# Wireless Gecko
# EFR32MG22 Errata

This document contains information on the EFR32MG22 errata. The latest available revision of this device is revision C.

Errata that have been resolved remain documented and can be referenced for previous revisions of this device.

The device data sheet explains how to identify the chip revision, either from package marking or electronically.

Errata effective date: September, 2020.

# 1. Errata Summary

The table below lists all known errata for the EFR32MG22 and all unresolved errata in revision C of the EFR32MG22.

**Table 1.1. Errata Overview**

| Designator | Title/Problem | Workaround Exists | Exists on Revision: | |
|---|---|---|---|---|
| | | | B | C |
| CMU_E301 | Hard Fault Exiting EM2 or EM3 with Debugger Attached | Yes | X | — |
| CMU_E302 | LFRCO Precision Mode Is Not Functional | No | X | — |
| CMU_E303 | Outputting the HFXO or HFRCO to a Pin Can Hang the Device in EM2/EM3 | Yes | X | — |
| EMU_E301 | Request for Averaged Temperature Reading Can Be Missed | Yes | X | — |
| EMU_E302 | DC-DC is Disabled after a Soft Reset | Yes | X | — |
| EMU_E303 | Watchdog Reset Hangs System Entering EM2 or EM3 | Yes | X | X |
| I2C_E303 | I2C Fails to Indicate New Incoming Data | Yes | X | X |
| RADIO_E302 | Data Whitening is not Selective | No | X | X |
| RADIO_E303 | RAIL Packet Filters Work Incorrectly When Header is Enabled | Yes | X | X |
| TIMER_E301 | Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode | Yes | X | X |
| USART_E301 | Possible Data Transmission on Wrong Edge in Synchronous Mode | Yes | X | X |
| USART_E302 | Additional SCLK Pulses Can Be Generated in USART Synchronous Mode | Yes | X | X |
| WDOG_E301 | Clear Command is Lost Upon EM2 Entry | Yes | X | — |

# 2. Current Errata Descriptions

## 2.1 EMU_E303 – Watchdog Reset Hangs System Entering EM2 or EM3

| *Description of Errata* |
|---|
| The chip can hang and require a hard reset (pin or power-on) to recover if<br>1. The system is operating with VSCALE1 core voltage scaling (software has previously written a 1 to the EMU_CMD_EM01VSCALE1 bit),<br>2. The system is in the process or entering EM2 or EM3 (software has just executed the WFE or WFI instruction with the SLEEPDEEP bit in the System Control Register set) and<br>3. A Watchdog timeout reset is triggered |
| ***Affected Conditions / Impacts*** |
| Systems operating with core voltage scaling can hang if a Watchdog reset occurs immediately upon EM2 or EM3 entry. |
| ***Workaround*** |
| Systems that keep the Watchdog enabled in low energy modes should, as a matter of good programming practice, service the Watchdog before entering EM2 or EM3. Calling the emlib `WDOGn_Feed()` function followed by the `WDOGn_SyncWait()` function (to ensure that the servicing write to the WDOG_CMD register completes execution) immediately before entering EM2 or EM3 will prevent a Watchdog reset that could possibly hang the system under the specified circumstances. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

## 2.2 I2C_E303 – I²C Fails to Indicate New Incoming Data

| *Description of Errata* |
|---|
| A race condition exists in which the I²C fails to indicate reception of new data when both user software attempts to read data from and the I²C hardware attempts to write data to the I2C_RXFIFO in the same cycle. |
| ***Affected Conditions / Impacts*** |
| When this race condition occurs, the RXFIFO enters an invalid state in which both I2C_STATUS_RXDATAV = 0 and I2C_STATUS_RXFULL = 1. This causes the I²C to discard new incoming data bytes because RXFULL = 1 and would otherwise prevent user software from reading last byte written by the I²C hardware to RXFIFO because RXDATAV = 0. |
| ***Workaround*** |
| User software can recognize and clear this invalid RXDATAV = 0 and RXFULL = 1 condition by performing a dummy read of the RXFIFO (I2C_RXDATA). This restores the expected RXDATAV = 1 and RXFULL = 0 condition. The data from this read can be discarded, and user software can now read the last byte written by the I²C hardware to the RXFIFO (the byte which caused the invalid RXDATAV = 0 and RXFULL = 1 condition).<br>No data will be lost as long as user software completes this recovery procedure (performing the dummy read and then reading the remaining valid byte in the RXFIFO) before the I²C hardware receives the next incoming data byte. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

