



# Blue Gecko *Bluetooth*® Low Energy SoC

## Blue Gecko SoC (EFR32BG1) Errata

---



This document contains information on the errata of Blue Gecko SoC (EFR32BG1). The latest available revision of this device is revision C. Note that many issues on this device family are resolved in EFR32BG14 devices. The EFR32BG14 devices are very similar to EFR32BG1, and migration will require few changes. More information can be found here: <https://www.silabs.com/products/wireless/bluetooth>.

For errata on older revisions, please refer to the errata history section for the device. The revision information is typically specified in or near the trace code on the device. Refer to the package marking information in the data sheet for more information.

Errata effective date: June, 2018.

## 1. Active Errata Summary

These tables list all known errata for the Blue Gecko SoC (EFR32BG1) and all unresolved errata in revision C of the Blue Gecko SoC (EFR32BG1).

**Table 1.1. Errata History Overview**

Designator	Title/Problem	Exists on Revision:	
		B	C
ADC_E202	Wait After POR or EM4S Wakeup	X	X
ADC_E206	PROGERRIF (Program Error Interrupt Flag) Will Not Clear	X	X
ADC_E207	ADC Scan Repeat Mode with APORT	X	X
ADC_E208	ADC Interrupt Flags	X	X
ADC_E209	ADC and PRS Triggers	X	X
ADC_E210	ADC with PRS and Software Triggers	X	X
ADC_E211	ADC Single Repeat Mode and Tailgating	X	X
ADC_E212	ADC with PRS in ASYNC Mode	X	X
ADC_E213	ADC KEEPINSLOWACC Mode	X	X
ADC_E214	Using ADC CHCONMODE with PRS	X	X
ADC_E215	ADC CHCONMODE Set to MAXRESP Causes Extra Latency	X	X
ADC_E216	ADC Conversion Start Delay	X	X
ADC_E217	Multiple CLK Mode Switches	X	X
ADC_E218	SINGLEACT and SCANACT Status Flags Delayed	X	X
ADC_E219	STOP Command Causing FIFO Corruption	X	X
ADC_E220	AUXHFRCO in ASYNC mode with ASYNC CLK in ASNEEDED mode	X	X
ADC_E221	ADC Temperature Sensor Must be Used in LOWACC Mode	X	X
ADC_E222	ADC EM2 Wakeup on a Comparator Match Disables EM2 Entry	X	X
ADC_E223	Delayed ADC Conversion or Warmup Start	X	X
ADC_E224	ADC Warm-Up Ready Can Cause ACMP to Not Function	X	X
ADC_E226	SCANSTOP Does Not Immediately Stop the Ongoing Sample	X	X
ADC_E227	New Conversion Triggers Cause Jitter to the Ongoing Conversions	X	X
CORE_E201	SYSTICK and an External Clock	X	X
CRYPTO_E201	Full CRYPTO Not Available in All Value Devices	—	X
CUR_E201	EM2 and EM3 Current Consumption	X	X
CUR_E202	EM2/3 Current Consumption at Cold Temperatures	X	—
DBG_E201	AUXHFRCO Debug Limitations	X	X
DBG_E202	Debug Access to ADC and LEUART not Functioning as Intended	X	X
DBG_E204	Debug Recovery with JTAG Does Not Work	X	X
DCDC_E201	DCDC Stops Regulating During a Fast EM0/1 to EM2/3/4H Transition	X	—
DCDC_E202	Regulated DCDC Output Can Dip on EM2 Entry	X	X

Designator	Title/Problem	Exists on Revision:	
		B	C
DCDC_E203	Regulated DCDC Output Can Dip on EM2 Entry if not in LN Mode	X	X
DCDC_E205	Delay Required after Enabling the DC-DC Before Entering EM2/3/4H	X	X
DCDC_E206	Reset During Radio Operation With DC-DC Results in DVDD Brown Out	X	X
EFR_E201	Bit Access Not Supported for Low Energy Peripherals	X	X
EFR_E202	Read-Clear Access for LETIMER0 and RTCC Interrupts	X	X
EMU_E201	High Temperature Operation	X	X
EMU_E204	Restrictions Writing TEMPHIGH and TEMPLOW	X	X
EMU_E205	Restrictions Reading TEMP	X	X
EMU_E207	GPIO State can be Lost During EM4 Recovery	X	X
EMU_E208	Occasional Full Reset After Exiting EM4H	X	X
EMU_E209	Potential EM2 Lock-up when using IDAC or the Debugger with the LDMA	X	X
EMU_E210	Potential Power-Down When Entering EM2	X	X
EMU_E215	Device May Brown Out After Energy Mode Transition	X	X
EMU_E216	EM4H I/O Retention Cannot Be Disabled	X	X
FLASH_E201	Potential Program Failure after Power On	X	X
GPIO_E201	GPIO Default Slew Rate	X	—
I2C_E201	I2C ABORT Command	X	X
I2C_E206	Slave Holds SCL Low After Losing Arbitration	X	X
IDAC_E201	IDAC CURSTABLE Bit Not Reliable	X	X
LEUART_E201	Restrictions Setting TXDMAWU/RXDMAWU of LEUART <sub>n</sub> _CTRL	X	X
RADIO_E202	802.15.4 Channel 14	—	X
RADIO_E203	Bluetooth Smart Channel 26	—	X
RADIO_E204	Increased EVM on Selected Channels	—	X
RADIO_E206	Bluetooth Smart Receive Sensitivity	X	X
RADIO_E207	Sensitivity at 2.42 GHz	—	X
RADIO_E208	Receive Sensitivity	X	X
RMU_E201	CTRL Register Reset on All Resets	X	X
RMU_E202	External Debug Access Not Available After Watchdog or Lockup Full Reset	X	X
RTCC_E201	RTCC Does Not Support Compare/Capture Wrap with Prescaler	X	X
RTCC_E202	RTCC Triggers to LETIMER Not Safe	X	X
RTCC_E203	Potential Stability Issue with RTCC Registers	X	X
TIMER_E201	Timer in Input Capture Mode Can Stop Counting	X	X
USART_E202	Incorrect 8-bit Timer Operation in Asynchronous Mode	X	X

Table 1.2. Active Errata Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	ADC_E202	Wait After POR or EM4S Wakeup	Yes	C	Fixed in EFR32BG14
2	ADC_E206	PROGERRIF (Program Error Interrupt Flag) Will Not Clear	Yes	C	Fixed in EFR32BG14
3	ADC_E207	ADC Scan Repeat Mode with APORT	No	C	Fixed in EFR32BG14
4	ADC_E208	ADC Interrupt Flags	Yes	C	Fixed in EFR32BG14
5	ADC_E209	ADC and PRS Triggers	Yes	C	Fixed in EFR32BG14
6	ADC_E210	ADC with PRS and Software Triggers	Yes	C	Fixed in EFR32BG14
7	ADC_E211	ADC Single Repeat Mode and Tailgating	No	C	Fixed in EFR32BG14
8	ADC_E212	ADC with PRS in ASYNC Mode	Yes	C	Fixed in EFR32BG14
9	ADC_E213	ADC KEEPINSLOWACC Mode	No	C	—
10	ADC_E214	Using ADC CHCONMODE with PRS	Yes	C	Fixed in EFR32BG14
11	ADC_E215	ADC CHCONMODE Set to MAXRESP Causes Extra Latency	Yes	C	Fixed in EFR32BG14
12	ADC_E216	ADC Conversion Start Delay	Yes	C	Fixed in EFR32BG14
13	ADC_E217	Multiple CLK Mode Switches	Yes	C	Fixed in EFR32BG14
14	ADC_E218	SINGLEACT and SCANACT Status Flags Delayed	Yes	C	Fixed in EFR32BG14
15	ADC_E219	STOP Command Causing FIFO Corruption	Yes	C	Fixed in EFR32BG14
16	ADC_E220	AUXHFRCO in ASYNC mode with ASYNC CLK in ASNEEDED mode	Yes	C	Fixed in EFR32BG14
17	ADC_E221	ADC Temperature Sensor Must be Used in LOWACC Mode	Yes	C	Fixed in EFR32BG14
18	ADC_E222	ADC EM2 Wakeup on a Comparator Match Disables EM2 Entry	Yes	C	Fixed in EFR32BG14
19	ADC_E223	Delayed ADC Conversion or Warmup Start	Yes	C	Fixed in EFR32BG14
20	ADC_E224	ADC Warm-Up Ready Can Cause ACMP to Not Function	Yes	C	Fixed in EFR32BG14
21	ADC_E226	SCANSTOP Does Not Immediately Stop the Ongoing Sample	No	C	Fixed in EFR32BG14
22	ADC_E227	New Conversion Triggers Cause Jitter to the Ongoing Conversions	No	C	Fixed in EFR32BG14

