Mask Set Errata for Mask 8N30D

Introduction

This report applies to mask 8N30D for these products:

- KINETIS

Errata ID	Errata Title
2550	ADC: ADC abort conversion logic error
2776	CRC: May have incorrect CRC result when performing CRC 8-bit or 16-bit writes with transpose enabled.
2547	DAC: 12-bit DAC buffer registers cannot be read.
4588	DMAMUX: When using PIT with "always enabled" request, DMA request does not deassert correctly
2579	ENET: No support for IEEE 1588, TS_TIMER, timestamp timer overflow interrupt
6358	ENET: Write to Transmit Descriptor Active Register (ENET_TDAR) is ignored
2687	FMC: Flash clock divider setting for divide-by-1 not allowed
2590	FMC: Master Access Protection encoding for write only access does not work as specified.
3372	FTFL: Reset during an EEE program operation may result in an invalid EEE read access
2781	FlexBus: False bus error on back-to-back writes when flash memory is secure
2616	FlexCAN: Module receives data frames sent by itself although the self reception feature is disabled
3402	GPIO: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set.
2793	I2C: MCU does not wake as expected from STOP or VLPS mode on subsequent address matches if previous address is mismatched
3795	I2S: Fractional divider in SIM_CLKDIV2 is not reset when recovering from VLLSx low power modes
3714	I2S: MCLK output is disabled in asynchronous mode
2674	LLWU: The LLWU glitch filter for pin and reset is not supported
2678	MC: The MC_SRS[PIN] is not always set after exiting a VLLS mode due to a RESET pin assertion
2676	MC: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.
3794	NVIC: NMI interrupt does not wakeup MCU from STOP and VLPS
6328	PMC: Incorrect exit from VLLSx modes
2542	PMC: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported

Table continues on the next page...

Errata ID	Errata Title
2576	RTC: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode
3981	SDHC: ADMA fails when data length in the last descriptor is less or equal to 4 bytes
3982	SDHC: ADMA transfer error when the block size is not a multiple of four
4624	SDHC: AutoCMD12 and R1b polling problem
3977	SDHC: Does not support Infinite Block Transfer Mode
4627	SDHC: Erroneous CMD CRC error and CMD Index error may occur on sending new CMD during data transfer
3980	SDHC: Glitch is generated on card clock with software reset or clock divider change
3983	SDHC: Problem when ADMA2 last descriptor is LINK or NOP
3978	SDHC: Software can not clear DMA interrupt status bit after read operation
3984	SDHC: eSDHC misses SDIO interrupt when CINT is disabled
5708	SLCD: LCD waveforms can exceed voltage specification for 3V or 5V glass
2591	TSI: TSI_SCANC[SMOD] behaves as an inactive time instead of a scan period value
2638	TSI: The counter registers are not immediately updated after the EOSF bit is set
2582	UART: Flow control timing issue can result in loss of characters
4945	UART: ISO-7816 T=1 mode receive data format with a single stop bit is not supported
3892	UART: ISO-7816 automatic initial character detect feature not working correctly
2584	UART: Possible conflicts between UART interrupt service routines and DMA requests
5704	UART: TC bit in UARTx_S1 register is set before the last character is sent out in ISO7816 T=0 mode
5928	USBOTG: USBx_USBTRC0[USBRESET] bit does not operate as expected in all cases
3796	USBREG: Cannot use USB Regulator standby mode when output is powering MCU via VDD
2686	WDOG: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

e2550: ADC: ADC abort conversion logic error

Errata type: Errata

Description: The ADC abort conversion logic does not function as specified. Writes to the ADC CV1, CV2, OFS, PG, MG, CLPx, and CLMx registers will not abort a conversion.

Workaround: The abort conversion logic protects against changes to the ADC configuration during a conversion. To avoid this issue, do not change ADC settings during a conversion.

e2776: CRC: May have incorrect CRC result when performing CRC 8-bit or 16-bit writes with transpose enabled.

Errata type: Errata

Description: If performing CRC 8-bit or 16-bit writes with transpose enabled, the final checksum may have an incorrect CRC result.

Workaround: Write accesses to the CRC when transpose is enabled should always be 32-bit.

e2547: DAC: 12-bit DAC buffer registers cannot be read.

Errata type: Errata

Description: The 12-bit DAC buffer registers, DACx_DAT[1:15]L and DACx_DAT[1:15]H, cannot be read. The data that is written to these registers cannot be read. Only DACx_DAT0L and DACx_DAT0H can be read correctly.

