



Wireless Gecko™ Multiprotocol Lighting Module MGM210L Errata

This document contains information on the MGM210L errata. The latest available revision of this device is revision 3.

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

Errata effective date: March, 2026.

1. Errata Summary

The following table lists all the known and unresolved errata for the MGM210L.

Table 1.1. Errata Overview

Designator	Title/Problem	Workaround Exists	Exists on Revision	
			2	3
GPIO_E302	Increased Leakage Current When EM4WU Pins Are Enabled and the Pin State Is High	Yes	X	X
HFXO_E301	HFXO DISONDEMAND and FORCEEN Can Cause Device to Hang	Yes	X	X
I2C_E303	I ² C Fails to Indicate New Incoming Data	Yes	X	X
IADC_E304	Possible Data Loss in EM2/EM3	Yes	X	X
IADC_E306	Changing Gain During a Scan Sequence Causes an Erroneous IADC Result	Yes	X	X
IADC_E307	Immediate Conversion Trigger When Configured for PRS Edge	Yes	X	X
SE_E302	DPA Countermeasure Unavailable for Some Operations	Yes	X	X
TIMER_E301	Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode	Yes	X	X
TIMER_E302	Interrupts Do Not Correspond to ICEVCTRL Setting	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
USART_E304	PRS Transmit Unavailable in Synchronous Secondary Mode	No	X	X
WDOG_E301	Clear Command is Lost Upon EM2 Entry	Yes	X	X

2. Current Errata Descriptions

2.1 GPIO_E302 – Increased Leakage Current When EM4WU Pins Are Enabled and the Pin State Is High

Description of Errata
When any of the EM4WU pins are used with the input path enabled and the pin state is high, an extra leakage current of approximately 15 μ A per pin will be observed in EM0, EM1, EM2, and EM3.
Affected Conditions / Impacts
EM0, EM1, EM2, and EM3 current will be higher by approximately 15 μ A per pin when any of the EM4WU pins are used with the input path enabled and the pin state is high.
Workaround
There are two workarounds for this issue: <ol style="list-style-type: none"> 1. If the input path on the pad is not required, disable the input path on that pad by setting the DINDIS or DINDISALT bits in the GPIO_PORTX_CTRL register. Thus, an EM4WU pin can still be used to drive an output without incurring the extra current leakage when the pin is configured as an output and DINDIS or DINDISALT is set. 2. If an input path is required (i.e., MODEN is any value other than DISABLED and DINDIS = 0 or DINDISALT = 0), assign it to a pin which does not have EM4 wakeup capability. <p>Refer to the device data sheet to determine which pins have or do not have EM4 wake-up functionality.</p>
Resolution
There is currently no resolution for this issue.

2.2 HFXO_E301 — HFXO DISONDEMAND and FORCEEN Can Cause Device to Hang

Description of Errata
With HFXO enabled, when DISONDEMAND is toggled from 0 to 1 followed by a system reset request, a handshake between the EMU and CMU hangs, preventing the system reset from being asserted.
Affected Conditions / Impacts
The device will hang waiting for the EMU/CMU handshake to complete, requiring a pin reset to recover.
Workaround
When the HFXO is enabled, do not toggle DISONDEMAND from 0 to 1.
Resolution
There is currently no resolution for this issue.

2.3 I2C_E303 – I²C Fails to Indicate New Incoming Data

Description of Errata
A race condition exists in which the I ² C fails to indicate reception of new data when both user software attempts to read data from and the I ² C 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 I ² C to discard new incoming data bytes because RXFULL = 1 and would otherwise prevent user software from reading the last byte written by the I ² C 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 dummy read also sets the RXUFIF flag bit, which should be ignored and cleared. The data from this read can be discarded, and user software can now read the last byte written by the I ² C hardware to the RXFIFO (the byte which caused the invalid RXDATAV = 0 and RXFULL = 1 condition). 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 I ² C hardware receives the next incoming data byte.
Resolution
There is currently no resolution for this issue.

