Wireless Gecko™ Bluetooth Module BGM260P Errata

This document contains information on the BGM260P errata. The latest available revision of this device is revision V2.

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 BGM260P.

Table 1.1. Errata Overview

Designator	Title/Problem	Workaround Exists	Exists on Revision	
			V1	V2
DCDC_E304	Leakage Current at High Temperatures Exceeds PFM Mode Maximum Output Current	Yes	X	X
LCD_E301	LOADBUSY Status Goes Inactive Early With Prescaled Clock	Yes	X	X
SE_E302	DPA Countermeasure Unavailable for Some Operations	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_E304	PRS Transmit Unavailable in Synchronous Secondary Mode	No	X	X

2. Current Errata Descriptions

2.1 DCDC_E304 – Leakage Current at High Temperatures Exceeds PFM Mode Maximum Output Current

Description of Errata
By default, the DCDC operates in PFM mode, which supports up to 60 mA of output current. When additional current is required, 120 mA is available by enabling PFMX mode. RAIL typically enables PFMX mode for TX/RX events and then reverts to PFM mode after radio operation is complete. Because digital leakage currents on EFR32xG26 at high temperatures can exceed 60 mA with no MCU or radio activity at all, PFMX mode must be enabled at all times in EM0 or EM1. PFM mode may safely be used in EM2 and EM3.
Affected Conditions / Impacts
At lower load currents, PFMX mode may have reduced efficiency compared to PFM mode. See the efficiency curves in 4.28.3 DC-DC Converter of the EFR32BG26 data sheet.
Workaround
RAIL enables PFMX mode for EM0/1 operation on EFR32xG26 devices and no additional workaround is required. PFMX mode can be enabled before RAIL initialization by setting the `DCDC->CTRL.PFMXEXTREQ` bit.
Resolution
There is currently no resolution for this issue.

2.2 LCD_E301 — LOADBUSY Status Goes Inactive Early With Prescaled Clock

Description of Errata
The LCD_STATUS_LOADBUSY bit erroneously reports completion of writes to the LCD_BACTRL, LCD_AREGA, LCD_AREGB, and LCD_SEGDn registers before synchronization is complete when LCD_CTRL_PRESCALE > 3.
Affected Conditions / Impacts
If LOADBUSY is used to gate consecutive writes to one of the affected registers, only the data associated with the last write is guaranteed to be latched into the register.
Workaround
For each write to one of the affected registers, insert a delay equal to LCD_CTRL_PRESCALE ÷ f_{LCDCLK} after LOADBUSY transitions from 1 to 0 before issuing the next write to the same register. **Note:** LOADBUSY reports when data written from the PCLK register domain into the LCD controller's low-frequency clock domain has been synchronized. It does not indicate when data written into one of the affected registers is actually driven on the LCD controller pins. In cases where writes to these registers, such as LCD_SEGDn, are intended to have the change in pin state be observable on the connected display, LOADBUSY should not be used to gate consecutive writes. Instead, the CPU should issue the register write and wait to issue the next write until a display update event or frame counter update event occurs as reported by the LCD_IF register DISPLAY or FC flag bits. Interrupts associated with these flags can and should be enabled in such cases to minimize energy use by keeping the CPU in a low-energy mode (e.g., EM2) between such consecutive register writes.
Resolution
There is currently no resolution for this issue.

2.3 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.

2.4 TIMER_E302 — Interrupts Do Not Correspond to ICEVCTRL Setting

Description of Errata
Input capture event control (TIMER_CC_CTRL_ICEVCTRL) does not work as expected for all input capture edge settings (TIMER_CC_CTRL_ICEDGE). Specifically, the TIMER requests interrupts on the wrong edges of an incoming signal (such as a PWM waveform) when: • TIMER_CC_CTRL_ICEDGE = BOTH • TIMER_CC_CTRL_ICEVCTRL = RISING or FALLING
Affected Conditions / Impacts
When ICEVCTRL = RISING, instead of being requested on every rising edge: • 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. When ICEVCTRL = FALLING, instead of being requested on every falling edge: • 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 ICOFx 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. 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.
Workaround
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. 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.
Resolution
There is currently no resolution for this issue.

2.5 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 1. USART_CLKDIV_DIV = 0 (clock = $f_{HFPERCLK}$ ÷ 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
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.
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: • 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
There is currently no resolution for this issue.

2.6 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.

3. Revision History

Revision 0.3

March, 2026

- Added TIMER_E302 and USART_E301.

Revision 0.2

June, 2025

- Added SE_E302.

Revision 0.1

February, 2025

- Initial release.

SILICON LABS
CONNECTED INTELLIGENCE

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

SILICON LABS CONNECTED INTELLIGENCE

400 West Cesar Chavez
Austin, TX 78701 | USA

www.silabs.com