2.6 Flash BusFault address register might not be valid when an ECC double error occurs

Description

When a first read operation is performed without ECC error and a master accesses data with wait states, if a new access is done and contains an ECC double detection error, then the error message returns the address of the first data which has not generated the error.

Workaround

When a double ECC error flag is raised, check the failing address in the Flash interface (FAIL_ECC_ADDR1/2 in FLASH_ECC_FA1R/FA2R) and disregard the content of the BusFault address register.

2.2.7 48 MHz RC oscillator user calibration values are lost after system reset

Description

When a system reset occurs, the user calibration values of the 48 MHz oscillator (HSI48) are lost.

Note: This limitation is not present on other oscillators.

Workaround

The HSI48 (48 MHz RC) oscillator must be calibrated again after each system reset.

2.2.8 RTC stopped when a system reset occurs while the LSI is used as a clock source

Description

When the LSI clock is used as RTC clock source, the RTC is stopped (it does not received the clock anymore) when a system reset occurs.

Workaround

1. Check the RTC clock source after each system reset.
2. If the LSI clock is selected, enable it again.

2.2.9 Flash memory write sequence error flag PGSERR is not set

Description

When a write protection error is generated (WRPERR1/2 set to 1 in FLASH_CR1/2) without being cleared (CLR_WRPERR1/2 in FLASH_CCR1/2), and it is followed by another write operation, the programming sequence error flag (PGSERR1/2 in FLASH_CR1/2) corresponding to the second write operation is not set.

Workaround

Do not monitor PGSERR1/2. Use WRPERR instead.

2.2.10 Flash ECC address register might not be updated

Description

When two consecutive ECC errors occur, the content of the FLASH_ECC_FA1/2 register cannot be updated if the error correction flag (SNECCERR1/2 or DBECCERR1/2 in FLASH_SR1/2 register) is cleared at the same time as a new ECC error occurs.

Workaround

None.

2.2.11 Accessing the system memory might stall the system when Flash memory banks are swapped

Description

If the user application performs concurrent accesses to system memory and a main memory sector when Flash memory banks are swapped (SWAP_BANK = 1 in FLASH_OPTCR register), the Flash interface might stall the bus as well as the whole system.

Workaround

When Flash memory banks are swapped, accesses to the system memory and Flash memory must be performed in a non-concurrent way.

2.2.12 Write flags are not swapped when Flash memory banks are swapped

Description

When Flash memory banks are swapped, the write status and error flags located in the FLASH_SR1/2 register are sent to the wrong bank.

Workaround

When Flash memory banks are swapped, take into account the write status and error flags that correspond to the bank that is not targeted by the write operation.

2.2.13 PCROP-protected areas in Flash memory might be unprotected

Description

In case of readout protection level regression from level 1 to level 0, the PCROP protected areas in Flash memory might become unprotected.

Workaround

The user application must set the readout protection level to level 2 to avoid PCROP-protected areas from being unprotected.

2.2.14 USB OTG_FS PHY drive limit on DP/DM pins

Description

To avoid damaging parts, the user application must avoid to load more than 5 mA on OTG_FS_DP/DM pins.

Workaround

None

2.2.15 Reading from AXI SRAM might lead to data read corruption

Description

Read data might be corrupted when the following conditions are met:

- Several read transactions are performed to the AXI SRAM,
- and a master delays its data acceptance while a new transfer is requested.

Workaround

Set the READ_ISS_OVERRIDE bit in the AXI_TARG7_FN_MOD register. This will reduce the read issuing capability to 1 at AXI interconnect level and avoid data corruption.

2.2.16 Conversion overlap might impact the ADC accuracy

Description

The following conditions might impact the ADC accuracy

- Several ADC conversions are running simultaneously
- ADC and DAC conversions are running simultaneously

Workaround

Avoid conversion overlapping. The application should ensure that conversions are performed sequentially.

2.2.17 ADC resolution limited by LSE activity

Description

The following ADC3 input pins might be impacted by adjacent LSE activity:

- ADC3 channels on pins PF3 to PF10

Workaround

16-bit and 14-bit data resolutions are not recommended on these pins. This limits data resolution configuration to 8 bits, 10 bits or 12 bits.

