STM32F105xx and STM32F107xx Errata sheet

STM32F105xx and STM32F107xx revision Z connectivity line device limitations

Silicon identification

This errata sheet applies to the revision Z of the STMicroelectronics STM32F105xx and STM32F107xx connectivity line products. This family features an ARM™ 32-bit Cortex®-M3 core, for which an errata notice is also available (see *Section 1* for details).

The full list of part numbers is shown in *Table 2*. The products are identifiable as shown in *Table 1*:

- by the revision code marked below the sales type on the device package
- by the last three digits of the internal sales type printed on the box label

Table 1. Device Identification(1)

Sales type	Revision code(2) marked on device
STM32F105xx	“Z”
STM32F107xx	“Z”

1. The REV_ID bits in the DBGMCU_IDCODE register show the revision code of the device (see the STM32F10xxx reference manual for details on how to find the revision code).
2. Refer to *Appendix A: Revision and date codes on device marking* for details on how to identify the revision code on the different packages.

Table 2. Device summary

Reference	Part number
STM32F105xx	STM32F105R8, STM32F105V8 STM32F105RB, STM32F105VB STM32F105RC, STM32F105VC
STM32F107xx	STM32F107RB, STM32F107VB STM32F107RC, STM32F107VC

Contents

1 ARM™ 32-bit Cortex®-M3 limitations 4

1.1 Cortex-M3 limitation description for the STM32F105xx / STM32F107xx connectivity line devices 4

1.1.1 Cortex-M3 LDRD with base in list may result in incorrect base register when interrupted or faulted 5

1.1.2 Cortex-M3 event register is not set by interrupts and debug 5

1.1.3 Cortex-M3 BKPT in debug monitor mode can cause DFSR mismatch . . 5

1.1.4 Cortex-M3 may freeze for SLEEPONEXIT single instruction ISR 5

2 STM32F10xxx silicon limitations 7

2.1 Voltage glitch on ADC input 0 9

2.2 Flash memory read after WFI/WFE instruction 9

2.3 Alternate function 9

2.3.1 SPI1 in slave mode and USART2 in synchronous mode 10

2.3.2 SPI1 in master mode and USART2 in synchronous mode 10

2.3.3 SPI2 in slave mode and USART3 in synchronous mode 10

2.3.4 SPI2 in master mode and USART3 in synchronous mode 11

2.3.5 I2S2 in master/slave mode and Ethernet/USART3 in synchronous mode 11

2.3.6 USARTx_TX pin usage 11

2.4 Boundary scan TAP: wrong pattern sent out after the “capture IR” state . 12

2.5 Flash memory BSY bit delay versus STRT bit setting 12

2.6 I^2C peripheral 12

2.6.1 Some software events must be managed before the current byte is being transferred 12

2.6.2 SMBus standard not fully supported 13

2.6.3 Start cannot be generated after a misplaced Stop 13

2.6.4 Mismatch on the “Setup time for a repeated Start condition” timing parameter 13

2.6.5 Data valid time ($t_{VD;DAT}$) violated without the OVR flag being set 14

2.7 SPI peripheral 14

2.7.1 CRC still sensitive to communication clock when SPI is in slave mode even with NSS high 14

2.8 General-purpose timers 15

2.8.1 Missing capture flag 15

2.8.2 Overcapture detected too early . . . 15
2.8.3 General-purpose timer: regulation for 100% PWM . . . 15
2.9 LSI clock stabilization time . . . 16
2.10 PLL not locking when sourced by HSI/2 after reset if it was previously sourced by HSE with predivider >1 or PLL2 . . . 16
2.11 OTG_FS . . . 16
2.11.1 Data in RxFIFO are overwritten when all channels are disabled simultaneously . . . 16
2.11.2 OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured . . . 17
2.11.3 Host channel-halted interrupt not generated when the channel is disabled . . . 17
2.11.4 Error in software-read OTG_FS_DCFG register values . . . 17
2.12 Ethernet MAC . . . 18
2.12.1 Possible underflow when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode . . . 18
2.12.2 Possible CRC error when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode with transmit checksum offload enabled . . . 18
2.12.3 Erroneous automatic checksum insertion after TxFIFO is dynamically switched from Threshold to Store-and-Forward mode . . . 19
2.12.4 Erroneous automatic checksum insertion after a large frame (longer than the TxFIFO) transmission . . . 19
2.12.5 In half-duplex mode, the MAC transmitter ignores collisions after it is disabled during a frame transmission . . . 19
2.12.6 Interrupt due to an RMON (MMC) counter may be set again after it is cleared . . . 20
2.12.7 Incorrect layer 3 (L3) checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads . . . 20
2.12.8 The Ethernet MAC processes invalid extension headers in the received IPv6 frames . . . 20
2.12.9 MAC stuck in the Idle state on receiving the TxFIFO flush command exactly 1 clock cycle after a transmission completes . . . 21
2.12.10 Transmit frame data corruption . . . 21
2.13 Boot loader unavailability on STM32F105xx and STM32F107xx devices with a date code below 937 . . . 22

Appendix A Revision and date codes on device marking . . . 23

Revision history . . . 25

1 ARM™ 32-bit Cortex®-M3 limitations

An errata notice of the STM32F10xxx core is available from the following web address: *http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.eat0420a/*.
The direct link to the errata notice pdf is:
http://infocenter.arm.com/help/topic/com.arm.doc.eat0420a/Cortex-M3-Errata-r1p1-v0.2.pdf.