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
23	CORE_E201	SYSTICK and an External Clock	Yes	C	Fixed in EFR32BG14
24	CRYPTO_E201	Full CRYPTO Not Available in All Value Devices	No	C	C, date code 1636 (September 5, 2016)
25	DBG_E201	AUXHFRCO Debug Limitations	Yes	C	Fixed in EFR32BG14
26	DBG_E202	Debug Access to ADC and LEUART not Functioning as Intended	No	C	Fixed in EFR32BG14
27	DBG_E204	Debug Recovery with JTAG Does Not Work	Yes	C	—
28	DCDC_E202	Regulated DCDC Output Can Dip on EM2 Entry	No	C	Fixed in EFR32BG14
29	DCDC_E203	Regulated DCDC Output Can Dip on EM2 Entry if not in LN Mode	Yes	C	Fixed in EFR32BG14
30	DCDC_E205	Delay Required after Enabling the DC-DC Before Entering EM2/3/4H	Yes	C	Fixed in EFR32BG14
31	DCDC_E206	Reset During Radio Operation With DC-DC Results in DVDD Brown Out	Yes	C	Fixed in EFR32BG14
32	EFR_E201	Bit Access Not Supported for Low Energy Peripherals	Yes	C	Fixed in EFR32BG14
33	EFR_E202	Read-Clear Access for LETIMER0 and RTCC Interrupts	Yes	C	Fixed in EFR32BG14
34	EMU_E201	High Temperature Operation	Yes	C	Fixed in EFR32BG14
35	EMU_E204	Restrictions Writing TEMPHIGH and TEMPLOW	Yes	C	Fixed in EFR32BG14
36	EMU_E205	Restrictions Reading TEMP	Yes	C	Fixed in EFR32BG14
37	EMU_E207	GPIO State can be Lost During EM4 Recovery	Yes	C	Fixed in EFR32BG14
38	EMU_E208	Occasional Full Reset After Exiting EM4H	Yes	C	Fixed in EFR32BG14
39	EMU_E209	Potential EM2 Lock-up when using IDAC or the Debugger with the LDMA	Yes	C	Fixed in EFR32BG14
40	EMU_E210	Potential Power-Down When Entering EM2	Yes	C	Fixed in EFR32BG14
41	EMU_E215	Device May Brown Out After Energy Mode Transition	Yes	C	Fixed in EFR32BG14
42	EMU_E216	EM4H I/O Retention Cannot Be Disabled	Yes	C	Fixed in EFR32BG14
43	IDAC_E201	IDAC CURSTABLE Bit Not Reliable	Yes	C	Fixed in EFR32BG14
44	I2C_E201	I2C ABORT Command	Yes	C	Fixed in EFR32BG14
45	I2C_E206	Slave Holds SCL Low After Losing Arbitration	Yes	C	—

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
46	LEUART_E201	Restrictions Setting TXDMAWU/RXDMAWU of LEUARTn_CTRL	Yes	C	Fixed in EFR32BG14
47	RTCC_E201	RTCC Does Not Support Compare/Capture Wrap with Prescaler	Yes	C	Fixed in EFR32BG14
48	RTCC_E202	RTCC Triggers to LETIMER Not Safe	Yes	C	Fixed in EFR32BG14
49	RTCC_E203	Potential Stability Issue with RTCC Registers	Yes	C	Fixed in EFR32BG14
50	TIMER_E201	Timer in Input Capture Mode Can Stop Counting	Yes	C	Fixed in EFR32BG14
51	FLASH_E201	Potential Program Failure after Power On	Yes	C	Fixed in EFR32BG14
52	RMU_E201	CTRL Register Reset on All Resets	Yes	C	Fixed in EFR32BG14
53	RMU_E202	External Debug Access Not Available After Watchdog or Lockup Full Reset	Yes	C	—
54	USART_E202	Incorrect 8-bit Timer Operation in Asynchronous Mode	No	C	Fixed in EFR32BG14

## 2. Detailed Errata Descriptions

### 2.1 ADC\_E202 – Wait After POR or EM4S Wakeup

<b>Description of Errata</b>
Attempting to take an ADC sample too soon after POR or EM4S wakeup can result in an erroneous sample being returned by the ADC.
<b>Affected Conditions / Impacts</b>
ADC can return erroneous sample data.
<b>Workaround</b>
After POR or EM4S wakeup, users must wait 150 $\mu$ s before starting and using the ADC.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

### 2.2 ADC\_E206 – PROGERRIF (Program Error Interrupt Flag) Will Not Clear

<b>Description of Errata</b>
If an invalid selection is made on APORT via POSSEL or NEGSEL by selecting both POSSEL and NEGSEL on the X bus or both on the Y bus, then PROGERRIF will set to 1. If this is followed by a valid internal selection (POSSEL set to AVDD and NEGSEL set to VSS), the PROGERRIF flag will remain 1, even though the selection is valid. This PROGERRIF flag can only be cleared by first making a valid selection of the APORT channels, then moving to an internal selection.
<b>Affected Conditions / Impacts</b>
If firmware attempts to clear the PROGERRIF error condition by selecting a valid non-APORT channel, the PROGERRIF bit will remain set to 1 even after firmware attempts to clear the flag.
<b>Workaround</b>
Firmware can clear the flag by first making a valid selection on POSSEL/NEGSEL to an APORT channel before moving to an internal selection.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

### 2.3 ADC\_E207 – ADC Scan Repeat Mode with APORT

<b>Description of Errata</b>
If Scan repeat mode is enabled, then the ADC sets the correct APORT settings only on the first scan sequence conversion. APORT settings are cleared for all subsequent scan sequence conversions.
<b>Affected Conditions / Impacts</b>
Scan repeat mode does not work with the APORT.
<b>Workaround</b>
There is no known workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

## 2.4 ADC\_E208 – ADC Interrupt Flags

<b>Description of Errata</b>
The SCANCMP and SINGLECMP interrupt flags in ADCn_IF are not clearable in certain scenarios. These interrupts can trigger multiple times on the same event if the event conditions are not cleared before clearing the interrupt flags.
<b>Affected Conditions / Impacts</b>
Multiple SCANCMP or SINGLECMP interrupts can occur from the same source.
<b>Workaround</b>
Once an interrupt is received, either disable the compare logic (clear CMPEN in ADCn_SINGLECTRL or ADCn_SCANCTRL) or clear the FIFO (using ADCn_SINGLEFIFOCLEAR or ADCn_SCANFIFOCLEAR) before clearing the interrupt flags.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

## 2.5 ADC\_E209 – ADC and PRS Triggers

<b>Description of Errata</b>
Scan conversions may become dependent on Single triggers, and vice versa. If both Scan and Single channel mode are set to PRS Timed mode and both Scan and Single triggers are high at the same time before the approximation phase has started, the conversion is halted until both PRS triggers occur.
<b>Affected Conditions / Impacts</b>
A Scan conversion can end up waiting for a Single PRS trigger to go low before it starts the approximation phase, even if the Scan PRS trigger has already gone low, and vice versa.
<b>Workaround</b>
Do not set both ADC Single and Scan to use PRS Timed mode at the same time. Alternatively, if they are both set to use PRS Timed mode simultaneously, ensure that both PRS timed pulses are never high at the same time.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.