2.2.18 ADC maximum sampling rate when V_{DDA} is lower than 2 V

Description

If V_{DDA} is lower than 2 V, the ADC conversion accuracy is not guaranteed over the full ADC sampling rate.

Workaround

The application should avoid a sampling rate higher than 1.5 MSPS when operating with V_{DDA} below 2 V.

2.2.19 ADC maximum resolution when V_{DDA} is higher than 3.3 V

Description

If V_{DDA} is higer than 3.3V, the ADC conversion accuracy is not guaranteed for all data resolutions.

Workaround

16-bit, 14-bit and 12-bit data resolutions are not useful in this configuration. This limits available data resolution configuration to 8 bits and 10 bits.

2.2.20 LSE oscillator driving capability selection bits are swapped

Description

The LSEDRV[1:0] bits in the RCC_BDCR register, which are used to select LSE oscillator driving capability, are swapped (see *Table 4*).

Table 4. Expected vs effective LSE driving mode

LSEDRV[1:0]	LSE driving mode	
	Expected mode	Effective mode
01	Medium-low drive	Medium-high drive
10	Medium-high drive	Medium-low drive

Workaround

- Use LSEDRV[1:0]=01 to select LSE medium-high drive
- Use LSEDRV[1:0]=10 to select LSE medium-low drive

2.2.21 HRTIM internal synchronization does not work

Description

HRTIM synchronization input source from internal event (SYNCIN[1:0]=10 in the HRTIM_MCR register) does not work. Consequently, it is not possible to use the on-chip TIM1_TRGO output as synchronization event for HRTIM.

Workaround

None.

2.3 FMC

2.3.1 Dummy read cycles inserted when reading synchronous memories

Description

When performing a burst read access from a synchronous memory, two dummy read accesses are performed at the end of the burst cycle whatever the type of AXI burst access.

However, the extra data values that are read are not used by the FSMC and there is no functional failure.

Workaround

None.

2.3.2 Wrong data read from a busy NAND Flash memory

Description

When a read command is issued to the NAND memory, the R/B signal gets activated upon the de-assertion of the chip select. If a read transaction is pending, the NAND controller might not detect the R/B signal (connected to NWAIT) previously asserted and sample a wrong data. This problem occurs only when the MEMSET timing is configured to 0x00 or when ATTHOLD timing is configured to 0x00 or 0x01.

Workaround

Either configure MEMSET timing to a value greater than 0x00 or ATTHOLD timing to a value greater than 0x01.

2.3.3 Missed clocks with continuous clock feature enabled

Description

When the continuous clock feature is enabled, the FMC_CLK clock can be switched OFF in the following conditions:

- The FMC_CLK clock is divided by 2.
- An asynchronous byte transaction is performed on an FMC bank configured in 32-bit memory data width.

Note: When the FMC_CLK clock is switched OFF on static memories, it can be switched ON by issuing a synchronous transaction or any asynchronous transaction different from a byte access on 32-bit data bus width.

Workaround

When the continuous clock feature is enabled, do not use the FMC_CLK clock divider ratio of 2 when issuing a byte transaction to 32-bit asynchronous memories.

2.4 QUADSPI

2.4.1 First nibble of data is not written after a dummy phase

Description

The first nibble of data to be written to the external Flash memory is lost in the following conditions:

- The QUADSPI is used in the indirect write mode,
- and at least one dummy cycle is used.

Workaround

Use an alternate-bytes phase instead of a dummy phase in order to add a latency period between the address phase and the data phase. This workaround works only if the number of dummy cycles corresponds to a multiple of 8 bits of data.

As an example:

- To generate 1 dummy cycle, send 1 alternate-byte in 4 data line DDR mode or Dual-flash SDR mode.
- To generate 2 dummy cycles, send 1 alternate-byte in 4 data line SDR mode
- To generate 4 dummy cycles, send 2 alternate-bytes in 4 data line SDR mode or send 1 alternate-byte in 2 data line SDR mode
- To generate 8 dummy cycles, send 1 alternate-byte in 1 data line SDR mode.

2.5 I2C

2.5.1 Wrong data sampling when data set-up time ($t_{SU;DAT}$) is smaller than one i2c_ker_ck period

Description

The I2C bus specification and user manual specifies a minimum data set-up time ($t_{SU;DAT}$) at:

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

The I2C SDA line is not correctly sampled when $t_{SU;DAT}$ is smaller than one i2c_ker_ck (I2C clock) period: the previous SDA value is sampled instead of the current one. This can result in a wrong slave address reception, a wrong received data byte, or a wrong received acknowledge bit.