All the described limitations are minor and related to the revision r1p1-01rel0 of the Cortex-M3 core. *Table 3* summarizes these limitations and their implications on the behavior of the STM32F105xx / STM32F107xx connectivity line devices.

Table 3. Cortex-M3 core limitations and impact on microcontroller behavior

ARM ID	ARM category	ARM summary of errata	Impact on STM32F105xx / STM32F107xx connectivity line devices
602117	Cat 2	LDRD with base in list may result in incorrect base register when interrupted or faulted	Minor
563915	Cat 2	Event register is not set by interrupts and debug	Minor
531064	impl	SWJ-DP missing POR reset sync	No
511864	Cat 3	Cortex-M3 may fetch instructions using incorrect privilege on return from an exception	No
532314	Cat 3	DWT CPI counter increments during sleep	No
538714	Cat 3	Cortex-M3 TPIU clock domain crossing	No
548721	Cat 3	Internal write buffer could be active whilst asleep	No
463763	Cat 3	BKPT in debug monitor mode can cause DFSR mismatch	Minor
463764	Cat 3	Core may freeze for SLEEPONEXIT single instruction ISR	Minor
463769	Cat 3	Unaligned MPU fault during a write may cause the wrong data to be written to a successful first access	No

1.1 Cortex-M3 limitation description for the STM32F105xx / STM32F107xx connectivity line devices

Only the limitations described below have an impact, even though minor, on the implementation of STM32F105xx / STM32F107xx connectivity line devices.

All the other limitations described in the ARM errata notice (and summarized in *Table 3* above) have no impact and are not related to the implementation of the STM32F105xx / STM32F107xx connectivity line devices (Cortex-M3 r1p1-01rel0).

1.1.1 Cortex-M3 LDRD with base in list may result in incorrect base register when interrupted or faulted

Description

The Cortex-M3 Core has a limitation when executing an LDRD instruction from the system-bus area, with the base register in a list of the form LDRD Ra, Rb, [Ra, #imm]. The execution may not complete after loading the first destination register due to an interrupt before the second loading completes or due to the second loading getting a bus fault.

Workarounds

1. This limitation does not impact the STM32F10xxx code execution when executing from the embedded Flash memory, which is the standard use of the microcontroller.
2. Use the latest compiler releases. As of today, they no longer generate this particular sequence. Moreover, a scanning tool is provided to detect this sequence on previous releases (refer to your preferred compiler provider).

1.1.2 Cortex-M3 event register is not set by interrupts and debug

Description

When interrupts related to a WFE occur before the WFE is executed, the event register used for WFE wakeup events is not set and the event is missed. Therefore, when the WFE is executed, the core does not wake up from WFE if no other event or interrupt occur.

Workaround

Use STM32F10xxx external events instead of interrupts to wake up the core from WFE by configuring an external or internal EXTI line in event mode.

1.1.3 Cortex-M3 BKPT in debug monitor mode can cause DFSR mismatch

Description

A BKPT may be executed in debug monitor mode. This causes the debug monitor handler to be run. However, the bit 1 in the Debug fault status register (DFSR) at address 0xE000ED30 is not set to indicate that it was originated by a BKPT instruction. This only occurs if an interrupt other than the debug monitor is already being processed just before the BKPT is executed.

Workaround

If the DFSR register does not have any bit set when the debug monitor is entered, this means that we must be in this “corner case” and so, that a BKPT instruction was executed in debug monitor mode.

1.1.4 Cortex-M3 may freeze for SLEEPONEXIT single instruction ISR

Description

If the Cortex-M3 SLEEPONEXIT functionality is used and the concerned interrupt service routine (ISR) contains only a single instruction, the core becomes frozen. This freezing may

occur if only one interrupt is active and it is preempted by an interrupt whose handler only contains a single instruction.

However, any new interrupt that causes a preemption would cause the core to become unfrozen and behave correctly again.

Workaround

This scenario does not happen in real application systems since all enabled ISRs should at least contain one instruction. Therefore, if an empty ISR is used, then insert an NOP or any other instruction before the exit instruction (BX or BLX).

2 STM32F10xxx silicon limitations

Table 4 gives quick references to all documented limitations.

Table 4. Summary of silicon limitations in revision Z devices

Links to silicon limitations	
Section 2.1: Voltage glitch on ADC input 0	
Section 2.2: Flash memory read after WFI/WFE instruction	
Section 2.3: Alternate function	*Section 2.3.1: SPI1 in slave mode and USART2 in synchronous mode*
	Section 2.3.2: SPI1 in master mode and USART2 in synchronous mode
	Section 2.3.3: SPI2 in slave mode and USART3 in synchronous mode
	Section 2.3.4: SPI2 in master mode and USART3 in synchronous mode
	Section 2.3.5: I2S2 in master/slave mode and Ethernet/USART3 in synchronous mode
	Section 2.3.6: USARTx_TX pin usage
Section 2.4: Boundary scan TAP: wrong pattern sent out after the “capture IR” state	
Section 2.5: Flash memory BSY bit delay versus STRT bit setting	
Section 2.6: I2C peripheral	*Section 2.6.1: Some software events must be managed before the current byte is being transferred*
	Section 2.6.2: SMBus standard not fully supported
	Section 2.6.3: Start cannot be generated after a misplaced Stop
	Section 2.6.4: Mismatch on the “Setup time for a repeated Start condition” timing parameter
	Section 2.6.5: Data valid time (tVD;DAT) violated without the OVR flag being set
Section 2.7: SPI peripheral	*Section 2.7.1: CRC still sensitive to communication clock when SPI is in slave mode even with NSS high*
Section 2.8: General-purpose timers	*Section 2.8.1: Missing capture flag*
	Section 2.8.2: Overcapture detected too early
	Section 2.8.3: General-purpose timer: regulation for 100% PWM
Section 2.9: LSI clock stabilization time	
Section 2.10: PLL not locking when sourced by HSI/2 after reset if it was previously sourced by HSE with predivider >1 or PLL2	