## 2.6 ADC\_E210 – ADC with PRS and Software Triggers

Description of Errata
<p>Software triggered conversions are affected if the ADC is set to use PRS Timed mode, even when PRSEN is 0. If PRS Timed mode is selected using PRSMODE in ADCn_SCANCTRLX, regardless of what is set in PRSEN:</p> <ol style="list-style-type: none"> <li>1. All channels in the scan sequence (except the first one) experience an additional 2 raw ADC_CLK cycle latency in the conversion. The same 2 cycle latency will be experienced in all subsequent conversions in repeat mode and conversions done due to oversampling in both single and scan mode.</li> <li>2. If the software triggers a conversion and a PRS pulse comes in before the conversion has passed the acquisition phase, the software-triggered conversion will stall and wait for the PRS pulse to go low before starting the approximation phase.</li> </ol>
Affected Conditions / Impacts
<p>When using the PRS Timed mode with ADC scan, the overall scan conversion time will be longer than expected by:</p> <p>Extra ADC_CLK Cycles = <math>2 \times (\text{number of channels scanned} - 1)</math></p> <p>In addition, 1 MSPS sampling will not be reachable with a software trigger, and software triggers may experience a dependency on the PRS triggers.</p>
Workaround
<p>There is currently no workaround for extra clocks generated by PRS Timed mode. When doing software triggered conversions, do not select PRS Timed mode in PRSMODE.</p>
Resolution
<p>There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.</p>

## 2.7 ADC\_E211 – ADC Single Repeat Mode and Tailgating

Description of Errata
<p>If the ADC Single repetitive mode is enabled using REP in ADCn_SINGLECTRL and tailgating is enabled by setting TAILGATE in ADCn_CTRL, then the ADC waits for first Scan conversion. Once that completes, a Single conversion starts, but it is stopped by the ADC before it is complete.</p>
Affected Conditions / Impacts
<p>Single repeat mode does not work with tailgating enabled.</p>
Workaround
<p>There is currently no workaround for this issue.</p>
Resolution
<p>There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.</p>

## 2.8 ADC\_E212 – ADC with PRS in ASYNC Mode

Description of Errata
<p>The ADC is currently documented as an asynchronous PRS producer. However, when the ADC is setup in ASYNC mode (ADCCLKMODE in ADCn_CTRL is set to ASYNC), the PRS outputs are no longer synchronous to the HFPERCLK.</p>
Affected Conditions / Impacts
<p>The ADC PRS outputs will not be properly synchronized when the ADC is in ASYNC mode.</p>
Workaround
<p>Use the ADC PRS outputs only when the ADC is setup to use SYNC mode (ADCCLKMODE in ADCn_CTRL is set to SYNC).</p>
Resolution
<p>There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.</p>

**2.9 ADC\_E213 – ADC KEEPINSLOWACC Mode**

<b>Description of Errata</b>
When WARMUP-MODE in ADCn_CTRL is set to KEEPINSLOWACC, the ADC does not track the input voltage. Also, the ADC keeps the input muxes closed even during channel switching, making it not recommended to operate the ADC in KEEPINSLOWACC mode.
<b>Affected Conditions / Impacts</b>
KEEPINSLOWACC warmup mode does not function properly.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue.

**2.10 ADC\_E214 – Using ADC CHCONMODE with PRS**

<b>Description of Errata</b>
When CHCONMODE in ADCn_CTRL is set MAXRESP, the ADC does not work with PRS Timed mode.
<b>Affected Conditions / Impacts</b>
If the PRS pulse is longer than the ADC acquisition time, the input mux select lines are switched during the acquisition phase, causing the results to no longer be usable.
<b>Workaround</b>
PRS Timed mode should not be used with CHCONMODE in ADCn_CTRL set to MAXRESP.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.11 ADC\_E215 – ADC CHCONMODE Set to MAXRESP Causes Extra Latency**

<b>Description of Errata</b>
Setting CHCONMODE in ADCn_CTRL to MAXRESP introduces 7 extra raw ADC_CLK cycles of latency between scan conversions in a sequence.
<b>Affected Conditions / Impacts</b>
The 7 raw ADC_CLK cycles of extra latency impacts the time the sample is taken.
<b>Workaround</b>
Use CHCONMODE set to MAXSETTLE in order to avoid the extra latency.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.12 ADC\_E216 – ADC Conversion Start Delay**

<b>Description of Errata</b>
If CONVSTARTDELAYEN in both SINGLECTRLX and SCANCTRLX registers are set to 1, the ADC will look at CONVSTARTDELAY value set in SINGLECTRLX register regardless of the conversion type (SCAN or SINGLE). If a SCAN conversion is triggered in this scenario, the conversion will be delayed for the value specified in SINGLECTRLX. In all other cases, the ADC behaves as expected.
<b>Affected Conditions / Impacts</b>
Enabling both SINGLE and SCAN conversion start delay can result in unexpected behavior if the delay selection in the SINGLE and SCAN registers is different.
<b>Workaround</b>
If different CONVSTARTDELAY values are desired for SCAN and SINGLE, do not keep both SINGLE CONVSTARTDELAY and SCAN CONVSTARTDELAY enabled at the same time.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.13 ADC\_E217 – Multiple CLK Mode Switches**

<b>Description of Errata</b>
This issue can be encountered if the ADC clock (CLK) mode switches between asynchronous (ASYNC) and synchronous (SYNC) while data is being read. Specifically, this issue can occur if ADC operates in ASYNC mode with converted data being read, then some conversions are done in SYNC mode before being switched back to ASYNC mode again. FIFOCOUNT may show the wrong value when read in ASYNC mode after the last clock mode switch. Note that the recommended procedure for switching CLK modes should always be followed.
<b>Affected Conditions / Impacts</b>
An unexpected value may be read from FIFOCOUNT registers.
<b>Workaround</b>
When switching from SYNC to ASYNC mode more than once, clear the FIFOs using FIFOCLEAR before and after the CLK mode switch.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.14 ADC\_E218 – SINGLEACT and SCANACT Status Flags Delayed**

<b>Description of Errata</b>
Once the SINGELSTART/SCANSTART commands are issued, it takes a few cycles before the ADC SINGLEACT/SCANACT status flags are set.
<b>Affected Conditions / Impacts</b>
The status flags cannot be checked right after the conversion start commands are issued.
<b>Workaround</b>
Firmware that wants to check when a conversion has started after sending a software trigger will need to wait until the corresponding status flag (SINGELACT/SCANACT) goes high before proceeding.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.15 ADC\_E219 – STOP Command Causing FIFO Corruption**

<b>Description of Errata</b>
If a single or scan conversion is running and a software STOP command is issued, the conversion should stop immediately and the result should be discarded (no FIFO updates should happen). Currently in the ADC, if a single (scan) conversion is stopped by a software STOP command and there is a scan (single) conversion pending, then the conversion will incorrectly continue and the result will be used to update the FIFO.
<b>Affected Conditions / Impacts</b>
Issuing the STOP command can have two different effects.  If the command is sent during a conversion, the effect will be immediate if no other conversion is pending. The immediate effect means stopping the on-going conversion and discarding the current sample.  If the command is sent during a conversion, the on-going conversion will finish, and the current sample will be saved into the corresponding FIFO. This means that the single conversion will fully finish (regardless of the ADC mode, e.g. if in the oversampling mode, the full oversampling will be executed, or if in the repetition mode, the current conversion will finish and then the repetition mode for single will be disabled) and that the scan conversion will fully finish conversion of the current channel, but it will not finish the whole scan sequence (regardless of the ADC mode). In the last case, the scan FIFO will be updated with the data from channels converted so far.
<b>Workaround</b>
If using the STOP command, the effect will be immediate if no scan triggers were pending during the single conversion and vice versa.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.16 ADC\_E220 – AUXHFRCO in ASYNC mode with ASYNC CLK in ASNEEDED mode**

<b>Description of Errata</b>
ADC sampling in the ASYNC ASNEEDED mode when running from the AUXHFRCO can temporarily upset voltage references used in certain analog components. This issue effects the BOD trip point, DCDC voltage accuracy, ACMP reference accuracy, IDAC output current accuracy, and low voltage digital supply voltage accuracy.
<b>Affected Conditions / Impacts</b>
If the ADC is being used in ASYNC mode with AUXHFRCO, ASYNC CLK cannot be used in ASNEEDED mode.
<b>Workaround</b>
If the ADC is being used in ASYNC mode with AUXHFRCO, ASYNC CLK must be set to ALWAYS ON mode.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.17 ADC\_E221 – ADC Temperature Sensor Must be Used in LOWACC Mode**

<b>Description of Errata</b>
The ADC temperature sensor used in HIGHACC mode can temporarily upset voltage references used in certain analog components. This issue affects the BOD trip point, DCDC voltage accuracy, ACMP reference accuracy, IDAC output current accuracy, and low voltage digital supply voltage accuracy.
<b>Affected Conditions / Impacts</b>
If the ADC is being used to take a reading from its internal temperature sensor, ADCn_BIASPROG.GPBIASACC = 0 cannot be used.
<b>Workaround</b>
Use ADCn_BIASPROG.GPBIASACC = 1 when taking temperature measurements.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.18 ADC\_E222 – ADC EM2 Wakeup on a Comparator Match Disables EM2 Entry**

<b>Description of Errata</b>
If the ADC wakes up the system from EM2 on a comparator flag match (CMPEN must be set in SINGLECTRL/SCANCTRL), the wake-up handler will not be able to clear this EM2 wakeup request. This results in the core immediately exiting EM2 on subsequent EM2 entry.
<b>Affected Conditions / Impacts</b>
Systems using the ADC comparator flag match may not be able to enter EM2.
<b>Workaround</b>
To clear the wakeup request, the wakeup handler must do one of the following: <ul style="list-style-type: none"> <li>• Disable CMPEN in the SINGLECTRL/SCANCTRL register.</li> <li>• Reset the ADC FIFO.</li> <li>• Continue performing conversions until an incoming conversion does not pass the CMP threshold set in CMPTHR.</li> </ul> Once one of these conditions has been met, the comparator can be re-enabled (if it was disabled) and the core can enter EM2.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.19 ADC\_E223 – Delayed ADC Conversion or Warmup Start**