2.4 IADC_E304 – Possible Data Loss in EM2/EM3

Description of Errata
When the IADC wakes from EM2 or EM3 and generates conversion results that the LDMA transfers to RAM, it is possible under very rare circumstances to lose data when the ratio of the bus clock (HCLK) is slow compared to the prescaled IADC clock (ADC_CLK).
Affected Conditions / Impacts
Data from IADC conversions in these cases can potentially be lost due to FIFO overflow.
Workaround
To prevent data loss when the IADC awakens from EM2 or EM3 and performs conversions that are serviced by the LDMA before re-entering the low-energy state, make sure that: <ul style="list-style-type: none"> the rate at which the IADC takes samples in EM2 or EM3 is less than or equal to 125 kHz (samples are taken no faster than every 8 μs), and the frequency of the HCLK (bus clock) is at least four times the frequency of the IADCCLK.
Resolution
There is currently no resolution for this issue.

2.5 IADC_E306 – Changing Gain During a Scan Sequence Causes an Erroneous IADC Result

Description of Errata
Differences in the ANALOGGAIN setting within multiple IADC_CFGx groups during a scan sequence introduces a transient condition that may result in an inaccurate IADC conversion.
Affected Conditions / Impacts
The result of the IADC scan measurement may not match the expected result for the voltage present on the pin during the conversion.
Workaround
Both 1 and 2 shown below must be implemented. <ol style="list-style-type: none"> 1. If there is a difference in the ANALOGGAIN setting between IADC_CFGx groups during a scan sequence, the IADC_SCHEx clock prescaler must also change to an appropriate setting. This forces a warmup state (5 μs delay) in between ANALOGGAIN changes. Note that the same IADC_SCHEx clock prescaler value may be an appropriate setting for both ANALOGGAIN settings, but to force the warmup delay, the IADC_SCHEx must have different values. 2. The first and last entry of a scan group should use IADC_CFG0, which is the default configuration of the IADC at the start and end of a scan conversion sequence. If CONFIG1 is used at the start and end of the scan group, erroneous IADC results may occur.
Resolution
There is currently no resolution for this issue.

2.6 IADC_E307 – Immediate Conversion Trigger When Configured for PRS Edge

Description of Errata
Selecting PRSPOS as the single or scan conversion trigger when the PRS input is high immediately triggers a conversion upon writing to the IADC_CMD_SINGLESTART or IADC_CMD_SCANSTART bits. The same applies when starting a conversion if the PRS input is low and PRSNEG is the trigger.
Affected Conditions / Impacts
An immediate PRS trigger is registered when starting a conversion in the described configuration.
Workaround
There are multiple workarounds for this issue: <ol style="list-style-type: none"> 1. When using the PRSPOS trigger configuration, make sure the PRS input is low when starting the conversion. 2. When using the PRSNEG trigger configuration, make sure the PRS input is high when starting the conversion. 3. If a conversion is started when the PRSPOS trigger is selected and the PRS input is high, throw away the first result. 4. If a conversion is started when the PRSNEG trigger is selected and the PRS input is low, throw away the first result.
Resolution
There is currently no resolution for this issue.

2.7 SE_E302 – DPA Countermeasure Unavailable for Some Operations

Description of Errata
Differential power analysis (DPA) countermeasures for ECDH on Curve25519, ECDH on Curve448, and EdDSA signing on Curve25519 are unavailable due to a lack of hardware support on all Series 2 devices with a Hardware Secure Engine (HSE).
Affected Conditions / Impacts
A successful DPA attack may be possible if the impacted algorithms are implemented in a customer's product. However, a DPA attack is not an easy/straightforward attack as it requires specific equipment, many traces, physical access to the device, and some control over device operation. If a successful DPA attack occurs, an attacker may be able to gain access to confidential information, such as private keys or encrypted communications between devices.
Workaround
No fix is available to provide the affected DPA countermeasures on Series 2 devices. Refer to Security Advisory A-00000534 for mitigation recommendations, which include refreshing key pairs or using a key pair only once to reduce the risk of a successful DPA attack.
Resolution
There is currently no resolution for this issue.

2.8 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 continuously 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 continuously 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 HPERCLK, 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:
<pre>uint32 intFlags = TIMER_IntGet(TIMER0); if (intFlags & TIMER_IEN_OF) TIMER0->CNT += 1; if (intFlags & TIMER_IEN_UF) TIMER0->CNT -= 1;</pre>
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.