Table 4. Summary of silicon limitations in revision Z devices (continued)

Links to silicon limitations	
Section 2.11: OTG_FS	*Section 2.11.1: Data in RxFIFO are overwritten when all channels are disabled simultaneously*
	Section 2.11.2: OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured
	Section 2.11.3: Host channel-halted interrupt not generated when the channel is disabled
	Section 2.11.4: Error in software-read OTG_FS_DCFG register values
Section 2.12: Ethernet MAC	*Section 2.12.1: Possible underflow when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode*
	Section 2.12.2: Possible CRC error when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode with transmit checksum offload enabled
	Section 2.12.3: Erroneous automatic checksum insertion after TxFIFO is dynamically switched from Threshold to Store-and-Forward mode
	Section 2.12.4: Erroneous automatic checksum insertion after a large frame (longer than the TxFIFO) transmission
	Section 2.12.5: In half-duplex mode, the MAC transmitter ignores collisions after it is disabled during a frame transmission
	Section 2.12.6: Interrupt due to an RMON (MMC) counter may be set again after it is cleared
	Section 2.12.7: Incorrect layer 3 (L3) checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads
	Section 2.12.8: The Ethernet MAC processes invalid extension headers in the received IPv6 frames
	Section 2.12.9: MAC stuck in the Idle state on receiving the TxFIFO flush command exactly 1 clock cycle after a transmission completes
	Section 2.12.10: Transmit frame data corruption
Section 2.13: Boot loader unavailability on STM32F105xx and STM32F107xx devices with a date code below 937	

2.1 Voltage glitch on ADC input 0

Description

A low-amplitude voltage glitch may be generated (on ADC input 0) on the PA0 pin, when the ADC is converting with injection trigger. It is generated by internal coupling and synchronized to the beginning and the end of the injection sequence, whatever the channel(s) to be converted.

The glitch amplitude is less than 150 mV with a typical duration of 10 ns (measured with the I/O configured as high-impedance input and left unconnected). If PA0 is used as a digital output, this has no influence on the signal. If PA0 is used has a digital input, it will not be detected as a spurious transition, providing that PA0 is driven with an impedance lower than 5 kΩ. This glitch does not have any influence on the remaining port A pin or on the ADC conversion injection results, in single ADC configuration.

When using the ADC in dual mode with injection trigger, and in order to avoid any side effect, it is advised to distribute the analog channels so that Channel 0 is configured as an injected channel.

Workaround

None.

2.2 Flash memory read after WFI/WFE instruction

Conditions

- Flash prefetch on
- Flash memory timing set to 2 wait states
- FLITF clock stopped in Sleep mode

Description

If a WFI/WFE instruction is executed during a Flash memory access and the Sleep duration is very short (less than 2 clock cycles), the instruction fetch from the Flash memory may be corrupted on the next wakeup event.

Workaround

When using the Flash memory with two wait states and prefetch on, the FLITF clock must *not* be stopped during the Sleep mode – the FLITFEN bit in the RCC_AHBENR register must be set (keep the reset value).

2.3 Alternate function

In some specific cases, some potential weakness may exist between alternate functions mapped onto the same pin.

2.3.1 SPI1 in slave mode and USART2 in synchronous mode

Conditions

- SPI1 and USART2 are clocked
- I/O port pin PA4 is configured as an alternate function output.

Description

USART2 cannot be used in synchronous mode (USART2_CK signal), if SPI1 is used in slave mode.

Workaround

None.

2.3.2 SPI1 in master mode and USART2 in synchronous mode

Conditions

- SPI1 and USART2 are clocked
- I/O port pin PA4 is configured as an alternate function output.

Description

USART2 cannot be used in synchronous mode (USART2_CK signal) if SPI1 is used in master mode and SP1_NSS is configured in software mode. In this case USART2_CK is not output on the pin.

Workaround

In order to output USART2_CK, the SSOE bit in the SPI1_CR2 register must be set to configure the pin in output mode.

2.3.3 SPI2 in slave mode and USART3 in synchronous mode

Conditions

- SPI2 and USART3 are clocked
- I/O port pin PB12 is configured as an alternate function output.

Description

USART3 cannot be used in synchronous mode (USART3_CK signal) if SPI2 is used in slave mode.

Workaround

None.

2.3.4 SPI2 in master mode and USART3 in synchronous mode

Conditions

- SPI2 and USART3 are clocked
- I/O port pin PB12 is configured as an alternate function output.

Description

USART3 cannot be used in synchronous mode (USART3_CK signal) if SPI2 is used in master mode and SP2_NSS is configured in software mode. In this case USART3_CK is not output on the pin.

Workaround

In order to output USART3_CK, the SSOE bit in the SPI2_CR2 register must be set to configure the pin in output mode.

2.3.5 I2S2 in master/slave mode and Ethernet/USART3 in synchronous mode

Conditions

- USART3 in synchronous mode or Ethernet is clocked
- I2S2 is not clocked
- I/O port pin PB12 is configured as an alternate function output

Description

If I2S2 was used prior to operating USART3 in synchronous mode or the Ethernet, a conflict occurs between the I2S2_WS and the ETH_MII_TXD0 / USART3_CK signals even though the I2S2 clock was disabled.

