Mask Set Errata for Mask 1N29N

This report applies to mask 1N29N for these products:

- KINETIS_M

Errata ID	Errata Title
6946	Core: A debugger write to the I/O port might be corrupted during a processor write
6945	Core: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction
6749	I2C: The I2C_C1[MST] bit is not automatically cleared when arbitration is lost
7288	SOC : iRTC current increases when fast IRC is enabled
7289	SOC : iRTC current through Vbat increases when CPU accesses any peripheral registers and is proportional to the BUS frequency
7027	UART: During ISO-7816 T=0 initial character detection invalid initial characters are stored in the RxFIFO
7028	UART: During ISO-7816 initial character detection the parity, framing, and noise error flags can set
6472	UART: ETU compensation needed for ISO-7816 wait time (WT) and block wait time (BWT)
4647	UART: Flow control timing issue can result in loss of characters if FIFO is not enabled
7029	UART: In ISO-7816 T=1 mode, CWT interrupts assert at both character and block boundaries
7090	UART: In ISO-7816 mode, timer interrupts flags do not clear
7031	UART: In single wire receive mode UART will attempt to transmit if data is written to UART_D
5704	UART: TC bit in UARTx_S1 register is set before the last character is sent out in ISO7816 T=0 mode
7091	UART: UART_S1[NF] and UART_S1[PE] can set erroneously while UART_S1[FE] is set
7092	UART: UART_S1[TC] is not cleared by queuing a preamble or break character

e6946: Core: A debugger write to the I/O port might be corrupted during a processor write

Errata type: Errata
Description: A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.

Because of this erratum, a debugger write to the I/O port might be corrupted if it occurs while the processor is executing a write. The processor write completes successfully. However, under specific timing conditions, the matrix might incorrectly replace the debugger write data with the value zero.

This erratum does not affect debugger writes outside the I/O port region of the memory map, or debugger reads.

Conditions:

The following timing-specific conditions must all be met:

- The processor is running (not halted in Debug state).
- The debugger performs a write within the I/O port region of the memory map.
- The processor performs a write.

Implications:

The debugger might corrupt the targeted memory or configure the targeted device incorrectly.

Workaround: The debugger can work around this erratum by halting the processor in Debug state before performing writes to the I/O port region of the memory map.

e6945: Core: Processor executing at HardFault priority might enter Lockup state if an NMI occurs during a waited debugger transaction

Errata type: Errata

Description: A debugger can perform memory accesses through the Cortex-M0+ processor bus matrix while the processor is running.

Because of this erratum, the processor might erroneously enter Lockup state if a debugger-initiated access on the AHB-Lite master port is subject to wait states while the processor is running, executing at HardFault priority and taking a Non Maskable Interrupt (NMI). Under very specific timing conditions, the processor might incorrectly stack a ReturnAddress of 0xFFFFFFFE on NMI entry. On NMI return, the processor unstacks the incorrectly stacked ReturnAddress and enters Lockup state at HardFault priority.

Conditions:

The following timing-specific conditions must all be met:

- The processor is running (not halted in Debug state) and is executing at HardFault priority.
- The processor executes a single-cycle instruction at a word-aligned address.
- The debugger performs an access through the AHB-Lite master port that is subject to wait states.
- An NMI becomes pending.

Implications:

The processor stops executing the code in the HardFault handler and enters Lockup state at HardFault priority as if a fault had occurred.

Workaround: The debugger can work around this erratum by halting the processor in Debug state before performing accesses outside the Private Peripheral Bus (PPB) region of the memory map.

e6749: I2C: The I2C_C1[MST] bit is not automatically cleared when arbitration is lost

Errata type: Errata

Description: When the I2C module is used as a master device and loses bus arbitration, it correctly switches to be a slave device. The I2C_C1[MST] bit is not automatically cleared when this occurs but it does correctly operate as a slave.

Workaround: When the I2C module has been configured as a master device and the I2C_S[ARB] bit is set, indicating arbitration has been lost, the I2C_C1[MST] bit must be cleared by software before the I2C_S[ARB] bit is cleared.

e7288: SOC : iRTC current increases when fast IRC is enabled

Errata type: Errata

Description: iRTC current increases when fast IRC is enabled, whether crystal compensation is enabled or not.

NOTE : When VDD would be off then this extra current is not seen.

Workaround: External power switch should be used for VBAT. The iRTC block is permanently supplied from the VBAT domain (Battery power input), no matter if the VDD is active or not.

Data sheet already recommends to use external power switch for VBAT , this recommendation is there because Kinetis-M device does not have any internal power switch on VBAT and when the CPU accesses the RTC registers the current can increase proportional to the bus frequency. The external switch should ensure that the VBAT input pin will be switched to VDD domain when the VDD is active and stable in order to conserve the battery.

e7289: SOC : iRTC current through Vbat increases when CPU accesses any peripheral registers and is proportional to the BUS frequency

Errata type: Errata

Description: iRTC current through Vbat increases when CPU accesses any peripheral registers and is proportional to the BUS frequency .

NOTE : When VDD would be off then this extra current is not seen .

Workaround: External power switch should be used for VBAT. The iRTC block is permanently supplied from the VBAT domain (Battery power input), no matter if the VDD is active or not.

Data sheet already recommends to use external power switch for VBAT , this recommendation is there because Kinetis-M device does not have any internal power switch on VBAT and when the CPU accesses the RTC registers the current can increase proportional to the bus frequency. The external switch should ensure that the VBAT input pin will be switched to VDD domain when the VDD is active and stable in order to conserve the battery.

e7027: UART: During ISO-7816 T=0 initial character detection invalid initial characters are stored in the RxFIFO

Errata type: Errata

Description: When performing initial character detection (UART_C7816[INIT] = 1) in ISO-7816 T=0 mode with UART_C7816[ANACK] cleared, the UART samples incoming traffic looking for a valid initial character. Instead of discarding any invalid initial characters that are received, the UART will store them in the receive FIFO.

