EFM32 Gecko
EFM32GG11 Errata

This document contains information on the EFM32GG11 errata. The latest available revision of this device is revision B. Several revision A part numbers are available at Engineering Status, which is denoted with revision code X. For the purposes of this document, revision “A/X” refers to both revision A and revision X part numbers.

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

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

Errata effective date: September, 2020.

1. Errata Summary

The table below lists all known errata for the EFM32GG11 and all unresolved errata in revision B of the EFM32GG11.

Table 1.1. Errata Overview

Designator	Title/Problem	Workaround Exists	Exists on Revision:	
			A/X	B
ADC_E213	ADC KEEPINSLOWACC Mode	No	X	X
ADC_E224	ADC Warm-Up Ready Can Cause IDAC, ACMP, or CSEN to Not Function	Yes	X	—
ADC_E225	Using the ADC in High-Accuracy Bias Mode Will Force All Analog Peripherals to High-Accuracy Bias Mode	No	X	—
ADC_E228	Limited ADC Sampling Frequency in EM2	No	X	—
BU_E201	Extra Current from BUVDD to VREGVDD in BUMODE when Pulling Main Supply Low	No	X	—
BU_E202	Extra Current from DVDD to GND in BUMODE when Pulling BUVDD Low	Yes	X	—
CMU_E203	Peak Detector May Not Trip	Yes	X	—
CMU_E204	Initial Oscillator Calibration	Yes	X	—
CORE_E203	Invalid Data Cached After a Bus Fault	Yes	X	X
CORE_E204	SRAM Does Not Support Prefetch When ECC is Enabled	Yes	X	X
CORE_E205	SRAM Slave (RAM0, RAM1) ECC Related Busfaults Cannot be Blocked for Non-Word Accesses	No	X	—
CORE_E206	SRAM Slave (RAM0, RAM1) Ignores the Second AHB Transaction for Back-to-Back AHB Transactions of Which the First AHB Transaction Caused a 2-bit ECC Related Busfault	Yes	X	X
CSEN_E201	CSEN_DATA in Debug Mode	Yes	X	X
CSEN_E202	CSEN Baseline DMA Transfers	Yes	X	X
CUR_E203	Occasional Extra EM0/1 Current	No	X	X
CUR_E204	Extra EM4S Current When ANASW Set to 1	Yes	X	X
DBG_E204	Debug Recovery with JTAG Does Not Work	Yes	X	X
EMU_E214	Device Erase Cannot Occur if Voltage Scaling Level is Too Low	Yes	X	—
EMU_E217	EM4S Not Supported in 5V Sub-System Powered Devices at Temperatures Above 85°C	No	X	X
EMU_E220	DECBOD Reset During Voltage Scaling After EM2 or EM3 Wake-up	Yes	X	X
GPIO_E202	GPIO Mode PUSHPULLALT Not supported for High Speed Priority Locations of Alternate Functions	No	X	—
I2C_E202	Race Condition Between Start Detection and Timeout	Yes	X	—
I2C_E203	I2C Received Data Can be Shifted	Yes	X	—
I2C_E204	I2C0 Does Not Meet Fast Mode Plus (Fm+) Timing at Voltage Scale Level 0	No	X	X
I2C_E205	Go Idle Bus Idle Timeout Does Not Bring Device to Idle State	Yes	X	—

Designator	Title/Problem	Workaround Exists	Exists on Revision:	
			A/X	B
I2C_E206	Slave Holds SCL Low After Losing Arbitration	Yes	X	X
I2C_E207	I2C Fails to Indicate New Incoming Data	Yes	X	X
LCD_E201	LCD Boost Mode Does Not Work	No	X	—
LES_E201	LFPRESC Can Extend Channel Start-Up Delay	Yes	X	X
MSC_E201	Invalid Data Cached After a Bus Fault	Yes	X	X
MSC_E202	SRAM Does Not Support Prefetch When ECC is Enabled	Yes	X	X
MSC_E203	ECC-Related Bus Faults Cannot be Blocked for Non-Word Accesses	No	X	—
MSC_E204	Second Transaction of Back-to-Back SRAM Transactions is Ignored When First Transaction Causes a 2-bit ECC Bus Fault	Yes	X	X
RMU_E202	External Debug Access Not Available After Watchdog or Lockup Full Reset	Yes	X	X
RMU_E203	AVDD Ramp Issue	Yes	X	—
RTCC_E205	Wrap Event Can Be Missed	Yes	X	—
SDIO_E201	Internal Pull-up Resistors not Enabled	Yes	X	—
TIMER_E202	Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode	Yes	X	X
USART_E201	USART DMA Transactions Fail with Slow Peripheral Clocks	No	X	X
USART_E204	IrDA Modulation and Transmission of PRS Input Data	Yes	X	X
USART_E205	Possible Data Transmission on Wrong Edge in Synchronous Mode	Yes	X	X
USART_E206	Additional SCLK Pulses Can Be Generated in USART Synchronous Mode	Yes	X	X
USB_E201	USB PHY Short Circuit Failure	No	X	—
USB_E202	USB Regulator Does Not Read Tuning Register R5VOUTLEVEL After Power Loss and Restore	Yes	X	—
VDAC_E201	VDAC Output Drives All APORT Buses Simultaneously	Yes	X	—
VDAC_E202	PRS Outputs Not Generated when Interrupt Flag is Set	Yes	X	—
WDOG_E201	Clear Command is Lost Upon EM2 Entry	Yes	X	X
WTIMER_E201	Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode	Yes	X	X