Workaround

To use USART3 in synchronous mode, first disable the I2S2 clock, then perform a software reset of SPI2(I2S2).

To use the Ethernet, first disable the I2S2 clock, then either perform a software reset of SPI2(I2S2) or switch off the I2S mode of SPI2.

2.3.6 USARTx_TX pin usage

Description

In USART receive-mode-only communication (TE = 0 in the USARTx_CR1 register), even when the USARTx_TX pin is not being used, the corresponding I/O port pin cannot be used to output another alternate function (in this mode the USARTx_TX output is set to 1 and thus no other alternate function output can be used).

This limitation applies to all USARTx_TX pins that share another alternate function output.

Workaround

Do not use the corresponding I/O port of the USARTx_TX pin in alternate function output mode. Only the input mode can be used (TE bit in the USARTx_CR1 has to be cleared).

2.4 Boundary scan TAP: wrong pattern sent out after the “capture IR” state

Description

After the “capture IR” state of the boundary scan TAP, the two lower significant bits in the instruction register should be loaded with “01” for them to be shifted out whenever a next instruction is shifted in.

However, the boundary scan TAP shifts out the latest value loaded into the instruction register, which could be “00”, “01”, “10” or “11”.

Workaround

The data shifted out, after the capture IR state, in the boundary scan flow should therefore be ignored and the software should check not only the two least significant bits (XXX01) but all register bits (XXXXX).

2.5 Flash memory BSY bit delay versus STRT bit setting

Description

When the STRT bit in the Flash memory control register is set (to launch an erase operation), the BSY bit in the Flash memory status register goes high one cycle later.

Therefore, if the FLASH_SR register is read immediately after the FLASH_CR register is written (STRT bit set), the BSY bit is read as 0.

Workaround

Read the BSY bit at least one cycle after setting the STRT bit.

2.6 I^2C peripheral

2.6.1 Some software events must be managed before the current byte is being transferred

Description

When the EV7, EV7_1, EV6_1, EV2, EV8, and EV3 events are not managed before the current byte is being transferred, problems may be encountered such as receiving an extra byte, reading the same data twice or missing data.

Workarounds

When it is not possible to manage the EV7, EV7_1, EV6_1, EV2, EV8, and EV3 events before the current byte transfer and before the acknowledge pulse when changing the ACK control bit, it is recommended to:

1. use the I^2C with DMA in general, except when the Master is receiving a single byte
2. use I^2C interrupts and boost their priorities to the highest one in the application to make them uninterruptible

2.6.2 SMBus standard not fully supported

Description

The I^2C peripheral is not fully compliant with the SMBus v2.0 standard since It does not support the capability to NACK an invalid byte/command.

Workarounds

A higher-level mechanism should be used to verify that a write operation is being performed correctly at the target device, such as:

1. Using the SMBA pin if supported by the host
2. the alert response address (ARA) protocol
3. the Host notify protocol

2.6.3 Start cannot be generated after a misplaced Stop

Description

If a master generates a misplaced Stop on the bus (bus error), the peripheral cannot generate a Start anymore.

Workaround

In the I²C standard, it is allowed to send a Stop only at the end of the full byte (8 bits + acknowledge), so this scenario is not allowed. Other derived protocols like CBUS allow it, but they are not supported by the I²C peripheral.

A software workaround consists in asserting the software reset using the SWRST bit in the I2C_CR1 control register.

2.6.4 Mismatch on the “Setup time for a repeated Start condition” timing parameter

Description

In case of a repeated Start, the “Setup time for a repeated Start condition” (named Tsu;sta in the I²C specification) can be slightly violated when the I²C operates in Master Standard mode at a frequency between 88 kHz and 100 kHz.

The issue can occur only in the following configuration:

- in Master mode
- in Standard mode at a frequency between 88 kHz and 100 kHz (no issue in Fast-mode)
- SCL rise time:
  - If the slave does not stretch the clock and the SCL rise time is more than 300 ns (if the SCL rise time is less than 300 ns the issue cannot occur)
  - If the slave stretches the clock

The setup time can be violated independently of the APB peripheral frequency.

Workaround

Reduce the frequency down to 88 kHz or use the I²C Fast-mode if supported by the slave.

2.6.5 Data valid time ($t_{VD;DAT}$) violated without the OVR flag being set

Description

The data valid time ($t_{VD;DAT}$, $t_{VD;ACK}$) described by the I²C standard can be violated (as well as the maximum data hold time of the current data ($t_{HD;DAT}$)) under the conditions described below. This violation cannot be detected because the OVR flag is not set (no transmit buffer underrun is detected).

This issue can occur only under the following conditions:

- in Slave transmit mode
- with clock stretching disabled (NOSTRETCH=1)
- if the software is late to write the DR data register, but not late enough to set the OVR flag (the data register is written before)

Workaround

If the master device allows it, use the clock stretching mechanism by programming the bit NOSTRETCH=0 in the I2C_CR1 register.

If the master device does not allow it, ensure that the software is fast enough when polling the TXE or ADDR flag to immediately write to the DR data register. For instance, use an interrupt on the TXE or ADDR flag and boost its priority to the higher level.

2.7 SPI peripheral

2.7.1 CRC still sensitive to communication clock when SPI is in slave mode even with NSS high

Description

When the SPI is configured in slave mode with the CRC feature enabled, the CRC is calculated even if the NSS pin deselects the SPI (high level applied on the NSS pin).