<b>Description of Errata</b>
<p>When a new conversion trigger is received from PRS or a software start, the ADC is expected to either:</p> <ol style="list-style-type: none"> <li>1. Immediately start warmup (if ADCn_CTRL_WARMUPMODE is set to NORMAL, KEEPINSTANDBY, or KEEPINSLOWACC).</li> <li>2. Immediately start the conversion (if in KEEPADCWARM warmup mode).</li> </ol> <p>This expected behavior does not occur if the ADC prescaler (ADCn_CTRL_PRESC) is set to a non-zero value, as the start of the ADC warmup or conversion gets delayed by the number of ADC clock cycles specified by the PRESC field.</p>
<b>Affected Conditions / Impacts</b>
Systems using the ADC clock prescaler will see a delay after a start-of-conversion or start-of-warmup trigger and the actual conversion or warmup sequence.
<b>Workaround</b>
For systems that cannot tolerate a delay between the start-of-conversion and actual conversion or before ADC warmup, set ADCn_CTRL_PRESC to 0 and use the prescalars in the CMU to prescale the incoming ADC_CLK.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.20 ADC\_E224 – ADC Warm-Up Ready Can Cause ACMP to Not Function**

<b>Description of Errata</b>
<p>The ACMP module uses the warm up timing module in the ADC to determine when the peripherals are ready for use. However, if the ADC is enabled first, this timing module can fail to properly handshake with a low probability, causing the ACMP module to never finish warming up. The ADC is not affected by this issue and will always be available after it is enabled.</p>
<b>Affected Conditions / Impacts</b>
Systems using the ACMP module in conjunction with the ADC can see intermittent failures where these modules do not operate.
<b>Workaround</b>
To work around this issue, enable the ACMP module before enabling the ADC. This will ensure the handshaking logic between the ADC and other modules functions correctly.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.21 ADC\_E226 – SCANSTOP Does Not Immediately Stop the Ongoing Sample**

<b>Description of Errata</b>
<p>The stop behavior of conversions differs between scan mode (SCANSTOP in ADCn_CMD) or single conversion mode (SINGLESTOP in ADCn_CMD).</p> <ol style="list-style-type: none"> <li>1. If the SINGLESTOP command is sent during a single conversion, the effect will be immediate if no other conversion is pending. In this case, the ADC immediately stops the on-going conversion and discards the current sample.</li> <li>2. If the SCANSTOP command is sent during a scan conversion, the on-going conversion will finish and the current sample will be saved into the corresponding FIFO. If oversampling mode is enabled (RES in ADCn_SCANCTRL set to OVS), the full oversampling will be completed for this sample. If in repetition mode (REP in ADCn_SCANCTRL set to 1), the current conversion will finish and then the repetition mode for the single channel will be disabled. Once the current conversion completes, the rest of the scan sequence aborts.</li> </ol>
<b>Affected Conditions / Impacts</b>
Systems issuing a SCANSTOP command to the ADC may see an additional conversion in scan mode.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.22 ADC\_E227 – New Conversion Triggers Cause Jitter to the Ongoing Conversions**

<b>Description of Errata</b>
In some cases, an ADC conversion can take one conversion clock longer in the worst case if a new conversion trigger occurs while another conversion is ongoing. The results of the conversion are unaffected.
<b>Affected Conditions / Impacts</b>
Systems issuing conversion triggers while conversions are ongoing may see slightly longer conversion times.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.23 CORE\_E201 – SYSTICK and an External Clock**

<b>Description of Errata</b>
The core allows two different clock sources for the SysTick counter. The first one is the core free-running clock, which operates correctly. The second source uses the 32 kHz from the RTCC. This pulse width of this clock is not wide enough, which results in missed SysTick counts.
<b>Affected Conditions / Impacts</b>
Firmware should not use the external clock source for the SysTick counter.
<b>Workaround</b>
Use the core free-running clock for the SysTick, which is the default selection.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.24 CRYPTO\_E201 — Full CRYPTO Not Available in All Value Devices**

<b>Description of Errata</b>
The device documentation states that all devices support full CRYPTO capability. However, some Value devices (e.g. EFR32xG1V) before a date code of 1636 (September 5, 2016) will not have full CRYPTO capability. Instead, these affected devices will only have AES enabled.
<b>Affected Conditions / Impacts</b>
Some Value devices before a date code of 1636 (September 5, 2016) will not have full CRYPTO capability.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
Revision C Value devices after date code 1636 (September 5, 2016) will have full CRYPTO capability.

**2.25 DBG\_E201 – AUXHFRCO Debug Limitations**

<b>Description of Errata</b>
The AUXHFRCO is the default debug clock, set by DBG in CMU_DBGCLKSEL. Using AUXHFRCO as the debug clock while entering EM2 has the potential of corrupting the system, causing some registers in the TPIU to not retain their value.
<b>Affected Conditions / Impacts</b>
Firmware should not use the AUXHFRCO as the debug clock while entering EM2.
<b>Workaround</b>
When using AUXHFRCO as the debug clock, it must be stopped before entering the EM2 power mode. Alternatively, select another clock source as the debug clock before entering EM2.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.26 DBG\_E202 – Debug Access to ADC and LEUART not Functioning as Intended**

<b>Description of Errata</b>
ADC and LEUART registers that have a side-effect during a read access (i.e., LEUART_RXDATA or other registers that pop data read) continue to execute triggered actions when an attached debugger is performing read accesses. The intended behavior is to halt execution of pop actions when a debugger is attached.
<b>Affected Conditions / Impacts</b>
Some ADC and LEAURT registers will pop data from their respective buffers on read accesses from the debugger.
<b>Workaround</b>
The user needs to be aware that while debugging, reads via the debugger have the same affect as reads by the CPU during normal mode for these registers. To avoid the debugger triggering a data pop from the buffer, the user should avoid setting a memory window over the actionable register(s), as the reads to fetch the data for the debugger memory view would trigger the data pop from the buffer.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.



**2.27 DBG\_E204 – Debug Recovery with JTAG Does Not Work**

<b>Description of Errata</b>
The debug recovery algorithm of holding down pin reset, issuing a System Bus Stall AAP instruction, and releasing the reset pin does not work when using the JTAG debug interface. When using the JTAG debug interface, the core will continue to execute code as soon as the reset pin is released.
<b>Affected Conditions / Impacts</b>
The debug recovery sequence will not work when using the JTAG debug interface.
<b>Workaround</b>
Use the Serial Wire debug interface to implement the debug recovery sequence.
<b>Resolution</b>
There is currently no resolution for this issue.

**2.28 DCDC\_E202 – Regulated DCDC Output Can Dip on EM2 Entry**

<b>Description of Errata</b>
The regulated output on DVDD, when using the DCDC, can dip up to 4% during EM2 entry.
<b>Affected Conditions / Impacts</b>
External components operating from DVDD, when regulated by the DCDC, may suffer a depressed supply by up to 4%, which can last approximately 1 ms. Operation of the device is not affected when this occurs.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.29 DCDC\_E203 – Regulated DCDC Output Can Dip on EM2 Entry if not in LN Mode**

<b>Description of Errata</b>
The regulated output on DVDD, when using the DCDC, can dip up to approximately 9% during EM2 entry if the DCDC has not completed a transition into LN mode. Note that if a switch to LN mode completes prior to entry to EM2, the DCDC can exhibit the behavior described in <a href="#">DCDC_E202</a> .
<b>Affected Conditions / Impacts</b>
External components operating from DVDD, when regulated by the DCDC, may suffer a depressed supply by up to approximately 9%, which can last approximately 1 ms. A BOD reset of the device is possible, but unlikely.
<b>Workaround</b>
Firmware should wait to see LNRUNNING set after initiating a transition of the DCDC into LN mode, before attempting to enter EM2. This workaround is included in v5.0.0 or later of the Gecko SDK.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.30 DCDC\_E205 – Delay Required after Enabling the DC-DC Before Entering EM2/3/4H**

