

# ES\_KINETIS\_V\_0N36M

## Mask Set Errata

Rev. 12 — 10 June 2025

Errata

## 1 Mask Set Errata for Mask 0N36M

### 1.1 Revision History

This report applies to mask 0N36M for these products:

- MKV30F128VLF10
- MKV30F128VLH10
- MKV30F64VLH10
- MKV30F64VLF10
- MKV30F128VFM10
- MKV30F64VFM10

Table 1. Revision History

Revision	Release Date	Significant Changes
12	6/2025	The following errata were removed. <ul style="list-style-type: none"><li>• ERR010123</li></ul> The following errata were added. <ul style="list-style-type: none"><li>• ERR052537</li></ul> The following errata were revised. <ul style="list-style-type: none"><li>• ERR006940</li></ul>
11 July 2019	7/2019	The following errata were added. <ul style="list-style-type: none"><li>• ERR050117</li></ul>
1 Dec 2017	12/2017	Initial Revision

### 1.2 Errata and Information Summary

Table 2. Errata and Information Summary

Erratum ID	Erratum Title
<a href="#">ERR052537</a>	ADC: Incorrect ADC conversions when using high speed, long sample time or asynchronous clock configurations
<a href="#">ERR008992</a>	AWIC: Early NMI wakeup not detected upon entry to stop mode from VLPR mode
<a href="#">ERR006939</a>	Core: Interrupted loads to SP can cause erroneous behavior
<a href="#">ERR009004</a>	Core: ITM can deadlock when global timestamping is enabled
<a href="#">ERR009005</a>	Core: Store immediate overlapping exception return operation might vector to incorrect interrupt
<a href="#">ERR006940</a>	Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used



Table 2. Errata and Information Summary...continued

Erratum ID	Erratum Title
<a href="#">ERR008011</a>	eDMA: Possible corruption when writing an eDMA descriptor while the eDMA is active
<a href="#">ERR050117</a>	FAC: Execute-only access control feature has been deprecated
<a href="#">ERR008906</a>	Flashloader: UART and I2C peripherals not supported by the Kinetis Flashloader
<a href="#">ERR010121</a>	FTFA: For MCUs prior to work week 14 of 2016, FSEC[MEEN] = 10 disables Mass Erase only when the MCU is secured.
<a href="#">ERR009265</a>	FTM: Incorrect match may be generated if intermediate load feature is used in toggle mode
<a href="#">ERR008162</a>	I2C: IAAS and IICIF bits in the I2C Status Register are not set properly under certain conditions after low power recovery.
<a href="#">ERR008010</a>	LLWU: CMP flag in LLWU_Fx register cleared by multiple CMP out toggles when exiting LLSx or VLLSx modes.
<a href="#">ERR007950</a>	LLWU: When exiting from Low Leakage Stop (LLS) mode using the comparator, the comparator ISR is serviced before the LLWU ISR
<a href="#">ERR007993</a>	MCG: FLL frequency may be incorrect after changing the FLL reference clock
<a href="#">ERR007735</a>	MCG: IREFST status bit may set before the IREFS multiplexor switches the FLL reference clock
<a href="#">ERR007914</a>	PIT: After enabling the Periodic Interrupt Timer (PIT) clock gate, an attempt to immediately enable the PIT module may not be successful.
<a href="#">ERR008361</a>	SMC: Compute Operation cannot be enabled in HSRUN mode
<a href="#">ERR008184</a>	UART: During ISO-7816 T=0, TC bit set at 12 ETUs may cause loss of characters when UART is switched from transmit to receive mode
<a href="#">ERR004647</a>	UART: Flow control timing issue can result in loss of characters if FIFO is not enabled
<a href="#">ERR007857</a>	UART: WT timer in T=0 mode and CWT timer in T=1 mode can expire between 0.2 ETU to 0.8 ETU earlier than programmed.
<a href="#">ERR009646</a>	WDOG: Unexpected watchdog behavior on LLS exit

## 2 Known Errata

### ERR052537: ADC: Incorrect ADC conversions when using high speed, long sample time or asynchronous clock configurations

#### Description

When the ADC is configured to use high speed operation (ADHSC=1), long sample time (LSMP=1), or the asynchronous clock (ADICLK=11), the input comparator of the ADC can fail over time leading to incorrect conversion results.

#### Workaround

For new unused devices, do not use high speed configuration (ADHSC=1), long sample time (LSMP=1) or the asynchronous clock input (ADICLK=11).