Workaround: After a valid initial charcter is detected (UART_IS7816[INITD] sets), flush the RxFIFO to discard any invalid initial characters that might have been received before the valid initial character.

e7028: UART: During ISO-7816 initial character detection the parity, framing, and noise error flags can set

Errata type: Errata

Description: When performing initial character detection (UART_C7816[INIT] = 1) in ISO-7816 mode the UART should not set error flags for any receive traffic before a valid initial character is detected, but the UART will still set these error flags if any of the conditions are true.

Workaround: After a valid initial charcter is detected (UART_IS7816[INITD] sets), check the UART_S1[NF, FE, and PF] flags. If any of them are set, then clear them.

e6472: UART: ETU compensation needed for ISO-7816 wait time (WT) and block wait time (BWT)

Errata type: Errata

Description: When using the default ISO-7816 values for wait time integer (UARTx_WP7816T0[WI]), guard time FD multiplier (UARTx_WF7816[GTFD]), and block wait time integer (UARTx_WP7816T1[BWI]), the calculated values for Wait Time (WT) and Block Wait Time (BWT) as defined in the Reference Manual will be 1 ETU less than the ISO-7816-3 requirement.

Workaround: To comply with ISO-7816 requirements, compensation for the extra 1 ETU is needed. This compensation can be achieved by using a timer, such as the low-power timer (LPTMR), to introduce a 1 ETU delay after the WT or BWT expires.

e4647: UART: Flow control timing issue can result in loss of characters if FIFO is not enabled

Errata type: Errata

Description: On UARTx modules with FIFO depths greater than 1, when the /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: Always enable the RxFIFO if you are using flow control for UARTx modules with FIFO depths greater than 1. 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.

Note that only UARTx modules with FIFO depths greater than 1 are affected. The UARTs that do not have the RxFIFO feature are not affected. Check the Reference Manual for your device to determine the FIFO depths that are implemented on the UARTx modules for your device.

e7029: UART: In ISO-7816 T=1 mode, CWT interrupts assert at both character and block boundaries

Errata type: Errata

Description: When operating in ISO-7816 T=1 mode and switching from transmission to reception block, the character wait time interrupt flag (UART_IS7816[CWT]) should not be set, only block type interrupts should be valid. However, the UART can set the CWT flag while switching from transmit to receive block and at the start of transmit blocks.

Workaround: If a CWT interrupt is detected at a block boundary instead of a character boundary, then the interrupt flag should be cleared and otherwise ignored.

e7090: UART: In ISO-7816 mode, timer interrupts flags do not clear

Errata type: Errata

Description: In ISO-7816, when any of the timer counter expires, the corresponding interrupt status register bits gets set. The timer register bits cannot be cleared by software without additional steps, because the counter expired signal remains asserted internally. Therefore, these bits can be cleared only after forcing the counters to reload.

Workaround: Follow these steps to clear the UART_IS7816 WT, CWT, or BWT bits:

1. Clear the UART_C7816[ISO_7816E] bit, to temporarily disable ISO-7816 mode.
2. Write 1 to the WT, CWT, or BWT bits that need to be cleared.
3. Set UART_C7816[ISO_7816E] to re-enable ISO-7816 mode.

Note that the timers will start counting again as soon as the ISO_7816E bit is set. To avoid unwanted timeouts, software might need to wait until new transmit or receive traffic is expected or desired before re-enabling ISO-7816 mode.

e7031: UART: In single wire receive mode UART will attempt to transmit if data is written to UART_D

Errata type: Errata

Description: If transmit data is loaded into the UART_D register while the UART is configured for single wire receive mode, the UART will attempt to send the data. The data will not be driven on the pin, but it will be shifted out of the FIFO and the UART_S1[TDRE] bit will set when the character shifting is complete.

Workaround: Do not queue up characters to transmit while the UART is in receive mode. Always write UART_C3[TXDIR] = 1 before writing to UART_D in single wire mode.

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
```

e7091: UART: UART_S1[NF] and UART_S1[PE] can set erroneously while UART_S1[FE] is set

Errata type: Errata

Description: While the UART_S1[FE] framing error flag is set the UART will discard any received data. Even though the data is discarded, if characters are received that include noise or parity errors, then the UART_S1[NF] or UART_S1[PE] bits can still set. This can lead to triggering of unwanted interrupts if the parity or noise error interrupts are enabled and framing error interrupts are disabled.

Workaround: If a framing error is detected (UART_S1[FE] = 1), then the noise and parity error flags can be ignored until the FE flag is cleared. Note: the process to clear the FE bit will also clear the NF and PE bits.

e7092: UART: UART_S1[TC] is not cleared by queuing a preamble or break character

Errata type: Errata

Description: The UART_S1[TC] flag can be cleared by first reading UART_S1 with TC set and then performing one of the following: writing to UART_D, queuing a preamble, or queuing a break character. If the TC flag is cleared by queuing a preamble or break character, then the flag will clear as expected the first time. When TC sets again, the flag can be cleared by any of the three clearing mechanisms without reading the UART_S1 register first. This can cause a TC flag occurrence to be missed.

Workaround: If preamble and break characters are never used to clear the TC flag, then no workaround is required.

If a preamble or break character is used to clear TC, then write UART_D immediately after queuing the preamble or break character.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: KINETIS_M_1N29N
Rev. 12 Aug 2014