<b>Description of Errata</b>
The DC-DC hardware state machine may malfunction if the device transitions to EM2, EM3, or EM4H before the DC-DC starts running Low Noise mode after being enabled.
<b>Affected Conditions / Impacts</b>
Systems using the DC-DC converter should wait ~300 µs until the LNRUNNING status bit in the DCDCSTATUS register is set before transitioning to EM2, EM3, or EM4H after enabling the DC-DC converter.
<b>Workaround</b>
Firmware should wait for the DCDCLNRRUNNING bit to be set in the EMU_IF register before transitioning to EM2, EM3, or EM4H after enabling the DC-DC converter.  This workaround is included in v5.1.0 or later of the Gecko SDK.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.31 DCDC\_E206 – Reset During Radio Operation With DC-DC Results in DVDD Brown Out**

<b>Description of Errata</b>
<p>During radio operation with the DC-DC enabled, the protocol stacks enable radio interference minimization features which change the source of the DC-DC clock to reduce the impact of the DC-DC switching frequency on the radio. When a soft reset occurs (e.g., from a Pin or WDOG), the minimal interference DC-DC clock stops, causing the DC-DC to stop switching and charging up the output voltage, which results in a brown-out on the DVDD supply. The device then resets to its default start-up DC-DC configuration with the bypass switch enabled (i.e., DVDD=VREGVDD), and execution continues.</p> <p>As a result of this sequence, the original reset cause indicated in the RMU_RSTCAUSE register will be masked by the brown-out reset (DVddbOD) flag.</p>
<b>Affected Conditions / Impacts</b>
<p>Systems using the radio and DC-DC converter that are expecting to read valid results from the RMU_RSTCAUSE register for EM4RST, WDOGRST, SYSREQRST, LOCKUPRST, or EXTRST resets will see a brown-out reset event, instead.</p>
<b>Workaround</b>
<p>To prevent the DVDD brown-out, firmware can immediately enable the bypass switch early in code execution. For example, in the <code>SystemInit()</code> function in <code>system_efr32xglb.c</code>, add the following lines of code:</p> <pre>BUS_RegBitWrite(&amp;EMU-&gt;DCDCCLIMCTRL, _EMU_DCDCCLIMCTRL_BYPLIMEN_SHIFT, 1); EMU-&gt;DCDCCTRL = (EMU-&gt;DCDCCTRL &amp; ~_EMU_DCDCCTRL_DCDCMODE_MASK)   emuDcdcMode_Bypass; *(volatile uint32_t *) (0x400E3074) &amp;= ~(0x1UL &lt;&lt; 0); *(volatile uint32_t *) (0x400E3060) &amp;= ~(0x1UL &lt;&lt; 28);</pre> <p>Then, add the following line at the beginning of <code>main()</code>:</p> <pre>BUS_RegBitWrite(&amp;EMU-&gt;DCDCCLIMCTRL, _EMU_DCDCCLIMCTRL_BYPLIMEN_SHIFT, 0);</pre> <p>This will disable the bypass current limit, which was enabled in the first line of the <code>SystemInit()</code> sequence. This limit prevents excessive current through the bypass when transitioning across large steps between DVDD and VDD, but consumes ~10uA of current that is no longer necessary once DVDD has ascended to near VDD.</p> <p>Additional information on the workaround and examples provided is available from the following KB article URL:</p> <p><a href="http://community.silabs.com/t5/Mesh-Knowledge-Base/DCDC-E206-Reset-During-Radio-Operation-With-DC-DC-Results-in/ta-p/193802">http://community.silabs.com/t5/Mesh-Knowledge-Base/DCDC-E206-Reset-During-Radio-Operation-With-DC-DC-Results-in/ta-p/193802</a></p> <p><b>Note:</b> This workaround may not suffice for a pin reset initiated by a button press, which may be on the order of 50-300 ms. Because the device won't come out of reset to execute the workaround until the reset button is released, the DVDD supply will discharge during the entire duration the button is pressed, which will likely result in a DVDD brown-out.</p>
<b>Resolution</b>
<p>There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.</p>

**2.32 EFR\_E201 – Bit Access Not Supported for Low Energy Peripherals**

<b>Description of Errata</b>
Bit set and clear operations do not work properly for Low Energy Peripherals including WDOG, PCNT0, LEUART0, LETIMER0, and RTCC.
<b>Affected Conditions / Impacts</b>
To implement bit set or bit clear operations with Low Energy Peripherals, firmware must execute a read-modify-write operation address on the peripheral's registers.
<b>Workaround</b>
To implement bit set or bit clear operations with Low Energy Peripherals (WDOG, PCNT0, LEUART0, LETIMER0, and RTCC), firmware must execute a read-modify-write operation address on the peripheral's registers.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.33 EFR\_E202 – Read-Clear Access for LETIMER0 and RTCC Interrupts**

<b>Description of Errata</b>
The automatic read-clear mechanism for the LETIMER0 and RTCC modules does not actually clear the module interrupts.
<b>Affected Conditions / Impacts</b>
Firmware must be written to manually clear the interrupts for the LETIMER0 and RTCC modules.
<b>Workaround</b>
Firmware must read the LETIMER0 and RTCC interrupts using the module IFS register and clear the interrupts manually by writing to the module IFC register.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.34 EMU\_E201 – High Temperature Operation**

<b>Description of Errata</b>
The performance of analog peripherals at high temperatures (above 50 °C) may change over time. Firmware should periodically adjust the calibration of the analog peripherals to compensate for this behavior.
This issue affects the BOD trip point, dc-dc output voltage accuracy, ACMP reference accuracy, and IDAC output current accuracy in EM2 through EM4H power modes. This does not affect operation of these peripherals in the EM0, EM1, or EM4S power modes.
<b>Affected Conditions / Impacts</b>
The performance of analog peripherals at high temperatures may change over time.
<b>Workaround</b>
The TEMPDRV module in emdrv addresses this issue and should be included in application firmware. This module is automatically included for systems using Silicon Labs software stacks. For systems not using a Silicon Labs stack (i.e., writing code from scratch or using software examples as a starting point), firmware should include the TEMPDRV module and call the <code>TEMPDRV_Init()</code> function to improve high temperature operation (above 50 °C). The module documentation can be found in Simplicity Studio and contains more information about the firmware solution.
See AN1027: <i>EFR32xG1 and EFM32PG1/JG1 High-Temperature Operation</i> for more information.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.35 EMU\_E204 – Restrictions Writing TEMPHIGH and TEMPLOW**

<b>Description of Errata</b>
Writing TEMPHIGH and TEMPLOW in EMU_TEMPLIMITS at certain times can cause erroneous interrupts to occur.
<b>Affected Conditions / Impacts</b>
Writing to TEMPHIGH or TEMPLOW in EMU_TEMPLIMITS at any time may cause the TEMPHIGH and TEMPLOW interrupt flags in EMU_IF to trigger incorrectly.
<b>Workaround</b>
Firmware should only write TEMPHIGH and TEMPLOW within 250 ms of receiving a TEMPLOW, TEMPHIGH, or TEMP interrupt.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.36 EMU\_E205 – Restrictions Reading TEMP**

<b>Description of Errata</b>
Reads of TEMP in EMU_TEMP may not always return the correct value.
<b>Affected Conditions / Impacts</b>
Reading TEMP in EMU_TEMP at any time may read an incorrect value.
<b>Workaround</b>
Firmware should restrict its reading of TEMP to the following scenarios: <ol style="list-style-type: none"> <li>1. Read the TEMP field multiple times until the same value is returned in two consecutive reads.</li> <li>2. Read TEMP within 250 ms of receiving a TEMPLOW, TEMPHIGH, or TEMP interrupt.</li> </ol>
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.37 EMU\_E207 – GPIO State can be Lost During EM4 Recovery**

<b>Description of Errata</b>
Firmware can configure the I/O state to be retained when exiting EM4 by setting EM4IORETMODE in EMU_EM4CTRL. The desired behavior is that firmware can restore the state of the I/Os in EM0 after exit from EM4 via reset while the I/O state is maintained. It is possible for a GPIO saving a non-5 V tolerant (non-OVT) configuration pull-down configuration to lose the pull-down state if 5 V (OVT) tolerance is disabled using GPIO_Px_OVTDIS before restoring the pull-down configuration.
<b>Affected Conditions / Impacts</b>
Restoring the GPIO after an EM4 wakeup improperly can result in the I/O pull-down configuration being lost.
<b>Workaround</b>
The loss of the pull-down state can be avoided by re-enabling the pull-down before disabling the Over Voltage Tolerance for a GPIO using GPIO_Px_OVTDIS.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.38 EMU\_E208 – Occasional Full Reset After Exiting EM4H**