### ERR008992: AWIC: Early NMI wakeup not detected upon entry to stop mode from VLPR mode

#### Description

Upon entry into VLPS from VLPR, if NMI is asserted before the VLPS entry completes, then the NMI does not generate a wakeup to the MCU. However, the NMI interrupt will occur after the MCU wakes up by another wake-up event.

#### Workaround

There are two workarounds:

- 1) First transition from VLPR mode to RUN mode, and then enter into VLPS mode from RUN mode.
- 2) Assert NMI signal for longer than 16 bus clock cycles.

### ERR006939: Core: Interrupted loads to SP can cause erroneous behavior

#### Description

Arm Errata 752770: Interrupted loads to SP can cause erroneous behavior

This issue is more prevalent for user code written to manipulate the stack. Most compilers will not be affected by this, but please confirm this with your compiler vendor. MQX™ and FreeRTOS™ are not affected by this issue.

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

If an interrupt occurs during the data-phase of a single word load to the stack-pointer (SP/R13), erroneous behavior can occur. In all cases, returning from the interrupt will result in the load instruction being executed an additional time. For all instructions performing an update to the base register, the base register will be erroneously updated on each execution, resulting in the stack-pointer being loaded from an incorrect memory location.

The affected instructions that can result in the load transaction being repeated are:

- 1) LDR SP,[Rn],#imm
- 2) LDR SP,[Rn,#imm]!
- 3) LDR SP,[Rn,#imm]
- 4) LDR SP,[Rn]
- 5) LDR SP,[Rn,Rm]

The affected instructions that can result in the stack-pointer being loaded from an incorrect memory address are:

- 1) LDR SP,[Rn],#imm
- 2) LDR SP,[Rn,#imm]!

Conditions:

- 1) An LDR is executed, with SP/R13 as the destination.
- 2) The address for the LDR is successfully issued to the memory system.
- 3) An interrupt is taken before the data has been returned and written to the stack-pointer.

Implications:

Unless the load is being performed to Device or Strongly-Ordered memory, there should be no implications from the repetition of the load. In the unlikely event that the load is being performed to Device or Strongly-Ordered memory, the repeated read can result in the final stack-pointer value being different than had only a single load been performed.

Interruption of the two write-back forms of the instruction can result in both the base register value and final stack-pointer value being incorrect. This can result in apparent stack corruption and subsequent unintended modification of memory.

## Workaround

Most compilers are not affected by this, so a workaround is not required.

However, for hand-written assembly code to manipulate the stack, both issues may be worked around by replacing the direct load to the stack-pointer, with an intermediate load to a general-purpose register followed by a move to the stack-pointer.

If repeated reads are acceptable, then the base-update issue may be worked around by performing the stack pointer load without the base increment followed by a subsequent ADD or SUB instruction to perform the appropriate update to the base register.

## ERR009004: Core: ITM can deadlock when global timestamping is enabled

### Description

ARM ERRATA 806422

The Cortex-M4 processor contains an optional Instrumentation Trace Macrocell (ITM). This can be used to generate trace data under software control, and is also used with the Data Watchpoint and Trace (DWT) module which generates event driven trace. The processor supports global timestamping. This allows count values from a system-wide counter to be included in the trace stream.

When connected directly to a CoreSight funnel (or other component which holds ATREADY low in the idle state), the ITM will stop presenting trace data to the ATB bus after generating a timestamp packet. In this condition, the ITM\_TCR.BUSY register will indicate BUSY.

Once this condition occurs, a reset of the Cortex-M4 is necessary before new trace data can be generated by the ITM.

Timestamp packets which require a 5 byte GTS1 packet, or a GTS2 packet do not trigger this erratum. This generally only applies to the first timestamp which is generated.

Devices which use the Cortex-M optimized TPIU (CoreSight ID register values 0x923 and 0x9A1) are not affected by this erratum.

## Workaround

There is no software workaround for this erratum. If the device being used is susceptible to this erratum, you must not enable global timestamping.

## ERR009005: Core: Store immediate overlapping exception return operation might vector to incorrect interrupt

### Description

Arm Errata 838869: Store immediate overlapping exception return operation might vector to incorrect interrupt

Affects: Cortex-M4, Cortex-M4F

Fault Type: Programmer Category B Rare

Fault Status: Present in: r0p0, r0p1 Open.

The Cortex-M4 includes a write buffer that permits execution to continue while a store is waiting on the bus. Under specific timing conditions, during an exception return while this buffer is still in use by a store instruction, a late change in selection of the next interrupt to be taken might result in there being a mismatch between the interrupt acknowledged by the interrupt controller and the vector fetched by the processor.

### Configurations Affected

This erratum only affects systems where writeable memory locations can exhibit more than one wait state.