Workaround

The CRC has to be cleared on both Master and Slave sides between the slave deselection (high level on NSS) and the slave selection (low level on NSS), in order to resynchronize the Master and Slave for their respective CRC calculation.

To procedure to clear the CRC is the following:

1. disable the SPI (SPE = 0)
2. clear the CRCEN bit
3. set the CRCEN bit
4. enable the SPI (SPE = 1)

2.8 General-purpose timers

2.8.1 Missing capture flag

Description

In capture mode, when a capture occurs while the CCRx register is being read, the capture flag (CCxIF) may be cleared without the overcapture flag (CCxOF) being set. The new data are actually captured in the capture register.

Workaround

An external interrupt can be enabled on the capture I/O just before reading the capture register (in the capture interrupt), and disabled just after reading the captured data. Possibly, a missed capture will be detected by the EXTI peripheral.

2.8.2 Overcapture detected too early

Description

In capture mode, the overcapture flag (CCxOF) can be set even though no data have been lost.

Conditions

If a capture occurs while the capture register is being read, an overcapture is detected even though the previously captured data are correctly read and the new data are correctly stored into the capture register.

The system is at the limit of an overcapture but no data are lost.

Workaround

None.

2.8.3 General-purpose timer: regulation for 100% PWM

Description

When the OCREF_CLR functionality is activated, the OCxREF signal becomes de-asserted (and consequently OCx is deasserted / OCxN is asserted) when a high level is applied on

the OCREF_CLR signal. The PWM then restarts (output re-enabled) at the next counter overflow.

But if the PWM is configured at 100% (CCxR > ARR), then it does not restart and OCxREF remains de-asserted.

Workaround

None.

2.9 LSI clock stabilization time

Description

When the LSIRDY flag is set, the clock may still be out of the specified frequency range (f_{LSI} parameter, see LSI oscillator characteristics in the product datasheet).

Workaround

To have a fully stabilized clock in the specified range, a software temporization of 100 µs should be added.

2.10 PLL not locking when sourced by HSI/2 after reset if it was previously sourced by HSE with predivider >1 or PLL2

Description

The limitation occurs when the sequence below is followed:

- PLL source: HSI/2, SYSCLK source: PLL
- PLL source: HSE with predivider >1 or PLL2, SYSCLK source: PLL
- system reset
- PLL source: HSI/2, SYSCLK source: PLL

The PLL cannot be locked when sourced by HSI/2 after applying system reset if it was previously sourced by HSE with predivider >1 or by PLL2.

Workaround

Enable the HSE oscillator and let the PLL lock on it before switching the PLL source to HSI/2.

2.11 OTG_FS

2.11.1 Data in RxFIFO are overwritten when all channels are disabled simultaneously

Description

If the available RxFIFO is just large enough to host 1 packet + its data status, and is currently occupied by the last received data + its status and, at the same time, the

application requests that more IN channels be disabled, the OTG_FS peripheral does not first check for available space before inserting the disabled status of the IN channels. It just inserts them by overwriting the existing data payload.

Workaround

Use one of the following recommendations:

1. Configure the RxFIFO to host a *minimum* of 2 × MPSIZ + 2 × data status entries.
2. The application has to check the RXFLVL bit (RxFIFO non-empty) in the OTG_FS_GINTSTS register before disabling each IN channel. If this bit is not set, then the application can disable an IN channel at a time. Each time the application disables an IN channel, however, it first has to check that the RXFLVL bit = 0 condition is true.

2.11.2 OTG host blocks the receive channel when receiving IN packets and no TxFIFO is configured

Description

When receiving data, the OTG_FS core erroneously checks for available TxFIFO space when it should only check for RxFIFO space. If the OTG_FS core cannot see any space allocated for data transmission, it blocks the reception channel and no data are received.

Workaround

Set at least one TxFIFO equal to the maximum packet size. In this way, the host application, which intends to supports only IN traffic, also has to allocate some space for the TxFIFO.

Since a USB host is expected to support any kind of connected endpoint, it is good practice to always configure enough TxFIFO space for OUT endpoints.

2.11.3 Host channel-halted interrupt not generated when the channel is disabled

Description

When the application enables, then immediately disables the host channel before the OTG_FS host has had time to begin the transfer sequence, the OTG_FS core, as a host, does not generate a channel-halted interrupt. The OTG_FS core continues to operate normally.

Workaround

Do not disable the host channel immediately after enabling it.

2.11.4 Error in software-read OTG_FS_DCFG register values

Description

When the application writes to the DAD and PFIVL bitfields in the OTG_FS_DCFG register, and then reads the newly written bitfield values, the read values may not be correct.

The values written by the application, however, are correctly retained by the core, and the normal operation of the device is not affected.

Workaround

Do not read from the OTG_FS_DCFG register's DAD and PFIVL bitfields just after programming them.

2.12 Ethernet MAC

2.12.1 Possible underflow when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode

Description

In Store-and-Forward mode, the frame is transferred to the MAC only when the full frame is available in the TxFIFO. There is one exception to this rule, when the TxFIFO is almost full.

This problem may arise when a relatively large frame is loaded into the TxFIFO and a second frame is partially loaded (because of lack of space) into the free TxFIFO space, generating an almost full condition. In this case, if the first frame is aborted, due to a carrier loss, the absence of a carrier or a late collision, the second frame is sent to the MAC before it is completely loaded into the TxFIFO. If the DMA bandwidth does not allow the rest of the frame to be uploaded before the end of the frame transmission, an underflow may occur.

In this case, the software must detect the underflow condition and resend the frame.