Workaround: Treat the DACx_DAT[1:15]L and DACx_DAT[1:15]H registers as write-only registers because reads may return invalid data. The DAC buffer can still be used since the values written to these registers are valid.

e4588: DMAMUX: When using PIT with "always enabled" request, DMA request does not deassert correctly

Errata type: Errata

Description: The PIT module is not assigned as a stand-alone DMA request source in the DMA request mux. Instead, the PIT is used as the trigger for the DMAMUX periodic trigger mode. If you want to use one of the PIT channels for periodic DMA requests, you would use the periodic trigger mode in conjunction with one of the "always enabled" DMA requests. However, the DMA request does not assert correctly in this case.

Instead of sending a single DMA request every time the PIT expires, the first time the PIT triggers a DMA transfer the "always enabled" source will not negate its request. This results in the DMA request remaining asserted continuously after the first trigger.

Workaround: Use of the PIT to trigger DMA channels where the major loop count is greater than one is not recommended. For periodic triggering of DMA requests with major loop counts greater than one, we recommended using another timer module instead of the PIT.

If using the PIT to trigger a DMA channel where the major loop count is set to one, then in order to get the desired periodic triggering, the DMA must do the following in the interrupt service routine for the DMA_DONE interrupt:

1. Set the DMA_TCDn_CSR[DREQ] bit and configure DMAMUX_CHCFGn[ENBL] = 0

2. Then again DMAMUX_CHCFGn[ENBL] = 1, DMASREQ=channel in your DMA DONE interrupt service routine so that "always enabled" source could negate its request then DMA request could be negated.

This will allow the desired periodic triggering to function as expected.

e2579: ENET: No support for IEEE 1588, TS_TIMER, timestamp timer overflow interrupt

Errata type: Errata

Description: The TS_TIMER interrupt signal is not connected to the NVIC and will not generate an interrupt event. This interrupt is set when the 1588 counter matches the period register.

Workaround: One of the 1588 counter channels can be configured in output compare software-only mode to generate the periodic interrupt events.

This can be used to generate a counter periodic interrupt:

Initialize the timer:

1) Set the ENET_ATPER to the desired value

2) Set the ENET_ATINC register to match the selected 1588 clock.

3) Set the ENET_TCCRn register with ENET_ATPER – ENET_ATINC[INC] value. The ENET_ATINC[INC] offset is needed to match the internal 1588 clock synchronization.

4) Set the ENET_TCSRn[TMODE] register with the 0100 encoding for output compare software-only mode and the ENET_TCSRn[TIE] to enable the timer interrupt.

5) Set the ENET_TCCRn register again with ENET_ATPER – ENET_ATINC[INC] value because output compare value is double buffered.

6) Set the ENET_ATCR[PEREN] to enable periodical event and set the ENET_ATCR[EN] to start the timer

Configure inside the ISR:

1) 1588 interrupts are generated via the NVIC vector 91 using the periodic timer. For each interrupt event, load the output compare buffer (ENET_TCCRn register) with ENET_ATPER – ENET_ATINC[INC] value.

2) Clear ENET_TCSRn[TF] flag

3) Clear ENET_TGSRn respective channel flag.

e6358: ENET: Write to Transmit Descriptor Active Register (ENET_TDAR) is ignored

Errata type: Errata

Description: If the ready bit in the transmit buffer descriptor (TxBD[R]) is previously detected as not set during a prior frame transmission, then at a later time the ENET_TDAR[TDAR] bit is cleared even if additional TxBDs were added to the ring and the ENET_TDAR[TDAR] bit set. This results in frames not being transmitted until there is a 0 to 1 transition on ENET_TDAR[TDAR].

Workaround: Code can use the transmit frame interrupt flag (ENET_EIR[TXF]) as a method to detect that the ENET has completed transmission and that the ENET_TDAR[TDAR] might have been cleared. If ENET_TDAR[TDAR] is detected as cleared when packets are queued waiting for transmit, then a write to the TDAR bit will restart TxBD processing.

e2687: FMC: Flash clock divider setting for divide-by-1 not allowed

Errata type: Errata

Description: If the Flash clock divider is set for divide-by-1 and a system reset occurs the Flash may be clocked at too high a frequency (>25MHz) and the system may access bad data from the Flash array during reset recovery sequencing.

Workaround: The Flash Controller should enable buffering to offset improve performance when Flash clock is configured for divide-by-2 or greater.

e2590: FMC: Master Access Protection encoding for write only access does not work as specified.

Errata type: Errata

Description: The Access Protection Register (FMC_PFAPR) in the Flash Memory Controller provides write and/or read access control to the FlexRAM space on a per master granularity.