## Workaround

For software not using the memory protection unit, this erratum can be worked around by setting DISDEFWBUF in the Auxiliary Control Register.

In all other cases, the erratum can be avoided by ensuring a DSB occurs between the store and the BX instruction. For exception handlers written in C, this can be achieved by inserting the appropriate set of intrinsics or inline assembly just before the end of the interrupt function, for example:

ARMCC:

```
...  
schedule_barrier();  
asm{DSB};  
schedule_barrier();  
}
```

GCC:

```
...  
asm volatile ("dsb 0xf" ::: "memory"); volatile ("dsb 0xf" ::: "memory");
```

}

## ERR006940: Core: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

### Description

Arm Errata 776924: VDIV or VSQRT instructions might not complete correctly when very short ISRs are used

Affects: Cortex-M4F

Fault Type: Programmer Category B

Fault Status: Present in: r0p0, r0p1 Open.

On Cortex-M4 with FPU, the VDIV and VSQRT instructions take 14 cycles to execute. When an interrupt is taken a VDIV or VSQRT instruction is not terminated, and completes its execution while the interrupt stacking occurs. If lazy context save of floating point state is enabled then the automatic stacking of the floating point context does not occur until a floating point instruction is executed inside the interrupt service routine.

Lazy context save is enabled by default. When it is enabled, the minimum time for the first instruction in the interrupt service routine to start executing is 12 cycles. In certain timing conditions, and if there is only one or two instructions inside the interrupt service routine, then the VDIV or VSQRT instruction might not write its result to the register bank or to the FPSCR.

### Workaround

A workaround is only required if the floating point unit is present and enabled. A workaround is not required if the memory system inserts one or more wait states to every stack transaction.

There are two workarounds:

- 1) Disable lazy context save of floating point state by clearing LSPEN to 0 (bit 30 of the FPCCR at address 0xE000EF34).
- 2) Ensure that every interrupt service routine contains more than 2 instructions in addition to the exception return instruction.

## ERR008011: eDMA: Possible corruption when writing an eDMA descriptor while the eDMA is active

### Description

If software writes to the SADDR, DADDR, or NBYTES field of an idle eDMA channel at the same time the eDMA engine is writing the same field of an active channel, the eDMA engine write will corrupt the software write.

### Workaround

There are two potential workarounds for this issue:

Ensure that the eDMA is idle before writing TCD values by following this procedure:

- 1) Halt the DMA via the HALT bit or by disabling each channel.
- 2) Verify DMA is inactive via the ACTIVE bit.
- 3) Write the TCD information.

4) Re-enable the DMA (clear the HALT bit or enable each channel).

-or-

Verify all writes to SADDR, DADDR, and NBYTES before starting a DMA channel or enabling the external request:

1) Write SADDR, DADDR, or NBYTES.

2) Read-back and verify contents of each field.

3) If contents of any fields do not match the expected values, return to step 1.

4) After all contents have been verified as written correctly, then the DMA channel can be started using a software request or the external request for the channel can be enabled.

## **ERR050117: FAC: Execute-only access control feature has been deprecated**

### **Description**

The FAC feature is no longer recommended for use.

### **Workaround**

Do not program the XACCn registers to use the FAC feature.

## **ERR008906: Flashloader: UART and I2C peripherals not supported by the Kinetic Flashloader**

### **Description**

The UART and I2C peripherals are not supported by the Kinetic Flashloader based on the following table:

Port Signal

PTC3 UART1\_TX

PTC4 UART1\_RX

PTB0 I2C0\_SCL

PTB1 I2C0\_SDA

PTE16 SPI\_PCS0

PTE17 SPI\_SCK

PTE18 SPI\_SOUT

PTE19 SPI\_SIN

However, the SPI peripheral is not impacted and is supported.

### **Workaround**

For support of the UART and I2C peripherals, obtain MK material marked with date codes of XXXX1445 or greater.

**ERR010121: FTFA: For MCUs prior to work week 14 of 2016, FSEC[MEEN] = 10 disables Mass Erase only when the MCU is secured.****Description**

MCUs manufactured prior to work week 14 of 2016 do not have the following features:

1) FSEC[MEEN] = 10 disables Mass Erase in all MCU operations including

a) debug mode using MDM-AP

b) EZPORT

c) internal Erase all block commands

2) Flash commands

a) 0x4A Read 1s All Execute only Segments

b) 0x4B Erase All Execute-only Segments

**Workaround**

If these features are required, please obtain MCUs manufactured during work week 14 of 2016 or later.