Workaround

Resend the frame on underflow detection.

2.12.2 Possible CRC error when TxFIFO is configured in Store-and-Forward mode and a relatively large frame is aborted in Half-duplex mode with transmit checksum offload enabled

Description

In Store-and-Forward mode, the frame is transferred to the MAC only when the full frame is available in the TxFIFO. There is one exception to this rule, when the TxFIFO is almost full.

This problem may arise when a relatively large frame is loaded into the TxFIFO and a second frame is partially loaded (because of lack of space) into the free TxFIFO space, generating an almost full condition. In this case, if the first frame is aborted, due to a carrier loss, the absence of a carrier or a late collision, the second frame is sent to the MAC before it is completely loaded into the TxFIFO. Since for CRC computation, the entire frame must be available in the TxFIFO, the CRC of the subsequent frames will be erroneous.

TCP/IP checksum errors may be detected at the remote end, causing data to be dropped.

Workaround

Wait for the TxFIFO to become empty and then send the txFIFO flush command.

2.12.3 Erroneous automatic checksum insertion after TxFIFO is dynamically switched from Threshold to Store-and-Forward mode

Description

Automatic checksum insertion in transmitted frames can be enabled (through the CIC bits in the TDES0 register) and used only when the TxFIFO is configured to operate in Store-and-Forward mode. When the TxFIFO is operated in Threshold mode, the CIC bits are ignored.

When the TxFIFO dynamically switches from Threshold to Store-and-Forward mode, the CRC computation may be erroneous on the subsequent frames.

This results in the detection of TCP/IP checksum errors at the remote end, causing data to be dropped.

Workaround

Use the Store-and-Forward mode only.

2.12.4 Erroneous automatic checksum insertion after a large frame (longer than the TxFIFO) transmission

Description

Automatic checksum insertion in transmitted frames can be enabled (through the CIC bits in TDES0) and used only for frames with a length lesser than the TxFIFO depth.

If a long frame is transmitted, with checksum insertion disabled (CIC=00), immediately followed by a short frame, with checksum insertion enabled, the computed checksum may be erroneous. This is due to the fact that the second frame may have be transmitted too early from the TxFIFO, due to a TxFIFO almost full condition.

This results in the detection of TCP/IP checksum errors at the remote end, causing data to be dropped.

Workaround

1. Avoid enabling automatic checksum insertion for any frame if your system transmits frames with a size larger than the depth of the TxFIFO.
2. Wait for the long frame transmission completion before re-enabling the automatic checksum insertion.

2.12.5 In half-duplex mode, the MAC transmitter ignores collisions after it is disabled during a frame transmission

Description

If the TE bit is cleared during a transmission, the transmission part of the MAC is effectively disabled after the complete transmission of the current frame. From the clearing of the TE bit until the end of the transmission, potential collisions are ignored.

If a collision event occurs after the MAC is disabled, the transmitter does not recognize the event and continues to transmit the complete frame without any JAM pattern. It reports a successful frame transmission status (without any collision) even though the frame is corrupted at the remote receivers due to collision.

Since a JAM signal might not be sent during a collision, frames may be lost when a remote transmitter does not detect that a collision occurred.

Workaround

Disable the MAC transmitter only after the completion of the transmission of all scheduled frames in Half-duplex mode.

2.12.6 Interrupt due to an RMON (MMC) counter may be set again after it is cleared

Description

When enabled, an interrupt asserted due to an RMON (remote monitoring) counter becoming full, is cleared when the corresponding counter is read. The counter is also cleared by the read operation if bit 1 (counter stop rollover) in ETH_MMCCR is set. If this clear signal coincides with the counter update signal generated by the presence of a new frame, then the corresponding interrupt bit is set again. This results in the software getting a spurious interrupt from the MMC even though the corresponding counter value is very low.

Workaround

None.

2.12.7 Incorrect layer 3 (L3) checksum is inserted in transmitted IPv6 packets without TCP, UDP or ICMP payloads

Description

The application provides the per-frame control to instruct the MAC to insert the L3 checksums for TCP, UDP and ICMP packets. When automatic checksum insertion is enabled and the input packet is an IPv6 packet without the TCP, UDP or ICMP payload, then the MAC may incorrectly insert a checksum into the packet. For IPv6 packets without a TCP, UDP or ICMP payload, the MAC core considers the next header (NH) field as the extension header and continues to parse the extension header. Sometimes, the payload data in such packets matches the NH field for TCP, UDP or ICMP and, as a result, the MAC core inserts a checksum.

Workaround

When the IPv6 packets have a TCP, UDP or ICMP payload, enable checksum insertion for transmit frames, or bypass checksum insertion by using the CIC (checksum insertion control) bits in TDES0 (bits 23:22).

2.12.8 The Ethernet MAC processes invalid extension headers in the received IPv6 frames

Description

In IPv6 frames, there can be zero or some extension headers preceding the actual IP payload. The Ethernet MAC processes the following extension headers defined in the IPv6 protocol: Hop-by-Hop Options header, Routing header and Destination Options header. All extension headers except the Hop-by-Hop extension header can be present multiple times and in any order before the actual IP payload. The Hop-by-Hop extension header, if present, has to come immediately after the IPv6's main header.

The Ethernet MAC processes all (valid or invalid) extension headers including the Hop-by-Hop extension headers that are present after the first extension header. For this reason, the GMAC core will accept IPv6 frames with invalid Hop-by-Hop extension headers. As a consequence, it will accept any IP payload as valid IPv6 frames with TCP, UDP or ICMP payload, and then incorrectly update the Receive status of the corresponding frame.