The eight two bit Master Access Protection fields in the FMC_PFAPR are encoded as follows:

00 - No access may be performed by this master

01 - Read only accesses may be performed by this master

10 - Write only accesses may be performed by this master

11 - Read and Write accesses may be performed by this master

The "10" encoding (write only accesses) does not work as specified. This encoding blocks all accesses.

So, the "10" encoding functions the same as the "00" encoding. Due to this error, it is not possible to designate the FMC memory space as "write only" for any master.

Note that the only writable portion of the FMC memory space is the FlexRAM space.

Workaround: The "11" encoding, allowing read and write accesses, must be programmed to allow writes for a given master.

e3372: FTFL: Reset during an EEE program operation may result in an invalid EEE read access

Errata type: Errata
Description: In rare occurrences, a reset during an EEE program operation may result in invalid EEE read access.

Workaround: Avoid a reset during EEE programming.

e2781: FlexBus: False bus error on back-to-back writes when flash memory is secure

Errata type: Errata
Description: During back-to-back writes, the FlexBus incorrectly responds with a bus error on the second write when both of these conditions apply: the flash memory is secure (per the value of the FTFL module's FSEC[SEC] field), and the SIM's SOPT2[FBSL] field is 10b. This setting of the SOPT2[FBSL] field disallows instruction accesses but allows data accesses on the FlexBus interface when the flash memory is secure.

Workaround: When the flash memory is secure and FlexBus instruction accesses are inhibited but data accesses are allowed, do not use back-to-back writes. Insert a delay or NOP instruction between the write operations.

e2616: FlexCAN: Module receives data frames sent by itself although the self reception feature is disabled

Errata type: Errata
Description: The FlexCAN receives frames transmitted by itself although the self reception feature is disabled (MCR[SRX_DIS] is asserted). As a result, the transmitted data is moved into Mailbox and the IFLAG is asserted.

The error occurs when there is at least one reception Mailbox whose ID matches a frame that is being transmitted and the FlexCAN requests Freeze mode during the frame transmission.

The occurrence of this error depends on the software strategy used to request Freeze Mode and how often Freeze Mode is requested during module operation.

Workaround: If the self reception feature is disabled (MCR[SRX_DIS] = 1), in order to avoid receiving a self transmitted frame, Freeze Mode should only be requested when all Mailboxes that are configured as TX have been transmitted or aborted.

e3402: GPIO: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set.

Errata type: Errata

Description: XTAL pin cannot be used as GPIO if the ERCLKEN bit is set. Errata applies only when an external clock is being used (the crystal oscillator is not being used) and OSC_CR[ERCLKEN] bit set to 1. In this specific case, the analog block of the crystal oscillator is enabled and the oscillator output is driving the XTAL pin even if the respective pin control register has been configured as a GPIO. This prevents the pin from being used as either an input or an ouput.

Workaround: If an external clock is not being used, then the OSCERCLK is not available and there is no need to set the ERCLKEN bit. If the ERCLKEN bit is not set then the XTAL pin can be used as a GPIO.

If an external clock is being used but the OSCERCLK is not required, then the ERCLKEN bit should not be set and the XTAL pin can be used as a GPIO.

If an external clock is being used and the OSCERCLK is required and the ERCLKEN bit is set, then there is no workaround and the XTAL pin cannot be used as a GPIO.

e2793: I2C: MCU does not wake as expected from STOP or VLPS mode on subsequent address matches if previous address is mismatched

Errata type: Errata

Description: The I2C module, acting as a slave on the I2C bus, does not wake as expected from normal STOP mode or VLPS mode on a valid address match if the previous address was not a match.

When the external I2C master sends a non-matching address, the I2C slave state machine does not look for a start bit past the first start bit on the bus. Consequently, subsequent transmissions by the I2C master with a matching address do not, on the first matching address, wake the MCU from stop mode or VLPS via the I2C interrupt.

Workaround: There are multiple workarounds:

(1) The master must continually re-transmit the MCU's slave address upon not receiving a NACK from the slave device during the slave addressing phase of the transmission. For clarification, the master must perform the following:

a) Send slave device address

b) Check for ACK bit

c) If ACK was received, continue with data transmission. Else, send repeated start signal and repeat steps a-c.

NOTE: Due to the nature of the errata, the maximum number of retransmissions needed to wake the part is nine times.

(2) When the MCU, operating as an I2C slave, is in STOP or VLPS mode: Ensure that the external I2C master sends a matching address to wake the slave MCU before it sends any transaction to other I2C slaves. The user must also ensure that MCU does not return to STOP or VLPS until after all packets to non-matching addresses have been sent.