<b>Description of Errata</b>
Exiting EM4H may occasionally cause a full reset causing loss of RTCC counters and result in longer wakeup times on the order of power-on wakeup times.
<b>Affected Conditions / Impacts</b>
When waking up from EM4H, the system does not wait long enough for the BODs to settle before enabling it as a reset source. Even if the power is stable on DECOUPLE, AVDD, or DVDD, the part may see a BOD reset. The reset is reflected in the RESETCAUSE register.
<b>Workaround</b>
The work around is to disable the BOD prior to EM4H entry. This work around is automatically implemented when the <code>EMU_EnterEM4()</code> function is called in the Gecko SDK (versions v4.4.0 or later).
<pre>__disable_irq(); *(volatile uint32_t *) (EMU_BASE + 0x190) = 0x0000ADE8UL; *(volatile uint32_t *) (EMU_BASE + 0x198)  = (0x1UL &lt;&lt; 7);</pre>
The BODs will automatically be enabled after EM4H wakeup, but disabling the BOD will also result in disabling the EM4BOD protection. POR reset will still be valid. This work around is not needed for EM4S.
<b>Note:</b> Use the <code>RMU_ResetCauseGet()</code> function in <code>emlib</code> on EM4 wakeup to differentiate between a BOD reset and EM4RST. For Gecko SDK versions earlier than v5.0.0, the return value from <code>RMU_ResetCauseGet()</code> will be 0x0000001C (BOD reset) instead of 0x00010000 (EM4RST) when waking up from EM4H.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.39 EMU\_E209 – Potential EM2 Lock-up when using IDAC or the Debugger with the LDMA**

<b>Description of Errata</b>
The device can lock up if firmware updates the IDAC output just before entering EM2 while the LDMA module is enabled. Similarly, the device can lock up if the Debugger is connected and the firmware enters EM2 while the LDMA module is enabled.
<b>Affected Conditions / Impacts</b>
Systems using the LDMA and IDAC or LDMA and Debugger may no longer function properly after attempting to enter EM2.
<b>Workaround</b>
Two workarounds exist:
<ol style="list-style-type: none"> <li>1. If LDMA functionality in EM2 is not needed, firmware can disable the DMA via the <code>CMU-&gt;HFBUSCLKEN* LDMA</code> bit before entering EM2.</li> <li>2. If LDMA functionality in EM2 is needed, wait for the IDAC output to settle before entry into EM2 or disconnect the debugger before entry into EM2.</li> </ol>
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.40 EMU\_E210 – Potential Power-Down When Entering EM2**

<b>Description of Errata</b>
On the rare occasion when the firmware starts a transition to EM2 while the EMU is updating its Temperature Sensor measurement, and then within a 300 ns window of entering EM2 the system is woken up to EM0 or EM1, the device's internal power system can be erroneously disabled, powering down the device. When this occurs, the device will reset and will set the DECBOD flag in the RMU_RSTCAUSE register.
<b>Affected Conditions / Impacts</b>
Systems that do not use the emdrv TEMPDRV module may experience a power down on these very rare EM2 timing transitions.
<b>Workaround</b>
The TEMPDRV module in emdrv and <code>CHIP_Init()</code> (SDK version v5.0.0 or later) addresses this issue and should be included in application firmware. This module is automatically included for systems using Silicon Labs software stacks. For systems not using a Silicon Labs stack (i.e., writing code from scratch or using software examples as a starting point), firmware should include the TEMPDRV module and call the <code>TEMPDRV_Init()</code> and <code>CHIP_Init()</code> functions to prevent this issue from occurring. The module documentation can be found in Simplicity Studio and contains more information about the firmware solution.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.41 EMU\_E215 – Device May Brown Out After Energy Mode Transition**

<b>Description of Errata</b>
The logic controlling the internal voltage regulator may rarely transition to an incorrect state after a transition to EM0 or EM1. When this occurs, the device will reset and will set the DECBOD flag in the RMU_RSTCAUSE register.
<b>Affected Conditions / Impacts</b>
Systems that do not use the emdrv TEMPDRV module may experience a brown out on these rare EM0 or EM1 timing transitions.
<b>Workaround</b>
The TEMPDRV module in emdrv (SDK version v5.0.0 or later) addresses this issue and should be included in application firmware. This module is automatically included for systems using Silicon Labs software stacks. For systems not using a Silicon Labs stack (i.e., writing code from scratch or using software examples as a starting point), firmware should include the TEMPDRV module and call the <code>TEMPDRV_Init()</code> function to prevent this issue from occurring. The module documentation can be found in Simplicity Studio and contains more information about the firmware solution.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.42 EMU\_E216 – EM4H I/O Retention Cannot Be Disabled**

<b>Description of Errata</b>
Even if I/O retention is disabled by clearing EM4IORETMODE in EMU_EM4CTRL to the DISABLE setting, the affected devices behave as if I/O retention is configured as SWUNLATCH, meaning that the pin state is retained until software writes the EM4UNLATCH bit in the EMU_CMD register to remove retention.
<b>Affected Conditions / Impacts</b>
Systems that disable I/O retention mode may still see pins retain state until software writes the EM4UNLATCH bit in the EMU_CMD register to remove retention.
<b>Workaround</b>
Software should always perform an I/O unlatch after exiting EM4H by calling the <code>EMU_UnlatchPinRetention()</code> function.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.43 IDAC\_E201 – IDAC CURSTABLE Bit Not Reliable**

<b>Description of Errata</b>
The IDAC CURSTABLE flag in IDAC_STATUS may erroneously report that the current output is stable.
<b>Affected Conditions / Impacts</b>
Systems using the IDAC should not rely on the CURSTABLE to determine if the output is stable.
<b>Workaround</b>
The CURSTABLE bit must not be used. Firmware must wait the minimum required IDAC settling time listed in the data sheet.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.44 I2C\_E201 – I2C ABORT Command**

<b>Description of Errata</b>
For on-going transactions, the ABORT command in I2Cn_CMD may cause the I2C module to lock up indefinitely if it is issued in the middle of an I2C transaction.
<b>Affected Conditions / Impacts</b>
During on-going transactions, the ABORT command can cause the I2C receive module FSM to hang, locking up the I2C transaction indefinitely.
<b>Workaround</b>
The ABORT command should only be used after the I2C module is enabled in order to instruct the I2C module that the bus is IDLE. To use the ABORT command during a transaction, the I2C module should be disabled by clearing EN in I2Cn_CTRL.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.45 I2C\_E206 – Slave Holds SCL Low After Losing Arbitration**

<b>Description of Errata</b>
If, while transmitting data as a slave, arbitration is lost, SCL is unintentionally held low for an indefinite period of time.
<b>Affected Conditions / Impacts</b>
The winner of arbitration cannot use the bus because SCL is never released.
<b>Workaround</b>
If the I <sup>2</sup> C arbitration lost flag is asserted (I2C_IF_ARBLOST = 1) in slave mode (I2C_STATE_MASTER = 0), application software needs to wait for at least one SCL high time and then issue the transmission abort command (set I2C_CMD_ABORT = 1), thus releasing SCL.
<b>Resolution</b>
There is currently no resolution for this issue.



**2.46 LEUART\_E201 – Restrictions Setting TXDMAWU/RXDMAWU of LEUARTn\_CTRL**

<b>Description of Errata</b>
Changing the value of TXDMAWU while TXEN = 1 or RXDMAWU while RXEN = 1 in LEUARTn_CMD could potentially cause unpredictable behavior.
<b>Affected Conditions / Impacts</b>
If the TXDMAWU field is updated while TXEN = 1 or the RXDMAWU field is updated while RXEN = 1, a spurious DMA wake-up event could be created.
<b>Workaround</b>
Firmware should first disable the receive or transmit path using RXEN or TXEN in LEUARTn_CMD before changing RXDMAWU or TXDMAWU.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.47 RTCC\_E201 – RTCC Does Not Support Compare/Capture Wrap with Prescaler**

<b>Description of Errata</b>
When the RTCC is configured with a prescaler, the CCV1 top value enable feature enabled by setting CCV1TOP in RTCC_CTRL fails to wrap the counter when RTCC_CNT is equal to RTCC_CC1_CCV, as intended.
<b>Affected Conditions / Impacts</b>
Using CCV1TOP with a prescaled RTCC may result in the RTCC not wrapping at the desired time.
<b>Workaround</b>
Do not use a prescaler with the RTCC when using the CCV1TOP feature.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.48 RTCC\_E202 – RTCC Triggers to LETIMER Not Safe**

<b>Description of Errata</b>
The LETIMER inputs from the RTCC are connected to the LFECLK domain, while the LETIMER itself is clocked from the LFACLK domain. This results in synchronization issues between the RTCC inputs to the LETIMER and the LETIMER.
<b>Affected Conditions / Impacts</b>
RTCC triggers to LETIMER are not safe and can result in undefined behavior.
<b>Workaround</b>
Do not use RTCC triggers for LETIMER. PRS channels can be used as an alternative.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.49 RTCC\_E203 – Potential Stability Issue with RTCC Registers**