Read the Flash Version ID via the Read Resource Flash command or look at the date code marked on the device to confirm the availability of the features referenced above.

- Flash Version ID 08.01.01.00 indicates the features are available.

- Date Code 1614 (YYWW) or EN (YW) and later indicates the features are available.

Note: The last line marked on the part contains the date code.

- For parts with 8 characters on the last line, the date code is characters 4-7

- For parts with 6 characters on the last line, the date code is characters 4-5

- For parts with 5 characters on the last line, the date code is characters 3-4

**ERR009265: FTM: Incorrect match may be generated if intermediate load feature is used in toggle mode****Description**

When a channel (n) match is used as an intermediate reload, an incorrect second match may occur immediately following the correct match. The issue is problematic only if channel (n) is configured for output compare with the output configured to toggle mode. In this scenario, channel (n) toggles on the correct match and again on the incorrect match. The issue may also occur if a certain channel has a match which is coincident with an intermediate reload point of any other channel.

**Workaround**

If any channel is configured for output compare mode with the output set for toggle mode, the intermediate reload feature must not be used.

**ERR008162: I2C: IAAS and IICIF bits in the I2C Status Register are not set properly under certain conditions after low power recovery.****Description**

The IAAS and IICIF bits in the I2C Status Register indicating an address match and a pending interrupt will not be set if recovering from a low power mode and the Bus Clock is less than  $\frac{1}{4}$  of the Core Clock. Only the TCF flag will be set.

**Workaround**

When recovering from LLSx/VLLSx modes, the Asynchronous Wake-up Interrupt Controller (AWIC) correctly identifies the source of wakeup and can be used to confirm valid I2C addressing. For other low power modes, the TCF flag in conjunction with a software semaphore that establishes recovery from low power modes can be used to determine address match.

**ERR008010: LLWU: CMP flag in LLWU\_Fx register cleared by multiple CMP out toggles when exiting LLSx or VLLSx modes.****Description**

The comparator's corresponding wakeup flag in the LLWU\_Fx register is cleared prematurely if:

1. The CMP output is toggled more than one time during the LLSx wakeup sequence and the comparator's corresponding flag in the LLWU\_Fx register is cleared.

Or

2. The CMP output is toggled more than one time during the VLLSx wakeup sequence, PMC\_REGSC[ACKISO] is cleared, and the comparator's corresponding flag in the LLWU\_Fx register is cleared.

**Workaround**

When MCU is waking up from LLS, code can implement a software flag to retain the wakeup source, if required by software.

When MCU is waking up from VLLSx, code can implement a software flag prior to clearing PMC\_REGSC[ACKISO] to retain the wakeup source, if required by software.

**ERR007950: LLWU: When exiting from Low Leakage Stop (LLS) mode using the comparator, the comparator ISR is serviced before the LLWU ISR****Description**

The comparator's interrupt service routine when exiting from LLS mode is serviced before the LLWU ISR. Clearing the comparator flag in CMPx\_SCR clears the corresponding comparator flag in the LLWU\_Fx register which may be used to determine wakeup source in the LLWU ISR.

**Workaround**

Code can implement a software flag in the CMP ISR to retain wakeup source if required by software.

**ERR007993: MCG: FLL frequency may be incorrect after changing the FLL reference clock****Description**

When the FLL reference clock is switched between the internal reference clock and the external reference clock, the FLL may jump momentarily or lock at a higher than configured frequency. The higher FLL frequency can affect any peripheral using the FLL clock as its input clock. If the FLL is being used as the system clock source, FLL Engaged Internal (FEI) or FLL Engaged External (FEE), the maximum system clock frequency may be exceeded and can cause indeterminate behavior.

Only transitions from FLL External reference (FBE, FEE) to FLL Internal reference (FBI, FEI) modes and vice versa are affected. Transitions to and from BLPI, BLPE, or PLL clock modes (if supported) are not affected because they disable the FLL. Transitions between the external reference modes or between the internal reference modes are not affected because the reference clock is not changed.

**Workaround**

To prevent the occurrence of this jump in frequency either the MCG\_C4[DMX32] bit must be inverted or the MCG\_C4[DRST\_DRS] bits must be modified to a different value immediately before the change in reference clock is made and then restored back to their original value after the MCG\_S[IREFST] bit reflects the selected reference clock.

If you want to change the MCG\_C4[DMX32] or MCG\_C4[DRST\_DRS] to new values along with the reference clock, the sequence described above must be performed before setting these values to the new value(s).