2.9 TIMER_E302 — Interrupts Do Not Correspond to ICEVCTRL Setting

<p>Description of Errata</p> <p>Input capture event control (TIMER_CC_CTRL_ICEVCTRL) does not work as expected for all input capture edge settings (TIMER_CC_CTRL_ICEDGE).</p> <p>Specifically, the TIMER requests interrupts on the wrong edges of an incoming signal (such as a PWM waveform) when:</p> <ul style="list-style-type: none"> • TIMER_CC_CTRL_ICEDGE = BOTH • TIMER_CC_CTRL_ICEVCTRL = RISING or FALLING
<p>Affected Conditions / Impacts</p> <p>When ICEVCTRL = RISING, instead of being requested on every rising edge:</p> <ul style="list-style-type: none"> • The first interrupt is requested on the first rising edge and one capture result is written to TIMER_CCx_ICF. • The second interrupt is requested on the first falling edge and one capture result is written to TIMER_CCx_ICF. • The third interrupt is requested on the second falling edge and two capture results are written to TIMER_CCx_ICF. • Subsequent interrupts are consistently requested on falling edges. <p>When ICEVCTRL = FALLING, instead of being requested on every falling edge:</p> <ul style="list-style-type: none"> • The first interrupt is requested on the second rising edge, two capture values are written to TIMER_CCx_ICF, a third is written to TIMER_CCx_ICOF, and the ICOF_x flag corresponding to the capture/compare channel in use is set in the TIMER_IF register. • The second interrupt is requested on the third rising edge and two capture results are written to TIMER_CCx_ICF. • Subsequent interrupts are consistently requested on rising edges. <p>The behavior of TIMER capture/compare PRS producers is similarly affected such that occurrences of "interrupt(s)" and "requested" may be replaced with "PRS pulse(s)" and "generated" in the previous descriptions.</p>
<p>Workaround</p> <ol style="list-style-type: none"> 1. Set ICEVCTRL to the opposite of the desired edge. For example, select FALLING when rising edges are required, and select RISING when falling edges are required. 2. Ignore the first and (possibly) second interrupt requests and corresponding capture values until two clean results are written to the corresponding TIMER_ICF register. <p>PRS use cases require additional consideration. In general, mask unintended PRS pulses by setting FNSEL = LOGICAL_ZERO in the corresponding PRS_ASYNC_CHx_CTRL register until at least the second associated interrupt is received. At this time, unmask the PRS pulses by setting FNSEL = A and, if necessary, saving relevant capture values or flushing invalid data from the ICF and ICOF registers.</p>
<p>Resolution</p> <p>There is currently no resolution for this issue.</p>

2.10 USART_E301 — Possible Data Transmission on Wrong Edge in Synchronous Mode

Description of Errata
<p>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</p> <ol style="list-style-type: none"> 1. USART_CLKDIV_DIV = 0 (clock = $f_{HPPERCLK} \div 2$), 2. USART_CTRL_CLKPHA = 0, 3. USART_TIMING_CSHOLD = 1 and 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
<p>Reception of each data bit by the secondary is tied to a specific clock edge. Therefore, the late transmission by the main of the first bit of a word may cause the secondary to receive the incorrect data, especially if the data setup time for the secondary approaches or exceeds one half the shift clock period.</p>
Workaround
<p>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:</p> <ul style="list-style-type: none"> • Set USART_CLK_DIV > 0. • Use USART_TIMING_CSHOLD = 0 or USART_TIMING_CSHOLD > 1. • Use USART_CTRL_CLKPHA = 1. This option is particularly useful with SPI flash memories as many support operation in both the CLKPOL = CLKPHA = 0 and CLKPOL = CLKPHA = 1 modes.
Resolution
<p>There is currently no resolution for this issue.</p>

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

Description of Errata
<p>When inter-character spacing is enabled (USART_TIMING_ICS > 0) and USART_CTRL_CLKPHA = 1 in synchronous main 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).</p>
Affected Conditions / Impacts
<p>The extra clock pulse generated at the end of the first frame would cause a secondary 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 secondary would lose synchronization with the main and erroneously receive all frames after the first.</p>
Workaround
<p>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 transmitted using DMA in this case.</p>
Resolution
<p>There is currently no resolution for this issue.</p>

