Mask Set Errata for Mask 1N89E

This document contains errata information for Kinetis Mask Set 1N89E but excludes any information on security-related modules.

A nondisclosure agreement (NDA) is required for any security-related module information.

For more information on obtaining an NDA, please contact your local Freescale sales representative.

Mask Set Errata for Mask 1N89E

Introduction

This report applies to mask 1N89E for these products:

- KINETIS

Errata ID	Errata Title
4588	DMAMUX: When using PIT with "always enabled" request, DMA request does not deassert correctly
5751	FTFx: Launching the Read 1's Section command (RD1SEC) on an entire flash block results in access error (ACCER).
5706	FTFx: MCU security is inadvertently enabled (secured) if a mass erase is executed when the flash blocks/ halves are swapped. This issue only affects applications that use the flash swap feature.
5499	MCG: A reset or interrupt request due to a PLL loss of lock (LOL) condition will not occur asynchronously
4590	MCG: Transitioning from VLPS to VLPR low power modes while in BLPI clock mode is not supported.
6665	Operating requirements: Limitation of the device operating range
6348	PMC: Incorrect reset source indication when waking up from VLLS0 mode.
5130	SAI: Under certain conditions, the CPU cannot reenter STOP mode via an asynchronous interrupt wakeup event
5472	SMC: Mode transition VLPR->VLLS0(POR disabled)->RUN, will cause POR & LVD.
5952	SMC: Wakeup via the LLWU from LLS/VLLS to RUN to VLPR incorrectly triggers an immediate wakeup from the next low power mode entry
4647	UART: Flow control timing issue can result in loss of characters if FIFO is not enabled
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

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.

e5751: FTFx: Launching the Read 1's Section command (RD1SEC) on an entire flash block results in access error (ACCER).

Errata type: Errata

Description: FTFx: Launching the Read 1's Section command on an entire flash block (i.e. with flash address = flash block base address & number of longwords = total number of longwords in the flash block) results in an incorrectly asserted access error (ACCER).

Workaround: To verify an entire flash block, use the Read 1's Block command. Use the Read 1's Section command only to verify sections that are smaller than an entire flash block.

e5706: FTFx: MCU security is inadvertently enabled (secured) if a mass erase is executed when the flash blocks/halves are swapped. This issue only affects applications that use the flash swap feature.

Errata type: Errata

Description: When the logical addresses of the flash blocks (halves) are swapped via the flash swap control command sequence and a mass erase is executed (via the MDM-AP or EzPort), the MCU security can go from un-secure to secure. Thus, when using a debugger to erase the entire flash memory and re-download a software application, the debugger may report that the device is secure after the erase completes. This issue only affects applications that use the flash swap feature.

Workaround: Issue the mass erase request (via the MDM-AP or EzPort) a second time to un-secure the device.

e5499: MCG: A reset or interrupt request due to a PLL loss of lock (LOL) condition will not occur asynchronously

Errata type: Errata

Description: If a PLL loss of lock condition exists, a reset or interrupt request will not occur asynchronously when the MCG is configured, using the LOLRE or LOLIE0 bits, to generate a reset or an interrupt request upon a loss of lock condition.

Workaround: System designs should expect that any resets or interrupts that occur as a result of a loss of lock condition are synchronous.

e4590: MCG: Transitioning from VLPS to VLPR low power modes while in BLPI clock mode is not supported.

Errata type: Errata

Description: Transitioning from VLPS mode back to VLPR (LPWUI control bit = 0) while using BLPI clock mode only, is not supported. During Fast IRC startup, the output clock frequency may exceed the maximum VLPR operating frequency. This does not apply to the BLPE clock mode.

Workaround: There are two options for workarounds

a) Exit to Run instead of VLPR. Before entering VLPR set the LPWUI bit so that when exiting VLPS mode the MCU exits to RUN mode instead of VLPR mode. With LPWUI set any interrupt will exit VLPR or VLPS back into RUN mode. To minimize the impact of the higher RUN current re-enter VLPR quickly.

or

b) Utilize MCG clock mode BLPE when transitioning from VLPS to VLPR modes.

e6665: Operating requirements: Limitation of the device operating range

Errata type: Errata

Description: Some devices, when power is applied, may not consistently begin to execute code under certain voltage and temperature conditions. Applications that power up with either VDD >= 2.0 V or temperature >= -20C are not impacted. Entry and exit of low-power modes is not impacted.

Workaround: To avoid this unwanted behavior, one or both of these conditions must be met:

a) Perform power on reset of the device with a supply voltage (VDD) equal-to or greater-than 2.0 V , or

b) Perform power on reset of the device at a temperature at or above -20 C.

e6348: PMC: Incorrect reset source indication when waking up from VLLS0 mode.

Errata type: Errata

Description: If MCU exits VLLS0 with the VBAT pin floating and with POR detect circuit enabled (SMC_VLLSCTRL[PORPO] = 0), the MCU may incorrectly indicate tamper detect as a reset source in the RCM_SRS1 register after it wakes up. Under these conditions, the indication of a tamper detect as a false reset source occurs regardless of whether the device features the tamper detection unit (DryIce).

Workaround: Either ensure that VBAT is powered when MCU is exiting VLLS0 mode or disable the POR detect circuit (SMC_VLLSCTRL[PORPO] = 1) before entering VLLS0 mode.

e5130: SAI: Under certain conditions, the CPU cannot reenter STOP mode via an asynchronous interrupt wakeup event

Errata type: Errata

Description: If the SAI generates an asynchronous interrupt to wake the core and it attempts to reenter STOP mode, then under certain conditions the STOP mode entry is blocked and the asynchronous interrupt will remain set.

This issue applies to interrupt wakeups due to the FIFO request flags or FIFO warning flags and then only if the time between the STOP mode exit and subsequent STOP mode reentry is less than 3 asynchronous bit clock cycles.

Workaround: Ensure that at least 3 bit clock cycles elapse following an asynchronous interrupt wakeup event, before STOP mode is reentered.

e5472: SMC: Mode transition VLPR->VLLS0(POR disabled)->RUN, will cause POR & LVD.

Errata type: Errata

Description: The Mode transition of VLPR into VLLS0 (POR disabled) then Exit, with LLWU event, back to to RUN mode will cause a POR and LVD reset instead of the expected WAKEUP exit.

Workaround: The recommendation is to transition from VLPR to RUN before entering VLLS0 with POR disabled mode.

e5952: SMC: Wakeup via the LLWU from LLS/VLLS to RUN to VLPR incorrectly triggers an immediate wakeup from the next low power mode entry

Errata type: Errata

Description: Entering VLPR immediately after an LLWU wakeup event from LLS/VLLS, will cause any subsequent entry into LLS/VLLS to fail if entry into VLPR mode occurs before clearing the pending LLWU interrupt.

Workaround: After an LLWU wakeup event from LLS/VLLS, the user must clear the LLWU interrupt prior to entering VLPR mode.

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

Errata type: Errata

Description: On UART0 and UART1 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: Always enable the RxFIFO if you are using flow control for UART0 or UART1. 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 UART0 and UART1 are affected. The UARTs that do not have the RxFIFO feature are not affected.

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.

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.

ARM® POWERED

freescale™