



# Wireless Gecko™ EFR32BG26 Errata

---

This document contains information on the EFR32BG26 errata. The latest available revision of this device is revision B.

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 the package marking or electronically.

Errata effective date: March, 2026.

## 1. Errata Summary

The following table lists all the known and unresolved errata for the EFR32BG26.

**Table 1.1. Errata Overview**

Designator	Title/Problem	Workaround Exists	Exists on Revision	
			A	B
DCDC_E304	<a href="#">Leakage Current at High Temperatures Exceeds PFM Mode Maximum Output Current</a>	Yes	X	X
LCD_E301	<a href="#">LOADBUSY Status Goes Inactive Early With Prescaled Clock</a>	Yes	X	X
SE_E302	<a href="#">DPA Countermeasure Unavailable for Some Operations</a>	Yes	X	X
TIMER_E302	<a href="#">Interrupts Do Not Correspond to ICEVCTRL Setting</a>	Yes	X	X
USART_E301	<a href="#">Possible Data Transmission on Wrong Edge in Synchronous Mode</a>	Yes	X	X
USART_E304	<a href="#">PRS Transmit Unavailable in Synchronous Secondary Mode</a>	No	X	X

## 2. Current Errata Descriptions

### 2.1 DCDC\_E304 – Leakage Current at High Temperatures Exceeds PFM Mode Maximum Output Current

<b>Description of Errata</b>
<p>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.</p> <p>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.</p> <p>PFM mode may safely be used in EM2 and EM3.</p>
<b>Affected Conditions / Impacts</b>
<p>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.</p>
<b>Workaround</b>
<p>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-&gt;CTRL.PFMXEXTREQ bit.</p>
<b>Resolution</b>
<p>There is currently no resolution for this issue.</p>

### 2.2 LCD\_E301 — LOADBUSY Status Goes Inactive Early With Prescaled Clock

<b>Description of Errata</b>
<p>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 &gt; 3.</p>
<b>Affected Conditions / Impacts</b>
<p>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.</p>
<b>Workaround</b>
<p>For each write to one of the affected registers, insert a delay equal to <math>\text{LCD\_CTRL\_PRESCALE} \div f_{\text{LCDCLK}}</math> after LOADBUSY transitions from 1 to 0 before issuing the next write to the same register.</p> <p><b>Note:</b> 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.</p> <p>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.</p>
<b>Resolution</b>
<p>There is currently no resolution for this issue.</p>

### 2.3 SE\_E302 – DPA Countermeasure Unavailable for Some Operations

<b>Description of Errata</b>
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).
<b>Affected Conditions / Impacts</b>
<p>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.</p> <p>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.</p>
<b>Workaround</b>
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.
<b>Resolution</b>
There is currently no resolution for this issue.

## 2.4 TIMER\_E302 — Interrupts Do Not Correspond to ICEVCTRL Setting

<p><b>Description of Errata</b></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"> <li>• TIMER_CC_CTRL_ICEDGE = BOTH</li> <li>• TIMER_CC_CTRL_ICEVCTRL = RISING or FALLING</li> </ul>
<p><b>Affected Conditions / Impacts</b></p> <p>When ICEVCTRL = RISING, instead of being requested on every rising edge:</p> <ul style="list-style-type: none"> <li>• The first interrupt is requested on the first rising edge and one capture result is written to TIMER_CCx_ICF.</li> <li>• The second interrupt is requested on the first falling edge and one capture result is written to TIMER_CCx_ICF.</li> <li>• The third interrupt is requested on the second falling edge and two capture results are written to TIMER_CCx_ICF.</li> <li>• Subsequent interrupts are consistently requested on falling edges.</li> </ul> <p>When ICEVCTRL = FALLING, instead of being requested on every falling edge:</p> <ul style="list-style-type: none"> <li>• The first interrupt is requested on the <b>second</b> rising edge, two capture values are written to TIMER_CCx_ICF, a third is written to TIMER_CCx_ICOF, and the ICOF<sub>x</sub> flag corresponding to the capture/compare channel in use is set in the TIMER_IF register.</li> <li>• The second interrupt is requested on the <b>third</b> rising edge and two capture results are written to TIMER_CCx_ICF.</li> <li>• Subsequent interrupts are consistently requested on rising edges.</li> </ul> <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><b>Workaround</b></p> <ol style="list-style-type: none"> <li>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.</li> <li>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.</li> </ol> <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><b>Resolution</b></p> <p>There is currently no resolution for this issue.</p>

## 2.5 USART\_E301 — Possible Data Transmission on Wrong Edge in Synchronous Mode

<b>Description of Errata</b>
<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"> <li>1. USART_CLKDIV_DIV = 0 (clock = <math>f_{HFPERCLK} \div 2</math>),</li> <li>2. USART_CTRL_CLKPHA = 0,</li> <li>3. USART_TIMING_CSHOLD = 1 and</li> <li>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).</li> </ol>
<b>Affected Conditions / Impacts</b>
<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>
<b>Workaround</b>
<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"> <li>• Set USART_CLK_DIV &gt; 0.</li> <li>• Use USART_TIMING_CSHOLD = 0 or USART_TIMING_CSHOLD &gt; 1.</li> <li>• 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.</li> </ul>
<b>Resolution</b>
<p>There is currently no resolution for this issue.</p>

## 2.6 USART\_E304 — PRS Transmit Unavailable in Synchronous Secondary Mode

<b>Description of Errata</b>
<p>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).</p>
<b>Affected Conditions / Impacts</b>
<p>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.</p>
<b>Workaround</b>
<p>There is currently no workaround for this issue.</p>
<b>Resolution</b>
<p>There is currently no resolution for this issue.</p>

### 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

# The Leading Innovator in Low-Power Wireless.



## IoT Portfolio

[silabs.com/products](https://silabs.com/products)



## Simplicity Studio

[silabs.com/simplicity](https://silabs.com/simplicity)



## Quality

[silabs.com/quality](https://silabs.com/quality)



## Support & Community

[silabs.com/community](https://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.