(3) Use a pin interrupt (any pin, whether that pin is or is not being used by the active I2C module) to wake up the part before receiving I2C packets. NOTE: If using the SDA or SCL pin that the active I2C module is using, the part will wake-up on every I2C transaction on the bus.

(4) Use Wait mode instead of STOP or VLPS mode.

e3795: I2S: Fractional divider in SIM_CLKDIV2 is not reset when recovering from VLLSx low power modes

Errata type: Errata

Description: The Fractional divider in the SIM_CLKDIV2 register is not reset when recovering from VLLSx low power modes.

Workaround: In order to reset the Fractional divider when recovering from VLLSx low power modes, the I2S clock gate must be enabled, then disabled, and re-enabled after exiting any VLLSx mode.

e3714: I2S: MCLK output is disabled in asynchronous mode

Errata type: Errata

Description: When using I2S MCLK output in asynchronous mode, the output is disabled.

Workaround: In order to output MCLK, use synchronous mode (I2Sx_CR[SYN]) along with setting the transmit direction to use an internally generated clock output through the serial transmit clock port (I2Sx_TCR[TXDIR]).

e2674: LLWU: The LLWU glitch filter for pin and reset is not supported

Errata type: Errata

Description: LLWU glitch filter for pin and reset is not supported. Do not enable the filters. Writing a 1 to FLTEP or FLTR bits in the LLWU_CS register will result in abnormal LLWU behavior.

Workaround: It is recommended that the glitch filter functionality is not used. Always write a zero to the FLTEP or FLTR bits in the LLWU_CS register. If filtering is required, external components or software filters are required.

e2678: MC: The MC_SRS[PIN] is not always set after exiting a VLLS mode due to a RESET pin assertion

Errata type: Errata

Description: When waking the device from VLLS low power modes via a RESET pin, the MC_SRSL[PIN] bit is not reliably set on recovery and cannot be used to differentiate VLLS wakeup from the RESET pin from other low power mode wakeup sources enabled in the LLWU module. This condition is encountered when the device is in a VLLS low power mode and the RESET pin is held asserted for greater than 100us to initiate a VLLS wakeup.

Workaround: When using the RESET pin as a wakeup trigger for exit from VLLS, ensure that the wakeup pulse is short in duration (<100us).

An alternate workaround is to only use the MC_SRS[WAKEUP] bit to indicate VLLS exit recoveries and reset initialization routines to not distinguish between RESET pin triggered recoveries and other sources of VLLS wakeup.

e2676: MC: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.

Errata type: Errata

Description: When waking the system from VLLS modes via a RESET pin, the I/O are not immediately released to their reset state.

Workaround: The reset initialization/recovery software must set the LLWU_CS[ACKISO] bit just as it would due to a low power wakeup via any of the other low power mode wakeup sources.

e3794: NVIC: NMI interrupt does not wakeup MCU from STOP and VLPS

Errata type: Errata

Description: NMI interrupt does not wakeup MCU from STOP and VLPS when the bits CSYSPWRUPREQ and CDBGPWRUPREQ in the Control/Status Register of the DAP Port are cleared.

Workaround: If a debugger connection is established, the CSYSPWRUPREQ and CDBGPWRUPREQ bits are set by default, so an NMI interrupt will wake up the MCU from STOP and VLPS modes. In the absence of a debug connection and after a POR event, the bits will be cleared and thus an NMI interrupt will not wake the MCU.

e6328: PMC: Incorrect exit from VLLSx modes

Errata type: Errata

Description: When VDD < 1.9V, device may not return to run mode when exiting VLLSx modes. The device will remain in a non-operational state until VDD is lowered to force either an low-voltage detect (LVD) reset or a power-on-reset (POR) event.

Workaround: Either use VLLSx with VDD > 1.9V or use another low power mode such as LLS.

e2542: PMC: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported

Errata type: Errata

Description: Very Low Power Run (VLPR) and Very Low Power Wait (VLPW) power modes are not supported.

Workaround: Do not use the part in these two power modes.

e2576: RTC: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode

Errata type: Errata

Description: When the RTC is configured to allow supervisor access only, the write and read access registers can be modified in user mode. A bus error is still generated.