Workaround

Increase the i2c_ker_ck frequency to obtain an i2c_ker_ck period smaller than the transmitter minimum data setup time, or, if it is possible, increase the transmitter minimum data setup time.

2.5.2 NOSTRETCH setting might impact Master mode

Description

When the I2C operates in Master mode and the Clock stretching is disabled (NOSTRETCH = 1 in I2C_CR1), the setup time programmed in SCLDEL[3:0] bits of I2C_TIMINGR is not applied on the bus.

Consequently, when the clock is stretched, the SDA data line changes simultaneously with the SCL rising edge.

Workaround

Clock stretching should not be disabled when the I2C is used as master since in multimaster networks, all the masters need to be able to stretch the bus clock to manage arbitration.

When the I2C is used as master, clear NOSTRETCH bit in I2C_CR1.

2.5.3 START bit not cleared when first part of the 10-bit address is not acknowledged

Description

In Master mode, when the slave does not acknowledge a byte during address transmission (NACKF bit set in I2C_ISR), the master automatically sends a STOP condition. The byte can either belong to the 10-bit header or to the 8 address LSBs in case of read (RD_WRN = 1).

The I2C should then be able to start a new transfer. However, this is not the case because the START bit has not been cleared by the hardware.

Workaround

1. Wait for STOP condition detection (STOPF = 1 in I2C_ISR).
2. Disable the I2C interface.
3. Wait for a minimum of three APB cycles.
4. Enable again the I2C interface.

2.6 USART

2.6.1 Underrun flag is set when the USART is used in SPI slave receive mode

Description

When the USART is used in SPI slave receive mode, the underrun flag (UDR bit in USART_ISR register) might be set even if the transmitter is disabled (TE bit set to 0 in USAR_CR1 register).

Workaround

Three workarounds are possible

- Ignore the UDR flag when the transmitter is disabled.
- Clear the UDR flag every time it is set, even if the Transmitter is disabled.
- Write dummy data in the USART_TDR register to avoid setting the UDR flag.

2.7 SPI

2.7.1 Wrong DMA receive requests might be generated in Half-duplex mode

Description

In Half-duplex mode, when a DMA receive transfer is configured after a transmit-only transfer, the transmit operation completes successfully. However, once the RXDMAEN bit is set in SPI_CFG1 register, the SPI interface might trigger wrong DMA receive requests without data in the receive FIFO.

Workaround

Several workarounds are possible, depending on the application:

- Read the first data from the receive transfer before enabling DMA transfers.
- Perform a hardware reset of the SPI controller before starting DMA receive transfers.

2.8 SDMMC

2.8.1 Busy not detected when a write operation suspended during busy phase resumes

Description

When a card accepts a suspend command during a block write operation busy phase, the card might drive the data line 0 (SDMMC_D0) when the write transfer is resumed. The SDMMC does not detect that the data line 0 is Low when the write transfer resumes.

Workaround

To suspend a write transfer:

1. Set DTHOLD bit in the SDMMC_CMDR register.
2. Wait till the DHOLD status flag is set in SDMMC_STAR register to make sure the busy line has been released.
3. Send a suspend command to the card (CMDSUSPEND = 1, CMDTRANS = 0 and CPSMEN = 1 in SDMMC_CMDR).

2.8.2 Wrong data line 2 generation between two blocks during DDR transfer with Read wait mode enabled

Description

The Read wait mode allows suspending an SDIO multiple block read operation when the host is not ready to receive the next bytes. The host can request the card to suspend temporarily the transfer by driving data line 2 (SDMMC_D2) low between two blocks.

When a double data rate (DDR) read operation is ongoing, data line 2 is not driven low but toggles constantly. Consequently, some bytes are not received and a CRC error failure is reported.

Workaround

Use the clock stretching method (RWMOD = 1) instead of data line 2 to suspend temporarily the transfer between two blocks.

2.8.3 Unwanted overrun detection when an AHB error is reported whereas all bytes have been received

Description

When the internal DMA is used and a write transfer initiated by the SDMMC on the AHB fails, the IDMATE flag is set in SDMMC_STAR and the transfer is aborted by flushing the FIFO.