### 2.3 RADIO_E302 – Data Whitening is not Selective

| ***Description of Errata*** |
| --- |
| Data whitening is not selective. Enabling whitening for only the packet header also whitens the payload, while enabling whitening for only the payload also whitens the packet header. |
| ***Affected Conditions / Impacts*** |
| Radio PHYs that require selective whitening of either only the packet header or the payload will not function correctly. |
| ***Workaround*** |
| There is currently no workaround for this issue. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

### 2.4 RADIO_E303 – RAIL Packet Filters Work Incorrectly When Header is Enabled

| ***Description of Errata*** |
| --- |
| When header is enabled in the radio configurator, RAIL's address filter and 15.4 packet type filter work incorrectly |
| ***Affected Conditions / Impacts*** |
| Packets that should have been filtered will be received. |
| ***Workaround*** |
| Simplicity Studio v4.1.13.6 or later and RAIL v2.8.4 in Flex SDK v2.7.4 mitigate this issue by using a workaround. The workaround is effective if:<br>• Header and payload have the same CRC & whitening configuration.<br>• Header and payload have the same whitening configuration, different CRC configuration and the header is less than 4 bytes long.<br>When an incompatible radio configuration setting is used, such as a 4-byte or longer header length with CRC disabled, RAIL generates a RAIL_ASSERT_INVALID_FILTERING_CONFIG error upon enabling filtering. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

### 2.5 TIMER_E301 — Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode

***Description of Errata***

When the TIMER is configured to operate in quadrature decoder mode with the overflow interrupt enabled and the counter value (TIMER_CNT) reaches the top value (TIMER_TOP), the overflow interrupt is requested contiunously even if the interrupt flag (TIMER_IF_OF) is cleared. Similarly, if the underflow interrupt is enabled and the counter value reaches zero, the underflow interrupt is requested contiunously even if the interrupt flag (TIMER_IF_UF) is cleared. Only after the counter value has incremented or decremented so that the overflow or underflow condition no longer applies can the interrupt be cleared.

***Affected Conditions / Impacts***

Because the counter is clocked by its CC0 and CC1 inputs in quadrature decoder mode and not the prescaled HFPERCLK, overflow and underflow events remain latched as long TIMER_CNT remains at the value that triggered the overflow or underflow condition. Until the counter is no longer in the overflow or underflow condition, it is not possible to clear the associated interrupt flag.

***Workaround***

Short of disabling the relevant interrupts, the simplest workaround is to manually increment or decrement TIMER_CNT so that the overflow or underflow condition no longer exists. Insert the following or similar code in the interrupt handler for the timer in question (TIMER0 in this case) to do this:

```
uint32 intflags = TIMER_IntGet(TIMER0);

if (intFlags & TIMER_IEN_OF)
  TIMER0->CNT += 1;

if (intFlags & TIMER_IEN_UF)
  TIMER0->CNT -= 1;
```

It may be necessary for firmware to account for this adjustment in calculations that include the counter value.

***Resolution***

There is currently no resolution for this issue.

### 2.6 USART_E301 — Possible Data Transmission on Wrong Edge in Synchronous Mode

| ***Description of Errata*** |
|---|
| The first bit of the new data word is incorrectly transmitted on the leading clock edge of the subsequent data bit and not the trailing clock edge of the current data bit if the USART is configured to operate in synchronous mode with<br>1. USART_CLKDIV_DIV = 0 (clock = $f_{HFPERCLK}$ ÷ 2),<br>2. USART_CTRL_CLKPHA = 0,<br>3. USART_TIMING_CSHOLD = 1 and<br>4. Data is loaded into the transmit FIFO (say, by the LDMA) at the exact same time as the USART state machine begins to insert the requested one bit time extension of the chip select hold time (USART_TIMING_CSHOLD = 1). |
| ***Affected Conditions / Impacts*** |
| Reception of each data bit by the slave is tied to a specific clock edge. Therefore, the late transmission by the master of the first bit of a word may cause the slave to receive the incorrect data, especially if the data setup time for the slave approaches or exceeds one half the shift clock period. |
| ***Workaround*** |
| Because there is no way to specifically time a write to the transmit FIFO such that it does not occur when the USART state machine changes state, use one of the following workarounds to avoid the risk for data corruption described above:<br>• Set USART_CLK_DIV > 0.<br>• Use USART_TIMING_CSHOLD = 0 or USART_TIMING_CSHOLD > 1.<br>• Use USART_CTRL_CLKPHA = 1. This is option is particularly useful with SPI flash memories as many support operation in both the CLKPOL = CLKPHA = 0 and CLKPOL = CLKPHA = 1 modes. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