Workaround: RTC supervisor mode access only option is not supported.

e3981: SDHC: ADMA fails when data length in the last descriptor is less or equal to 4 bytes

Errata type: Errata

Description: A possible data corruption or incorrect bus transactions on the internal AHB bus, causing possible system corruption or a stall, can occur under the combination of the following conditions:

1. ADMA2 or ADMA1 type descriptor

2. TRANS descriptor with END flag

3. Data length is less than or equal to 4 bytes (the length field of the corresponding descriptor is set to 1, 2, 3, or 4) and the ADMA transfers one 32-bit word on the bus

4. Block Count Enable mode

Workaround: The software should avoid setting ADMA type last descriptor (TRANS descriptor with END flag) to data length less than or equal to 4 bytes. In ADMA1 mode, if needed, a last NOP descriptor can be appended to the descriptors list. In ADMA2 mode this workaround is not feasible due to ERR003983.

e3982: SDHC: ADMA transfer error when the block size is not a multiple of four

Errata type: Errata

Description: Issue in eSDHC ADMA mode operation. The eSDHC read transfer is not completed when block size is not a multiple of 4 in transfer mode ADMA1 or ADMA2. The eSDHC DMA controller is stuck waiting for the IRQSTAT[TC] bit in the interrupt status register.

The following examples trigger this issue:

1. Working with an SD card while setting ADMA1 mode in the eSDHC

2. Performing partial block read

3. Writing one block of length 0x200

4. Reading two blocks of length 0x22 each. Reading from the address where the write operation is performed. Start address is 0x512 aligned. Watermark is set as one word during read. This read is performed using only one ADMA1 descriptor in which the total size of the transfer is programmed as 0x44 (2 blocks of 0x22).

Workaround: When the ADMA1 or ADMA2 mode is used and the block size is not a multiple of 4, the block size should be rounded to the next multiple of 4 bytes via software. In case of write, the software should add the corresponding number of bytes at each block end, before the write is initialized. In case of read, the software should remove the dummy bytes after the read is completed.

For example, if the original block length is 22 bytes, and there are several blocks to transfer, the software should set the block size to 24. The following data is written/stored in the external memory:

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

2 Bytes valid data + 2 Byte dummy data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

4 Bytes valid data

2 Bytes valid data + 2 Byte dummy data

In this example, 48 (24 x 2) bytes are transferred instead of 44 bytes. The software should remove the dummy data.

e4624: SDHC: AutoCMD12 and R1b polling problem

Errata type: Errata

Description: Occurs when a pending command which issues busy is completed. For a command with R1b response, the proper software sequence is to poll the DLA for R1b commands to determine busy state completion. The DLA polling is not working properly for the ESDHC module and thus the DLA bit in PRSSTAT register cannot be polled to wait for busy state ompletion. This is relevant for all eSDHC ports (eSDHC1-4 ports).

Workaround: Poll bit 24 in PRSSTAT register (DLSL[0] bit) to check that wait busy state is over.

e3977: SDHC: Does not support Infinite Block Transfer Mode

Errata type: Errata

Description: The eSDHC does not support infinite data transfers, if the Block Count register is set to one, even when block count enable is not set.

Workaround: The following software workaround can be used instead of the infinite block mode:

1. Set BCEN bit to one and enable block count

2. Set the BLKCNT to the maximum value in Block Attributes Register (BLKATTR) (0xFFFFfor 65535 blocks)

e4627: SDHC: Erroneous CMD CRC error and CMD Index error may occur on sending new CMD during data transfer

Errata type: Errata

Description: When sending new, non data CMD during data transfer between the eSDHC and EMMC card, the module may return an erroneous CMD CRC error and CMD Index error. This occurs when the CMD response has arrived at the moment the FIFO clock is stopped. The following bits after the start bit of the response are wrongly interpreted as index, generating the CRC and Index errors.

The data transfer itself is not impacted.

The rate of occurrence of the issue is very small, as there is a need for the following combination of conditions to occur at the same cycle:

• The FIFO clock is stopped due to FIFO full or FIFO empty

• The CMD response start bit is received

Workaround: The recommendation is to not set FIFO watermark level to a too small value in order to reduce frequency of clock pauses.

The problem is identified by receiving the CMD CRC error and CMD Index error. Once this issue occurs, one can send the same CMD again until operation is successful.

e3980: SDHC: Glitch is generated on card clock with software reset or clock divider change

Errata type: Errata

Description: A glitch may occur on the SDHC card clock when the software sets the RSTA bit (software reset) in the system control register. It can also be generated by setting the clock divider value. The glitch produced can cause the external card to switch to an unknown state. The occurrence is not deterministic.

Workaround: A simple workaround is to disable the SD card clock before the software reset, and enable it when the module resumes the normal operation. The Host and the SD card are in a master-slave relationship. The Host provides clock and control transfer across the interface. Therefore, any existing operation is discarded when the Host controller is reset.