<b>Description of Errata</b>
RTCC_LOCK and RTCC_POWERDOWN have the potential to be momentarily unstable under some PCLK, Low Energy Peripheral Clock, and APB write scenarios. This stability issue resolves in approximately 160 ns as the write completes with the assertion of the APB clock pulse.
<b>Affected Conditions / Impacts</b>
A write to RTCC_LOCK or RTCC_POWERDOWN may have unintended effects if the write is completed with the Low Energy Peripheral clock enabled (RTCC in the CMU_LFECLKEN0 register is set to 1).
<b>Workaround</b>
To avoid this stability issue, configure the RTCC_LOCK and RTCC_POWERDOWN registers with the Low Energy Peripheral clock disabled (RTCC in the CMU_LFECLKEN0 register is cleared to 0).  This workaround is included in v5.1.0 or later of the Gecko SDK.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.50 TIMER\_E201 – Timer in Input Capture Mode Can Stop Counting**

<b>Description of Errata</b>
When RISEA/FALLA is configured to RELOADSTART and then changed to any other mode at the same time as when a pulse from CC0 or PRS input occurs, the counter could stop running.
<b>Affected Conditions / Impacts</b>
The timer may not count properly in all situations.
<b>Workaround</b>
To prevent input pulses while changing RISEA/FALLA: <ol style="list-style-type: none"> <li>1. If using PRS, before changing RISEA/FALLA from RELOADSTART to any other value, change the input to some other PRS input by using PRSSEL in TIMERN_CC0_CTRL. After clearing RISEA/FALLA, set PRSSEL back to the original value</li> <li>2. If using CC0 in, set GPIO mode to DISABLE for that pin before changing RISEA/FALLA. Then, set it back to INPUT mode after changing RISEA/FALLA.</li> </ol>
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.51 FLASH\_E201 – Potential Program Failure after Power On**

<b>Description of Errata</b>
There is very small probability that, with some devices, the first program of flash after a power on cycle or wake-up from EM2, EM3, EM4H, or EM4S may not take effect unless the flash has first been programmed or erased. This issue affects all devices, though the probability of the failure occurring on each device may differ.
<b>Affected Conditions / Impacts</b>
After a power-on, the first program of flash initiated by firmware may not take effect.
<b>Workaround</b>
To ensure the flash is programmed correctly, firmware should program the flash, verify the flash contents, and reprogram the flash, if necessary. This workaround is included in v4.4.0 or later of the Gecko SDK. Note that typical word (32-bit) programming time is 26 $\mu$ s, while it only takes several tens of ns to read and verify, so the overhead is minimal.  An alternative workaround is to erase the flash page before programming after a power on cycle or wake-up from EM2, EM3, EM4S, or EM4H.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.52 RMU\_E201 – CTRL Register Reset on All Resets**

<b>Description of Errata</b>
The RMU->CTRL register is reset to default state on every system reset, not only POR and hard pin reset as stated in the reference manual.
<b>Affected Conditions / Impacts</b>
The RMU->CTRL register is reset to default state on every system reset, not only POR and hard pin reset as stated in the reference manual.
<b>Workaround</b>
Firmware should always update RMU->CTRL after a reset. Note that the LOCKUP reset is disabled by default.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

**2.53 RMU\_E202 – External Debug Access Not Available After Watchdog or Lockup Full Reset**

<b>Description of Errata</b>
When a reset is triggered in full-reset mode, a debugger will not be able to read AHB-AP or ARM core registers.
<b>Affected Conditions / Impacts</b>
Systems using the full reset mode for watchdog or lockup resets will see limited debugging capability after one of these resets triggers.
<b>Workaround</b>
There are three possible workarounds: <ul style="list-style-type: none"> <li>• Software should configure peripherals to either LIMITED or EXTENDED mode if full debugger functionality is needed after a watchdog or lockup reset.</li> <li>• When using FULL reset mode, appending at least 9 idle clock cycles to the last debug command will allow the transaction to complete.</li> <li>• A power cycle or hard pin reset will restore normal operation.</li> </ul>
<b>Resolution</b>
There is currently no resolution for this issue.

**2.54 USART\_E202 — Incorrect 8-bit Timer Operation in Asynchronous Mode**

<b>Description of Errata</b>
The 8-bit Timer logic that creates events within the USART works correctly in synchronous (USART) mode, but is not correct for asynchronous (UART) mode. As a result, it is not recommended to use the 8-bit Timer feature with asynchronous mode on affected devices.
<b>Affected Conditions / Impacts</b>
Systems using the USART module in asynchronous (UART) mode should not use the 8-bit Timer feature.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
There is currently no resolution for this issue on EFR32BG1. This issue is resolved on EFR32BG14.

### 3. Errata History

This section contains the errata history for Blue Gecko SoC (EFR32BG1) devices. Note that many issues on this device family are resolved in EFR32BG14 devices. The EFR32BG14 devices are very similar to EFR32BG1, and migration will require few changes. More information can be found here: <https://www.silabs.com/products/wireless/bluetooth>.

For errata on latest revision, please 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 Errata History Summary

This table lists all resolved errata for the Blue Gecko SoC (EFR32BG1).

**Table 3.1. Errata History Status Summary**

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Resolution
1	CUR_E201	<a href="#">EM2 and EM3 Current Consumption</a>	No	C	Documented in the revision 1.3 data sheet. Fixed in EFR32BG14.
2	RADIO_E206	<a href="#">Bluetooth Smart Receive Sensitivity</a>	No	C	Documented in the revision 1.3 data sheet. Fixed in EFR32BG14.
3	RADIO_E208	<a href="#">Receive Sensitivity</a>	No	C	Documented in the revision 1.3 data sheet. Fixed in EFR32BG14.
4	CUR_E202	<a href="#">EM2/3 Current Consumption at Cold Temperatures</a>	No	B	B, date code 1547 (November 16, 2015)
5	RADIO_E202	<a href="#">802.15.4 Channel 14</a>	No	C	Documented in the revision 1.0 data sheet
6	RADIO_E203	<a href="#">Bluetooth Smart Channel 26</a>	No	C	Documented in the revision 1.0 data sheet
7	RADIO_E204	<a href="#">Increased EVM on Selected Channels</a>	No	C	Documented in the revision 1.0 data sheet
8	RADIO_E207	<a href="#">Sensitivity at 2.42 GHz</a>	No	C	Documented in the revision 1.0 data sheet
9	GPIO_E201	<a href="#">GPIO Default Slew Rate</a>	Yes	B	B, date code 1603 (January 18, 2016)
10	DCDC_E201	<a href="#">DCDC Stops Regulating During a Fast EM0/1 to EM2/3/4H Transition</a>	Yes	B	C

## 3.2 Detailed Errata Descriptions

### 3.2.1 CUR\_E201 – EM2 and EM3 Current Consumption

<b>Description of Errata</b>		
The current consumption on revision C devices is typically:		
Mode	Data Sheet Specified Value	Measured Value
EM2 (RTCC, LFXO, 32KB) 3.3 V with DC-DC	1.4 $\mu$ A	2.5 $\mu$ A
EM2 (RTCC, LFRCO, 4KB) 3.3 V with DC-DC	1.4 $\mu$ A	2.2 $\mu$ A
EM3 (CRYO, ULFRCO, 32KB) 3.3 V with DC-DC	1.1 $\mu$ A	2.1 $\mu$ A
As shown, the measured values are higher than the values specified in the device data sheet.		
<b>Affected Conditions / Impacts</b>		
The increased current consumption impacts applications that spend the majority of their time in these sleep modes. For applications that are dominated by current consumption from the higher energy modes, EM0 and EM1, the impact will be negligible.		
<b>Workaround</b>		
There is currently no workaround for this issue.		
<b>Resolution</b>		
This issue is documented in the revision 1.3 and later device data sheet. This issue is also resolved on EFR32BG14.		

### 3.2.2 RADIO\_E206 – Bluetooth Smart Receive Sensitivity

<b>Description of Errata</b>	
The average receive sensitivity measured for Bluetooth Smart is –92.5 dBm (Reference Signal), which is worse than the specified –94 dBm (Reference Signal).	
<b>Affected Conditions / Impacts</b>	
The receive sensitivity is worse than expected. For applications in noisy RF environments, the impact will be negligible.	
<b>Workaround</b>	
There is currently no workaround for this issue.	
<b>Resolution</b>	
This issue is documented in the revision 1.3 and later device data sheet. This issue is also resolved on EFR32BG14.	