### 2.7 USART_E302 — Additional SCLK Pulses Can Be Generated in USART Synchronous Mode

| ***Description of Errata*** |
|---|
| When inter-character spacing is enabled (USART_TIMING_ICS > 0) and USART_CTRL_CLKPHA = 1 in synchronous master mode, an extra clock pulse is generated after each frame transmitted except the last (that frame which when sent results in both the transmit FIFO and transmit shift register being empty). |
| ***Affected Conditions / Impacts*** |
| The extra clock pulse generated at the end of the first frame would cause a slave device to clock in the first bit of the next frame it expects to receive even though the USART is not yet driving that data. The slave would lose synchronization with the master and erroneously receive all frames after the first. |
| ***Workaround*** |
| Do not enable inter-character spacing when CLKPHA = 1. If a delay between frames is necessary, insert one manually with a software delay loop. Data cannot be transmiited using DMA in this case. |
| ***Resolution*** |
| There is currently no resolution for this issue. |

# 3. Resolved Errata Descriptions

This section contains previous errata for EFR32MG22 devices.

For errata on the latest revision, refer to the beginning of this document. The device data sheet explains how to identify chip revision, either from package marking or electronically.

### 3.1 CMU_E301 – Hard Fault Exiting EM2 or EM3 with Debugger Attached

| ***Description of Errata*** |
|---|
| When waking from EM2 or EM3 with a debugger attached, the CPU clock starts approximately 40 cycles in advance of the ICACHE clock. Because the CPU resumes execution before the ICACHE is ready, the data returned in response to instruction fetches is corrupted, resulting in a hard fault exception. |
| ***Affected Conditions / Impacts*** |
| Executing code that resides in flash and wakes from EM2 or EM3 while a debugger is connected causes the system to take a hard fault exception. |
| ***Workaround*** |
| Depending on the functionality required, the hard fault condition can be avoided by:<br>• Detach the debugger before entering EM2 or EM3. When the debugger is attached, certain high frequency clocks remain active in EM2 or EM3, which is why the the ICACHE clock is delayed relative to the CPU clock upon wake-up. Without the debugger connected, these clocks shutdown when entering EM2 or EM3 and restart together upon wake-up, thus avoiding the data corruption described above. Reconnect the debugger once the system is back in EM0 or EM1.<br>• Execute the WFI or WFE instruction that places the system in EM2 or EM3 from RAM. Upon wake-up, use a software delay loop to stall for the approximately 40 clock cycles of headstart that the CPU has before the ICACHE restarts.<br>• As above, execute WFI or WFE from RAM, but, instead of using a software delay, wait for the ICACHE_STATUS_PCRUNNING bit to change state from 0 to 1. The ICACHE performance counter must first be started by writing a 1 to ICACHE_CMD_STARTPC, which can be done either when running from flash before entering EM2 or EM3 or when running from RAM after wake-up. Stop the performance counter by writing a 1 to ICACHE_CMD_STOPPC. |
| ***Resolution*** |
| This issue is resolved in revision C devices. |

### 3.2 CMU_E302 – LFRCO Precision Mode Is Not Functional

| ***Description of Errata*** |
|---|
| The precision mode of the LFRCO is not functional. |
| ***Affected Conditions / Impacts*** |
| It is not possible to use the LFRCO in precision mode as a replacement for a 32.768 kHz crystal. |
| ***Workaround*** |
| There is currently no workaround for this issue. Use the LFXO and a suitable 32.768 kHz crystal in applications with such requirements. |
| ***Resolution*** |
| This issue is resolved in revision C devices. |