The recommended flow is as follows:

1. Software disable bit[3], SDCLKEN, of the System Control Register

2. Trigger software reset and/or set clock divider

3. Check bit[3], SDSTB, of the Present State Register for stable clock

4. Enable bit[3], SDCLKEN, of the System Control Register.

Using the above method, the eSDHC cannot send command or transfer data when there is a glitch in the clock line, and the glitch does not cause any issue.

e3983: SDHC: Problem when ADMA2 last descriptor is LINK or NOP

Errata type: Errata

Description: ADMA2 mode in the eSDHC is used for transfers to/from the SD card. There are three types of ADMA2 descriptors: TRANS, LINK or NOP. The eSDHC has a problem when the last descriptor (which has the End bit '1') is a LINK descriptor or a NOP descriptor.

In this case, the eSDHC completes the transfers associated with this descriptor set, whereas it does not even start the transfers associated with the new data command. For example, if a WRITE transfer operation is performed on the card using ADMA2, and the last descriptor of the WRITE descriptor set is a LINK descriptor, then the WRITE is successfully finished. Now, if a READ transfer is programmed from the SD card using ADMA2, then this transfer does not go through.

Workaround: Software workaround is to always program TRANS descriptor as the last descriptor.

e3978: SDHC: Software can not clear DMA interrupt status bit after read operation

Errata type: Errata

Description: After DMA read operation, if the SDHC System Clock is automatically gated off, the DINT status can not be cleared by software.

Workaround: Set HCKEN bit before starting DMA read operation, to disable SDHC System Clock auto-gating feature; after the DINT and TC bit received when read operation is done, clear HCKEN bit to re-enable the SDHC System Clock auto-gating feature.

e3984: SDHC: eSDHC misses SDIO interrupt when CINT is disabled

Errata type: Errata

Description: An issue is identified when interfacing the SDIO card. There is a case where an SDIO interrupt from the card is not recognized by the hardware, resulting in a hang.

If the SDIO card lowers the DAT1 line (which indicates an interrupt) when the SDIO interrupt is disabled in the eSDHC registers (that is, CINTEN bits in IRQSTATEN and IRQSIGEN are set to zero), then, after the SDIO interrupt is enabled (by setting the CINTEN bits in IRQSTATEN and IRQSIGEN registers), the eSDHC does not sense that the DAT1 line is low. Therefore, it fails to set the CINT interrupt in IRQSTAT even if DAT1 is low.

Generally, CINTEN bit is disabled in interrupt service.

The SDIO interrupt service steps are as follows:

1. Clear CINTEN bit in IRQSTATEN and IRQSIGEN.

2. Reset the interrupt factors in the SDIO card and write 1 to clear the CINT interrupt in IRQSTAT.

3. Re-enable CINTEN bit in IRQSTATEN and IRQSIGEN.

If a new SDIO interrupt from the card occurs between step 2 and step 3, the eSDHC skips it.

Workaround: The workaround interrupt service steps are as follows:

1. Clear CINTEN bit in IRQSTATEN and IRQSIGEN.

2. Reset the interrupt factors in the SDIO card and write 1 to clear CINT interrupt in IRQSTAT.

3. Clear and then set D3CD bit in the PROCTL register. Clearing D3CD bit sets the reverse signal of DAT1 to low, even if DAT1 is low. After D3CD bit is re-enabled, the eSDHC can catch the posedge of the reversed DAT1 signal, if the DAT1 line is still low.

4. Re-enable CINTEN bit in IRQSTATEN and IRQSIGEN.

e5708: SLCD: LCD waveforms can exceed voltage specification for 3V or 5V glass

Errata type: Errata

Description: The LCD controller power supply can be sourced using VDD, internal regulated voltage VIREG, or an external supply on the VLL3. Using the internal Vireg (VSUPPLY[1:0] = 11, LCD_GCR [RVEN] = 1, and LCD_GCR[CPSEL] =1) to generate the LCD bias voltages can cause the generated LCD waveform voltages to go out of specification for 3V and 5V glass. For LCD_GCR[HREFSEL] = 0, Vireg = 1V, but bias voltages can exceed specificationVLL1 > 1V, VLL2 > 2V, and VLL3 >3V. For LCD_GCR[HREFSEL] = 1 Vireg = 1.67V, but the bias voltages can exceed specification VLL1>1.67, VLL2 >3.3V, and VLL3>5V.

Workaround: Use an alternate LCD power supply configuration.

1. Vdd can be used as the LCD controller power supply.

a. For VSUPPLY[1:0] = 00 and LCD_GCR[CPSEL1] = 1, Vdd is used to drive VLL2. For 3V glass Vdd must be 2V and for 5V glass Vdd must be 3.3V.

b. For VSUPPLY[1:0] = 01 and LCD_GCR[CPSEL1] = 1, Vdd is used to drive VLL3. For 3V glass Vdd must be 3V and for 5V glass Vdd must be 5V.