2.12 USART_E304 — PRS Transmit Unavailable in Synchronous Secondary Mode

Description of Errata
When the USART is configured for synchronous secondary operation, the transmit output (MISO) is not driven if the signal is routed to a pin using the PRS producer (e.g., SOURCESEL = 0x20 and SIGSEL = 0x4 for USART0).
Affected Conditions / Impacts
Systems cannot operate the USART in synchronous secondary mode if the PRS is used to route the transmit output to the RX (MISO) pin. Operation is not affected in main mode when the transmit output is routed to the TX (MOSI) pin using the PRS producer nor is operation affected in any mode when the GPIO_USARTn_RXROUTE and GPIO_USARTn_TXROUTE registers are used.
Workaround
There is currently no workaround for this issue.
Resolution
There is currently no resolution for this issue.

2.13 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. 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
There is currently no resolution for this issue.

3. Revision History

Revision 0.6

March, 2026

- Added [TIMER_E302](#).

Revision 0.5

June, 2025

- Added [SE_E302](#).

Revision 0.4

May, 2024

- Updated to module revision 3.
- Added [IADC_E307](#).

Revision 0.3

November, 2022

- Updated errata description and workaround for [HFXO_E301](#).
- Added [USART_E304](#).
- Added [IADC_E306](#).

Revision 0.2

September, 2021

- Added [I2C_E303](#), [USART_E301](#), [USART_E302](#) and [WDOG_E301](#).
- Removed [RADIO_E301](#).

Revision 0.1

September, 2019

- Initial release.

The Leading Innovator in Low-Power Wireless.



IoT Portfolio

silabs.com/products



Simplicity Studio

silabs.com/simplicity



Quality

silabs.com/quality



Support & Community

silabs.com/community

Disclaimer

Silicon Labs provides customers with data sheets, application notes and product briefs (“Product Documentation”) that are intended to present current, accurate and in-depth information about Silicon Labs’ integrated circuits, peripherals, modules and other products (“Products”). Silicon Labs may also provide reference designs, schematics, design files, software, tools, documentation, consultation and other customer support materials (“Support Materials”) that are intended solely to assist customers in developing their own modules, boards or other applications or products that include one or more Silicon Labs Products and may not be used for any other purposes.

Characterization data, device specifications, memory sizes, memory addresses and typical parameters can and do vary in different applications and are provided for illustrative purposes only, are subject to change without notice, and should not be relied upon without independent verification. Product firmware may be updated during the manufacturing process for security or reliability purposes.

Product Documentation does not imply or expressly grant any license to design or fabricate integrated circuits or modules. Customers are solely responsible for the design and distribution of their products including the acquisition of licenses to any necessary intellectual property rights.

Product Documentation and Support Materials are not authorized for use in or in connection with FDA Class III devices or military, automotive or aviation applications and shall in no circumstances be used in weapons of mass destruction or systems capable of delivering such weapons.

SILICON LABS DOES NOT WARRANT THAT PRODUCT DOCUMENTATION OR SUPPORT MATERIALS WILL MEET YOUR REQUIREMENT OR THAT THEIR OPERATION WILL BE UNINTERRUPTED OR THAT ANY SOFTWARE WILL BE ERROR-FREE. PRODUCT DOCUMENTATION AND SUPPORT MATERIALS ARE PROVIDED “AS IS” AND “WITH ALL FAULTS.” SILICON LABS MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, REGARDING THE ACCURACY, COMPLETENESS, RELIABILITY, OR SUITABILITY OF PRODUCT DOCUMENTATION OR SUPPORT MATERIALS, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SILICON LABS DISCLAIMS ALL IMPLIED WARRANTIES.

Trademark Notice

Silicon Laboratories®, Silicon Labs®, Silabs®, and the Silicon Labs S-logo as well as other product or service names used herein are trademarks or registered trademarks of Silicon Laboratories Inc. For more information, please visit: [Silicon Labs Trademark Use Guidelines - Silicon Labs](#). Arm®, Cortex® and Keil® are trademarks or registered trademarks of Arm Holdings or Arm Limited. All other third-party products or brand names mentioned herein are trademarks of their respective holders.