### 3.2.3 RADIO\_E208 – Receive Sensitivity

<b>Description of Errata</b>	
The average receive sensitivity measured when configured to support 2.4 GHz 250 kbps, DSSS-OQPSK is –99 dBm.	
<b>Affected Conditions / Impacts</b>	
The receive sensitivity is lower than expected.	
<b>Workaround</b>	
There is currently no workaround for this issue.	
<b>Resolution</b>	
This issue is documented in the revision 1.3 and later device data sheet. This issue is also resolved on EFR32BG14.	

### 3.2.4 CUR\_E202 – EM2/3 Current Consumption at Cold Temperatures

<b>Description of Errata</b>
A small probability exists that the current consumption in EM2 and EM3 on some revision B devices could be on the order of 20 $\mu$ A. This issue can only be observed at cold temperatures with devices that are fabricated under certain semiconductor process variations.
<b>Affected Conditions / Impacts</b>
The increased current consumption impacts applications that spend the majority of their time in these sleep modes. For applications that are dominated by current consumption from the higher energy modes, EM0 and EM1, the impact will be negligible.
<b>Workaround</b>
The higher leakage current can be significantly reduced by setting RAMPOWERDOWN in EMU_RAM0CTRL to BLK1TO4 (power down RAM blocks 1 and above) in EM2 and EM3.
<b>Resolution</b>
Revision B devices after date code 1547 (November 16, 2015) will not exhibit high current on the order of 20 $\mu$ A. More information on the date code can be found in the Package Marking diagrams in the device data sheet.

### 3.2.5 RADIO\_E202 – 802.15.4 Channel 14

<b>Description of Errata</b>
Receive sensitivity of 802.15.4 channel 14 is lower than expected. The average receive sensitivity of 802.15.4 channel 14 is currently –98 dBm.
<b>Affected Conditions / Impacts</b>
The average receive sensitivity of 802.15.4 channel 14 is currently lower than expected.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
This issue is documented in the revision 1.0 and later device data sheet.

### 3.2.6 RADIO\_E203 – Bluetooth Smart Channel 26

<b>Description of Errata</b>
Receive sensitivity of Bluetooth Smart channel 26 (2458 MHz) is lower than expected. The average receive sensitivity of Bluetooth Smart Channel 26 is currently –86 dBm (Reference Signal) and –85 dBm (Dirty Transmitter).
<b>Affected Conditions / Impacts</b>
Receive sensitivity of Bluetooth Smart channel 26 is worse than expected.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
This issue is documented in the revision 1.0 and later device data sheet.

### 3.2.7 RADIO\_E204 – Increased EVM on Selected Channels

<b>Description of Errata</b>
EVM is increased for one 802.15.4 channel.
<b>Affected Conditions / Impacts</b>
Typical EVM will be increased to 7.8% for the 2415 MHz 802.15.4 channel.
<b>Workaround</b>
No workaround required. The 802.15.4 specification requires an EVM of less than 35%.
<b>Resolution</b>
This issue is documented in the revision 1.0 and later device data sheet.

### 3.2.8 RADIO\_E207 – Sensitivity at 2.42 GHz

<b>Description of Errata</b>
Receive sensitivity at 2.42 GHz is lower than expected. The average receive sensitivity (1% PER) at 2 Mbps, 2GFSK is currently – 89.9 dBm. The amount of sensitivity degradation may vary slightly depending on the modulation type and bit rate.
<b>Affected Conditions / Impacts</b>
The average receive sensitivity at 2.42 GHz is currently lower than expected.
<b>Workaround</b>
There is currently no workaround for this issue.
<b>Resolution</b>
This issue is documented in the revision 1.0 and later device data sheet.

### 3.2.9 GPIO\_E201 – GPIO Default Slew Rate

<b>Description of Errata</b>
The default SLEWRATE and SLEWRATEALT value in the GPIO_Pn_CTRL registers is set too high.
<b>Affected Conditions / Impacts</b>
The default SLEWRATE and SLEWRATEALT setting of 0x6 may result in I/O ringing and excessive undershoot, which can lead to a risk of excessive current injection.
<b>Workaround</b>
<p>The SLEWRATE and SLEWRATEALT fields for all GPIO_Pn_CTRL registers should be changed to a maximum value of 0x5 for most MCU applications. The control of SLEWRATE and SLEWRATEALT is application specific. For GPIO pins that are actively toggling during RF activity, consider reducing their slew rate to a minimum possible value in order to avoid spurs and interference with radio communications.</p> <p>Firmware can call the CHIP_Init() function in versions v4.3.0 or later of emlib to write a default value of 0x5 to the SLEWRATE and SLEWRATEALT fields for all GPIO_Pn_CTRL registers. If using a software stack on top of emlib, check the documentation for the version of emlib used.</p>
<b>Resolution</b>
Revision B devices after date code 1603 (January 18, 2016) will have the default slew rate set to 0x05. More information on the date code can be found in the Package Marking diagrams in the device data sheet.



### 3.2.10 DCDC\_E201 – DCDC Stops Regulating During a Fast EM0/1 to EM2/3/4H Transition

<b>Description of Errata</b>
The DC-DC module can stop regulating during a fast transition from EM0 or EM1 to EM2, EM3, or EM4H.
<b>Affected Conditions / Impacts</b>
The LP controller stops charging the capacitor on the DC-DC output, resulting in a brown-out.
<b>Workaround</b>
Before changing DCDCCTRL->DCDCMODE (to turn on the DCDC), clear DCDCSMCTRL->LPCMPWAITDIS (bit 0 of DCDCSMCTRL). Wait for the low noise controller to start running before changing energy modes by polling DCDCSTATUS->LNRUNNING (bit 16 of the register at address EMU_BASE+0x07C).
<b>Resolution</b>
This issue is resolved in revision C devices.

## 4. Revision History

### Revision 1.7

June, 2018

- Adjusted the resolution wording for almost all errata listed.
- Updated the workaround in [RMU\\_E202](#).
- Moved [CUR\\_E201](#), [RADIO\\_E206](#), and [RADIO\\_E208](#) to [3. Errata History](#) since these new electrical specifications are documented in revision 1.3 and later of the data sheet.
- Added [I2C\\_E206](#).

### Revision 1.6

December, 2017

- Updated the date for the future product to 2018.
- Updated the workaround description for [EMU\\_E208](#).
- Adjusted the resolution wording for almost all errata listed.
- Updated the resolution for [ADC\\_E222](#).
- Added [ADC\\_E224](#), [ADC\\_E226](#), [ADC\\_E227](#), [DBG\\_E204](#), [DCDC\\_E205](#), [DCDC\\_E206](#), [EMU\\_E215](#), [EMU\\_E216](#), [RMU\\_E202](#), [RTCC\\_E203](#), and [USART\\_E202](#).
- Updated the workaround description of [EMU\\_E210](#) to include mention of `CHIP_Init()` in addition to `TEMPDRV`.
- Merged errata history and errata into one document.
- Updated revision history format.

### Revision 1.5

November, 2016

- Added [CRYPTO\\_E101](#), [DCDC\\_E202](#), and [DCDC\\_E203](#).

### Revision 1.4

July, 2016

- Moved [RADIO\\_E202](#), [RADIO\\_E203](#), [RADIO\\_E204](#), and [RADIO\\_E207](#) to the errata history.
- Updated the resolution for all remaining errata other than [ADC\\_E213](#) as fixed in a future revision.
- Added [ADC\\_E220](#), [ADC\\_E221](#), [ADC\\_E222](#), [ADC\\_E223](#), [EMU\\_E209](#), and [EMU\\_E210](#).
- Added a reference to AN1027 to [EMU\\_E201](#).

### Revision 1.3

April, 2016

- Updated the latest revision to revision C.
- Added [RADIO\\_E202](#), [RADIO\\_E207](#), [RADIO\\_E208](#), [RADIO\\_E204](#), [TIMER\\_E201](#), [ADC\\_E216](#), [ADC\\_E217](#), [ADC\\_E218](#), [ADC\\_E219](#), [EMU\\_E208](#), [FLASH\\_E201](#), [RTCC\\_E202](#), and [RMU\\_E201](#).
- Moved [CUR\\_E202](#) and [GPIO\\_E201](#) to the errata history. Also added [DCDC\\_E201](#) to the errata history.
- Updated the typical sensitivity and wording of [RADIO\\_E203](#).

### Revision 1.2

February, 2016

- Initial release.

Silicon Labs

# Simplicity Studio™4



## 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](http://www.silabs.com/IoT)



**SW/HW**  
[www.silabs.com/simplicity](http://www.silabs.com/simplicity)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

### Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

### Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOModem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



**SILICON LABS**

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

<http://www.silabs.com>