2. Drive VLL3 with an external supply VSUPPLY[1:0] = 11 and LCD_GCR [RVEN] = 0.

a. For 3V glass VLL3 must equal 3V. The resistor bias network or charge pump can be used to generate VLL1 and VLL2. Note, VLL3 should never be externally driven to any level other than VDD.

e2591: TSI: TSI_SCANC[SMOD] behaves as an inactive time instead of a scan period value

Errata type: Errata

Description: TSI_SCANC[SMOD] should configure the scanning interval. TSI_SCANC[SMOD] interacts with the TSI_SCANC[AMCLKS] prescaler to configure this interval: Reference clock -> AMCLKS -> SMOD. So the interval frequency should be: Reference clock / AMCLKS prescaler / SMOD. However, it is working as an inactive time. Therefore, when an SMOD value is configured, the TSI will scan all the enabled electrodes and then be inactive for as much time as the SMOD and AMCLKS registers are configured.

Workaround: TSI_SCANC[SMOD] is designed to provide a predictable and configurable scanning interval. Depending on the scenario, there are three possible workarounds:

1. When SMOD = 0, the TSI will continuously scan without a pause between scans. This scan time is dependent on the electrode capacitance value and the current configured for the electrode. If the application will use only the out-of-range interrupt and it is not necessary to continually log each measured value after each scan, then an adequate threshold configuration and SMOD = 0 will cause the TSI to only interrupt or enable the out-of-range flag whenever there is a touch detected.

2. Manually control scanning time: use a time base to trigger single scans with the TSI_GENCS[SWTS] bit with the desired period. This workaround is recommended when a stable, predictable scan period is desired.

3. Use SMOD as a complement to the scanning time. The scanning time depends on the capacitance of the external electrodes. As mentioned above, the SMOD will currently control how much time the TSI is inactive after a scan so assumming the normal scan and the inactive time SMOD is currently determining will give a variable scanning period. If the application doesn't require an exact scanning period, this mode is recommended because adding inactive time after each scan will save power between scans.

e2638: TSI: The counter registers are not immediately updated after the EOSF bit is set.

Errata type: Errata

Description: The counter registers are not immediately updated after the end of scan event (EOSF is set). The counter registers will become available 0.25 ms after the EOSF flag is set. This also applies for the end-of-scan interrupt, as it is triggered with the EOSF flag. This behavior will occur both in continuous scan and in software triggered scan modes.

Workaround: Insert a delay of 0.25 ms or greater prior to accessing the counter registers after an end of scan event or an end of scan interrupt that is triggered by the EOSF flag. This delay does not need to be a blocking delay, so it can be executing other actions before reading the counter registers. Notice that the out-of-range flag (OUTRGF) and interrupt occur after the counters have been updated, so if the OUTRGF flag is polled or the out-of-range interrupt is used, the workaround is not necessary.

e2582: UART: Flow control timing issue can result in loss of characters

Errata type: Errata

Description: When /RTS flow control signal is used in receiver request-to-send mode, the /RTS signal is negated if the number of characters in the Receive FIFO is equal to or greater than the receive watermark. The /RTS signal will not negate until after the last character (the one that makes the condition for /RTS negation true) is completely received and recognized. This creates a delay between the end of the STOP bit and the negation of the /RTS signal. In some cases this delay can be long enough that a transmitter will start transmission of another character before it has a chance to recognize the negation of the /RTS signal (the /CTS input to the transmitter).

Workaround: For UARTs that implement an eight entry FIFO: When the FIFO is enabled, the receive watermark should be set to seven or less. This will ensure that there is space for at least one more character in the FIFO when /RTS negates. So in this case no data would be lost.

For UARTs without a FIFO (or if the FIFO is disabled): Delay might need to be added between characters on the transmit side in order to allow time for the negation of /RTS to be recognized before the next character is sent.

e4945: UART: ISO-7816 T=1 mode receive data format with a single stop bit is not supported

Errata type: Errata

Description: Transmission of ISO-7816 data frames with single stop bit is supported in T=1 mode. Currently in order to receive a frame, two or more stop bits are required. This means that 11 ETU reception based on T=1 protocol is not supported. T=0 protocol is unaffected.

Workaround: Do not send T=1, 11 ETU frames to the UART in ISO-7816 mode. Use 12 ETU transmissions for T=1 protocol instead.

e3892: UART: ISO-7816 automatic initial character detect feature not working correctly

Errata type: Errata

Description: The ISO-7816 automatic initial character detection feature does not work. The direct convention initial character can be detected correctly, but the inverse convention initial character will only be detected if the S2[MSBF] and S2[RXINV] bits are set. This defeats the purpose of the initial character detection and automatic configuration of the S2[MSBF], S2[RXINV], and C3[TXINV] bits.