Workaround

None.

2.12.9 MAC stuck in the Idle state on receiving the TxFIFO flush command exactly 1 clock cycle after a transmission completes

Description

When the software issues a TxFIFO flush command, the transfer of frame data stops (even in the middle of a frame transfer). The TxFIFO read controller goes into the Idle state (TFRS=00 in ETH_MACDBGR) and then resumes its normal operation.

However, if the TxFIFO read controller receives the TxFIFO flush command exactly one clock cycle after receiving the status from the MAC, the controller remains stuck in the Idle state and stops transmitting frames from the TxFIFO. The system can recover from this state only with a reset (e.g. a soft reset).

Workaround

Do not use the TxFIFO flush feature.

If TXFIFO flush is really needed, wait until the TxFIFO is empty prior to using the TxFIFO flush command.

2.12.10 Transmit frame data corruption

Frame data corrupted when the TxFIFO is repeatedly transitioning from non-empty to empty and then back to non-empty.

Description

Frame data may get corrupted when the TxFIFO is repeatedly transitioning from non-empty to empty for a very short period, and then from empty to non-empty, without causing an underflow.

This transitioning from non-empty to empty and back to non-empty happens when the rate at which the data are being written to the TxFIFO is almost equal to or a little less than the rate at which the data are being read.

This corruption cannot be detected by the receiver when the CRC is inserted by the MAC, as the corrupted data are used for the CRC computation.

Workaround

Use the Store-and-Forward mode: TSF=1 (bit 21 in ETH_DMAOMR). In this mode the data are transmitted only when the whole packet is available in the TxFIFO.

2.13 Boot loader unavailability on STM32F105xx and STM32F107xx devices with a date code below 937

Description

During the boot loader activation phase, if the USART1_RX (PA10), USART2_RX (PD6, remapped pin), CAN2_Rx (PB5, remapped pin), OTG_FS_DM (PA11) and/or OTG_FS_DP (PA12) pin(s) are connected to low level or left floating, the boot loader cannot be used. It is not possible to connect to the boot loader through either of CAN2 (remapped), DFU (OTG FS in Device mode), USART1 or USART2 (remapped).

In 64-pin packages, the USART2_RX remapped pin PD6 is not available and is internally grounded. Therefore, the bootloader cannot be used at all.

Workaround

- For 64-pin packages: none. The boot loader cannot be used.
- For 100-pin packages: depending on the used peripheral, the pins for the unused peripherals have to be kept at a high level during the boot loader activation phase as described below:
  - If USART1 is used to connect to the boot loader: PD6 and PB5 have to be kept at a high level
  - If USART2 is used to connect to the boot loader: PA10, PB5, PA11 and PA12 have to be kept at a high level
  - If CAN2 is used to connect to the boot loader: PA10, PD6, PA11 and PA12 have to be kept at a high level
  - If DFU is used to connect to the boot loader: PA10, PB5 and PD6 have to be kept at a high level

Note: This limitation concerns o*nly STM32F105xx and STM32F107xx devices with a date code below 937.*

STM32F105xx and STM32F107xx devices with a date code of 937 and above are ***not*** *impacted.*

See Appendix A: Revision and date codes on device marking for where to find the date code on the device marking.

Appendix A Revision and date codes on device marking

Figure 1 and *Figure 2* show the marking compositions for the LQFP100 and LQFP64 packages, respectively. The only fields shown are the Additional field containing the revision code and the Year and Week fields making up the date code.