When an AHB error occurs on the three last bursts of a successful read transfer, the FIFO is considered as empty (DATAEND flag set in SDMMC_STAR) but some bytes, not yet transferred to the FIFO, might still be present in the internal receive buffer. As a result, the following read operation will fail and report an overrun error.

Workaround

1. When DATAEND = 1, check IDMATE flag.
2. If IDMATE = 1 and DTDIR = 1 in SDMMC_DCTRL, reset SDMMC.

2.8.4 Consecutive multiple block transfers can induce incorrect data length

Description

When a new transfer is started by setting the DTEN bit in SDMMC_DCTRL control register while less than eight SDMMC clock cycles elapsed since the end of the previous transfer, the second transfer is performed with the number of blocks configured for the previous transfer. This is due to the fact that the new number of data to be transferred has not been reloaded in the internal data block counter.

Workaround

The user application must ensure that at least 8 SDMMC clock cycles elapsed between the successful completion of a transfer and the moment DTEN bit is set.

2.8.5 Clock stop reported during Read wait mode sequence

Description

When the SDMMC clock is stopped at low level, CKSTOP flag might be wrongly set in the SDMMC_STAR register.

Workaround

When the multiple block transfer completes (DATAEND = 1 in SDMMC_STAR), simultaneously set CKSTOPC and DATAENDC to 1 in SDMMC_ICR register.

2.9 FDCAN

2.9.1 Writing FDCAN_TTTS during initialization corrupts FDCAN_TTTMC

Description

During TTCAN initialization, writing to FDCAN TT Trigger Select Register (FDCAN_TTTS) also affects FDCAN TT Trigger Memory Configuration Register (FDCAN_TTTMC).

Workaround

The user application must avoid writing to FDCAN_TTTS register during TTCAN initialization phase.

Note: Outside of TTCAN initialization phase, write operations to FDCAN_TTTS do not impact FDCAN_TTTMC since this register is write protected.

2.9.2 Wrong data might be read from Message RAM by the CPU when using two FDCANs

Description

When using two FDCAN controllers, and the CPU and FDCANs simultaneously request read accesses from Message RAM, the CPU read request might return erroneous data.

The issue is not present if the CPU requests write access to Message RAM.

Workaround

To avoid concurrent read accesses between the CPU and FDCANs, use only one FDCAN at a time.

2.10 HDMI-CEC

2.10.1 Unexpected switch to receive mode without automatic transmission retry and notification

Description

If the HDMI-CEC peripheral starts a transmit operation, and at the same time another CEC initiator attempts to perform the same operation but with a wrong start-bit timing and/or without following the signal free timing rules, the transmission is aborted and never restarts, and the HDMI switch to receive mode. The user application is not informed of the bus error.

Workaround

Use transmission timeout: when a timeout occurs, to enable again the transmission of the pending message, disable and enable again the HDMI-CEC peripheral to clear TXSOM bit in CEC_CR register.

2.10.2 CEC header not received due to unjustified Rx-Overrun detection

Description

In a multinode CEC network, two messages are sent to two different followers. The CEC device which is the destination of the second message should ignore the first message and receive the second one. However, the header of the second message is not received and an Rx-Overrun is wrongly detected and signaled by setting the RXOVR flag in the CEC_ISR register and the RXOVRIE interrupt in the CEC_IER register.

The issue is not present:

- in Listen mode,
- when the first message is a broadcast message
- when the first message is a ping message (header only),
- or when the first message header is not acknowledged.

Workaround

Configure the HDMI-CEC to operate in Listen mode and apply message filtering based on destination address. In this case all the messages sent over the CEC bus will be received and it is up to the user application to discard the messages that are not sent to its address or that are broadcast.

3 Revision history

Table 5. Document revision history

Date	Revision	Changes
19-Jun-2017	1	Initial release.
2-Nov-2017	2	Added STM32H743AI part number. Removed JPEG limitation. Updated *Section 2.3.1: Dummy read cycles inserted when reading synchronous memories* and *Section 2.3.2: Wrong data read from a busy NAND Flash memory*. Added *Section 2.9.2: Wrong data might be read from Message RAM by the CPU when using two FDCANs*.