Workaround: Use software to manually detect initial characters. Configure the UART with S2[MSBF] and S2[RXINV] cleared. Then check UART receive characters looking for 0x3B or 0x03. If 0x3B is received, then the connected card is direct convention. If 0x03 is received, then the connected card is inverse convention. If an inverse convention card is detected, then software should set S2[MSBF], S2[RXINV], and C3[TXINV].

e2584: UART: Possible conflicts between UART interrupt service routines and DMA requests

Errata type: Errata

Description: If the UARTn_S1[RDRF] and/or UARTn_S1[TDRE] flags are being used to generate DMA requests, there is a possible conflict that could occur if an interrupt service routine (ISR) or other code is used to clear any of the other flags in the UARTn_S1 register. The flags in the UARTn_S1 register use a side effect clearing mechanism where the procedure is to read the status register and then perform a read or write of the data register to clear the flag. If a DMA request for a flag bit is asserted while an ISR for another flag bit is executing, then in the process of clearing the ISR's flag bit, the ISR can also clear the flag bit for the DMA request, thereby negating the DMA request before the DMA responds to it. This could potentially cause servicing of the DMA event to be missed.

For example, assume a DMA request is being asserted for the RDRF flag. At the same time, the parity error flag (PF) sets and triggers an ISR. To clear the PF flag bit, the ISR must read the status register and read the data register. In the process, the RDRF flag would also be cleared, causing the DMA request to negate. If the DMA request asserts after the DMA has already prepared its next transfer, then it might still read from the data register, potentially causing an underflow.

Workaround: When possible, avoid enabling the UART for DMA requests and interrupts simultaneously. If error interrupts are needed while DMA requests are active, then the error ISR can be used to abort the current DMA transfer (by disabling the DMA request inside the UART and/or disabling the external request for the DMA channel) before clearing any error flags in the UARTn_S1 register.

e5704: UART: TC bit in UARTx_S1 register is set before the last character is sent out in ISO7816 T=0 mode

Errata type: Errata

Description: When using the UART in ISO-7816 mode, the UARTx_S1[TC] flag sets after a NACK is received, but before guard time expires.

Workaround: If using the UART in ISO-7816 mode with T=0 and a guard time of 12 ETU, check the UARTn_S1[TC] bit after each byte is transmitted. If a NACK is detected, then the transmitter should be reset.

The recommended code sequence is:

```
UART0_C2 &= ~UART_C2_TE_MASK; //make sure the transmitter is disabled at first
```

```
UART0_C3 |= UART_C3_TXDIR_MASK; //set the TX pin as output
UART0_C2 |= UART_C2_TE_MASK; //enable TX
UART0_C2 |= UART_C2_RE_MASK; //enable RX to detect NACK
for(i=0;i<length;i++)
{
while(!(UART0_S1&UART_S1_TDRE_MASK)){}
UART0_D = data[i];
while(!(UART0_S1&UART_S1_TC_MASK)){}//check for NACK
if(UART0_IS7816 & UART_IS7816_TXT_MASK)//check if TXT flag set
{
/* Disable transmit to clear the internal NACK detection counter */
UART0_C2 &= ~UART_C2_TE_MASK;
UART0_IS7816 = UART_IS7816_TXT_MASK;// write one to clear TXT
UART0_C2 |= UART_C2_TE_MASK; // re-enable transmit
}
}
UART0_C2 &= ~UART_C2_TE_MASK; //disable after transmit
```

e5928: USBOTG: USBx_USBTRC0[USBRESET] bit does not operate as expected in all cases

Errata type: Errata

Description: The USBx_USBTCR0[USBRESET] bit is not properly synchronized. In some cases using the bit can cause the USB module to enter an undefined state.

Workaround: Do not use the USBx_USBTCR0[USBRESET] bit. If USB registers need to be written to their reset states, then write those registers manually instead of using the module reset bit.

e3796: USBREG: Cannot use USB Regulator standby mode when output is powering MCU via VDD.

Errata type: Errata

Description: Cannot use USB Regulator standby mode when output is powering MCU via VDD.

Workaround: This feature is not supported.

e2686: WDOG: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

Errata type: Errata

Description: A watchdog reset while the system is in STOP or VLPS modes causes an incorrect wakeup sequence

Workaround: The watchdog should not be configured to continue operation when the system has entered STOP or VLPS.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

ARM POWERED®

Document Number: KINETIS_8N30D
Rev. 18 FEB 2013

freescale™