Figure 1. LQFP100 top package view

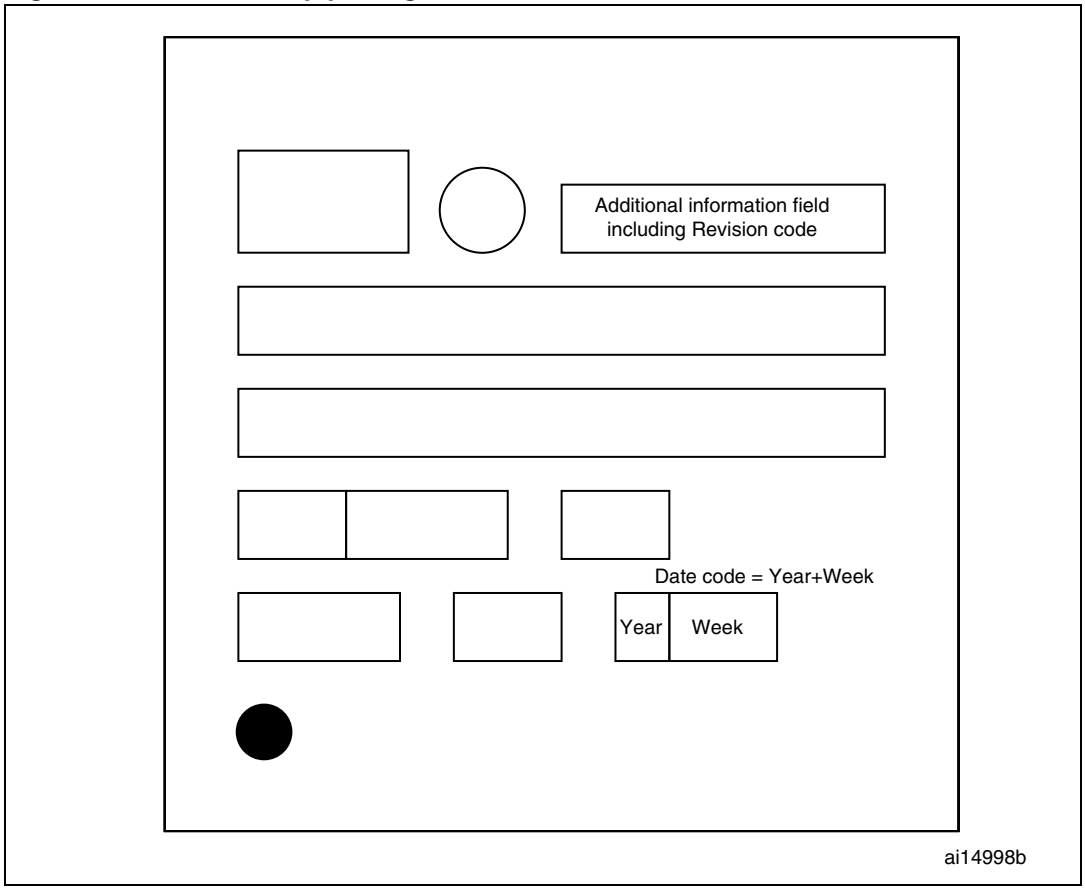
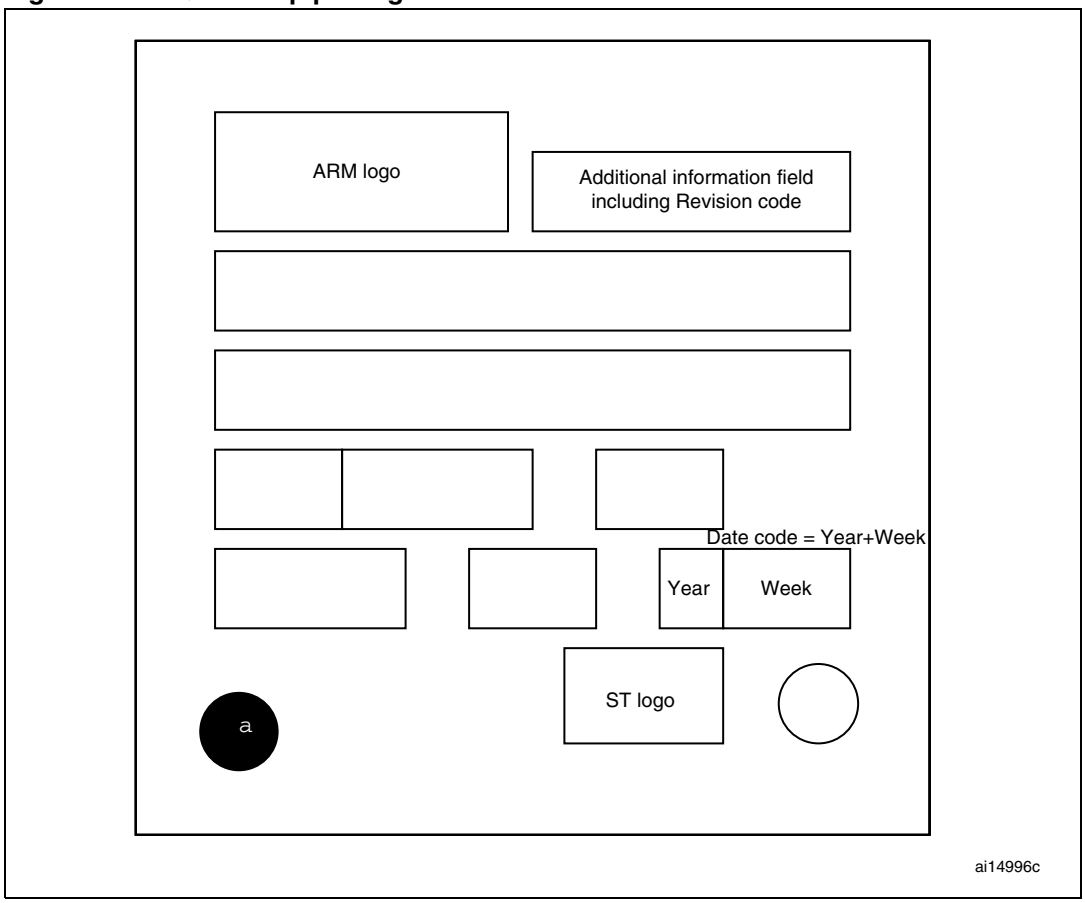


Figure 2. LQFP64 top package view



Revision history

Table 5. Document revision history

Date	Revision	Changes
16-Jun-2009	1	Initial release.
16-Oct-2009	2	*Section 2.3.5: I2S2 in master/slave mode and Ethernet/USART3 in synchronous mode* added. *Table 4: Summary of silicon limitations in revision Z devices* modified. *Section 2.11: OTG_FS* added. *Section 2.12: Ethernet MAC* added. *Section 2.13: Boot loader unavailability on STM32F105xx and STM32F107xx devices with a date code below 937* added. *Figure 1: LQFP100 top package view* and *Figure 2: LQFP64 top package view* updated to show the Date code.
15-Dec-2009	3	Added limitations: – *Section 2.3.6: USARTx_TX pin usage* – *Section 2.6.3: Start cannot be generated after a misplaced Stop* – *Section 2.6.4: Mismatch on the "Setup time for a repeated Start condition" timing parameter* – *Section 2.6.5: Data valid time (tVD;DAT) violated without the OVR flag being set* – *Section 2.7.1: CRC still sensitive to communication clock when SPI is in slave mode even with NSS high*

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com