**ERR007735: MCG: IREFST status bit may set before the IREFS multiplexor switches the FLL reference clock****Description**

When transitioning from MCG clock modes FBE or FEE to either FBI or FEI, the MCG\_S[IREFST] bit will set to 1 before the IREFS clock multiplexor has actually selected the slow IRC as the reference clock. The delay before the multiplexor actually switches is:

2 cycles of the slow IRC + 2 cycles of OSCERCLK

In the majority of cases this has no effect on the operation of the device.

**Workaround**

In the majority of applications no workaround is required. If there is a requirement to know when the IREFS clock multiplexor has actually switched, and OSCERCLK is no longer being used by the FLL, then wait the equivalent time of:

2 cycles of the slow IRC + 2 cycles of OSCERCLK

after MCG\_S[IREFST] has been set to 1.

**ERR007914: PIT: After enabling the Periodic Interrupt Timer (PIT) clock gate, an attempt to immediately enable the PIT module may not be successful.****Description**

If a write to the PIT module enable bit (PIT\_MCR[MDIS]) occurs within two bus clock cycles of enabling the PIT clock gate in the SIM\_CG register, the write will be ignored and the PIT will fail to enable.

**Workaround**

Insert a read of the PIT\_MCR register before writing to the PIT\_MCR register. This guarantees a minimum delay of two bus clocks to guarantee the write is not ignored.

**ERR008361: SMC: Compute Operation cannot be enabled in HSRUN mode****Description**

Compute Operation, which keeps the CPU enabled with full access to the SRAM and Flash read ports while placing all other bus masters and bus slaves into their stop mode, cannot be enabled in HSRUN mode.

**Workaround**

Transition from HSRUN to RUN mode to enable Compute Operation via the MCM\_CPO[CPOREQ]=1. Upon exiting Compute Operation (MCM\_CPO[CPOREQ]=0), you can transition from RUN back to HSRUN mode.

**ERR008184: UART: During ISO-7816 T=0, TC bit set at 12 ETUs may cause loss of characters when UART is switched from transmit to receive mode****Description**

In ISO-7816 T=0 mode, if S1[TC] is set at 12 ETUs to indicate end of transmission and software then switches the UART to receive mode by setting C2[RE], the first received character may be lost.

**Workaround**

For EMV card applications, no workaround is required since the maximum turnaround time for EMV-compliant cards is 15 ETUs, per the EMV L1 test specification (1CF.004.00).

No workaround is available for ISO-7816-compliant cards.

**ERR004647: UART: Flow control timing issue can result in loss of characters if FIFO is not enabled****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.

## **ERR007857: UART: WT timer in T=0 mode and CWT timer in T=1 mode can expire between 0.2 ETU to 0.8 ETU earlier than programmed.**

### Description

In ISO7816 receive mode, the wait timer (WT) used in T=0 mode and the character wait timer (CWT) used in T=1 mode can expire between 0.2 ETU to 0.8 ETU earlier than programmed. The early expiration of the timers can cause software to discard valid data during communication.

### Workaround

To minimize the possibility of discarded data, in T=0 mode, program the WI counter field for the wait timer as **WWT + D 480 ETUs for both receive and transmit**. ETUs for both receive and transmit.

To minimize the possibility of discarded data, in T=1 mode, program the CWI counter field for the character wait timer as **CWT + 5 ETUs when receiving and CWT + 4 ETUs when transmitting**.

## **ERR009646: WDOG: Unexpected watchdog behavior on LLS exit**

### Description

When exiting LLS mode, the watchdog counter can increment in some cases. This can cause the watchdog to timeout earlier than expected in applications where the watchdog is enabled and LLS mode is used.

### Workaround

When entering or exiting from LLS mode, refresh watchdog to avoid triggering timeout event.

## Legal information

### Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

### Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Limiting values** — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**No offer to sell or license** — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

**Quick reference data** — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Suitability for use in non-automotive qualified products** — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**HTML publications** — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

**Translations** — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

**NXP B.V.** — NXP B.V. is not an operating company and it does not distribute or sell products.

## Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

**NXP** — wordmark and logo are trademarks of NXP B.V.

---

## Contents

---

<b>1</b>	<b>Mask Set Errata for Mask 0N36M .....</b>	<b>1</b>
1.1	Revision History .....	1
1.2	Errata and Information Summary .....	1
<b>2</b>	<b>Known Errata .....</b>	<b>3</b>
	<b>Legal information .....</b>	<b>13</b>

---

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

---

© June 2025 NXP B.V.

All rights reserved.

For more information, please visit: <https://www.nxp.com>

[Document feedback](#)

Date of release: 10 June 2025

Document identifier: ES\_KINETIS\_V\_0N36M