### 3.3 CMU_E303 — Outputting the HFXO or HFRCO to a Pin Can Hang the Device in EM2/EM3

| ***Description of Errata*** |
| --- |
| The device hangs when attempting to enter EM2 or EM3 while the HFXO or HFRCO is driven on one of the CLKOUT pins without a debugger connected. |
| ***Affected Conditions / Impacts*** |
| It is not possible to enter EM2 or EM3 when the HFXO or HFRCO is driven on one of the CLKOUT pins nor will an interrupt wake the device that has hung in this way. |
| ***Workaround*** |
| Deselect the HFXO or HFRCO on any CLKOUT pins before entering EM2 or EM3. For example, to de-select the HFXO or HFRCO on pin PC03, add the following function call before entering EM2 or EM3:<br>`CMU_ClkOutPinConfig(0, cmuSelect_Disabled, 0, gpioPortC, 3);` |
| ***Resolution*** |
| This issue is resolved in revision C devices. |

### 3.4 EMU_E301 – Request for Averaged Temperature Reading Can Be Missed

| ***Description of Errata*** |
| --- |
| Depending on the system clock frequency, the request for a hardware-averaged temperature reading is sometimes not captured, and the state machine that generates the averaged reading is never started. |
| ***Affected Conditions / Impacts*** |
| Because the averaging state machine is never started, the EMU_IF_TEMPAVIF flag is never set, and any code that depends on the averaged reading is not going to execute. |
| ***Workaround*** |
| Subsequent EMU register accesses will cause the temperature averaging request to be recognized, so the simplest solution to ensure this is to use the following code sequence:<br>`EMU->CMD_SET = EMU_CMD_TEMPAVGREQ;`<br>`while (!(EMU->STATUS & EMU_STATUS_TEMPAVGACTIVE));` |
| ***Resolution*** |
| This issue is resolved in revision C devices. |

### 3.5 EMU_E302 – DC-DC is Disabled after a Soft Reset

| ***Description of Errata*** |
| --- |
| The DC-DC converter stops regulating after a soft reset until it is re-enabled. |
| ***Affected Conditions / Impacts*** |
| When disabled, the DC-DC operates in bypass mode. Supplies connected to the DC-DC output will be powered at the VREGIN voltage which increases current consumption, until the DC-DC is re-enabled. |
| ***Workaround*** |
| On devices prior to revision C, firmware must re-enable the DC-DC after a soft reset. |
| ***Resolution*** |
| This issue has been resolved. Revision C devices with PRODREV greater than or equal to 1 will not have this issue. |

### 3.6 WDOG_E301 – Clear Command is Lost Upon EM2 Entry

| ***Description of Errata*** |
| --- |
| If the device enters EM2, while the clear command is still being synchronized, the watchdog counter may not be cleared as expected. |
| ***Affected Conditions / Impacts*** |
| If the watchdog counter is not cleared as expected, the device can encounter a watchdog reset. |
| ***Workaround*** |
| Wait for WDOG_SYNCBUSY_CMD to clear before entering EM2.<br><br>Note that WDOG can be clocked from one of the low-frequency clock sources and will require additional time to enter EM2 when implementing this workaround. |
| ***Resolution*** |
| This issue is resolved in revision C devices. |

# 4. Revision History

### Revision 0.5

September, 2020

- Added I2C_E303.
- Clarified the affected conditions and impacts in WDOG_E301.
- Updated affected revisions and resolution for TIMER_E301.

### Revision 0.4

May, 2020

- Added RADIO_E303 and USART_E302.

### Revision 0.3

January, 2020

- Added EMU_E303 and RADIO_E302.

### Revision 0.2

October, 2019

- Updated to product revision C.
- Added CMU_E303, EMU_E302, TIMER_E301 and WDOG_E301.
- Resolved CMU_E301, CMU_E302 and EMU_E301
- Migrated to new errata document format.

### Revision 0.1

July, 2019

- Initial release.

# Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

**IoT Portfolio**
www.silabs.com/IoT

**SW/HW**
www.silabs.com/simplicity

**Quality**
www.silabs.com/quality

**Support & Community**
www.silabs.com/community



SILICON LABS


Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

**http://www.silabs.com**