2. Current Errata Descriptions

2.1 ADC_E213 – ADC KEEPINSLOWACC Mode

Description of Errata
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.
Affected Conditions / Impacts
KEEPINSLOWACC warmup mode does not function properly.
Workaround
There is currently no workaround for this issue.
Resolution
There is currently no resolution for this issue.

2.2 CORE_E203 – Invalid Data Cached After a Bus Fault

Description of Errata
The icache flush is not triggered at the event of a bus fault. As a result, when an instruction fetch results in a bus fault, invalid data may be cached.When an invalid data is cached due to bus fault, the next time the instruction that caused the bus fault is fetched, the processor core will get the invalid cached data without any bus fault.
Affected Conditions / Impacts
All EFM32GG11 devices are affected. This is most notable if software implements a custom bus fault handler.
Workaround
To workaround this issue, software can set the `MSC->CACHECMD.INVCACHE` bitfield to 1 at the event of a bus fault to invalidate the ic-ache.
Resolution
This issue has been reclassified as MSC_E201.

2.3 CORE_E204 – SRAM Does Not Support Prefetch When ECC is Enabled

Description of Errata
SRAM slave (RAM0, RAM1) does not support prefetch when ECC is enabled.
Affected Conditions / Impacts
1-bit errors can lead to further memory corruption due to erroneous write-back. The optional busfault upon 2-bit errors can cause invalid AHB responses leading to unpredictable bus master behaviour.
Workaround
Disabling the RAM prefetch feature will prevent the issue.
Resolution
This issue has been reclassified as MSC_E202.

2.4 CORE_E206 – SRAM Slave (RAM0, RAM1) Ignores the Second AHB Transaction for Back-to-Back AHB Transactions of Which the First AHB Transaction Caused a 2-bit ECC Related Busfault

Description of Errata
SRAM slave (RAM0, RAM1) ignores the second AHB transaction for back-to-back AHB transactions of which the first AHB transaction caused a 2-bit ECC related busfault.
Affected Conditions / Impacts
Upon an ignored read transaction the returned read RAM data is wrong. Upon an ignored write transaction RAM is not written.
Workaround
If back-to-back transactions to a specific SRAM slave (RAM0, RAM1) are limited to be from the same master and that master is either the Cortex-M4 (when configured to only use precise busfault errors) or the LDMA, then the issue will not occur.
Resolution
This issue has been reclassified as MSC_E204.

2.5 CSEN_E201 – CSEN_DATA in Debug Mode

Description of Errata
Reading CSEN_DATA in debug mode inadvertently clears pending CSEN data DMA requests.
Affected Conditions / Impacts
Reads of CSEN_DATA clear pending receive data DMA requests. This would be expected in normal operation as the DMA reads CSEN_DATA to transfer newly acquired results. These reads are intentional, but any read of CSEN_DATA, including while in debug mode, has the same effect. Thus, viewing the CSEN module registers in a debugger, such as in Simplicity Studio, can inadvertently clear pending CSEN DMA requests resulting in subsequent data being received out of order and with insertions of random data.
Workaround
Do not use a debugger to read the CSEN registers while DMA is enabled.
Resolution
There is currently no resolution for this issue.

2.6 CSEN_E202 – CSEN Baseline DMA Transfers

Description of Errata
DMA transfers to CSEN_DMBASELINE do not occur in the expected order.
Affected Conditions / Impacts
When using delta modulation, a baseline value must be written to CSEN_DMBASELINE before each conversion. However, when DMA is used to do this, these writes occur after the desired conversion instead of before the conversion as is required. This means that in a given sequence of conversions serviced by DMA, the write to CSEN_DMBASELINE that should happen before conversion N is actually written in advance of conversion N + 1, leading to potentially erroneous results.
Workaround
Manually write the first value to CSEN_DMBASELINE and then use the DMA to perform subsequent baseline writes. Therefore, in the case of a sequence consisting of four conversions, the first baseline value would be written to CSEN_DMBASELINE under software control (e.g., before the conversion trigger occurs). The next three values can be written by the DMA after the first and each subsequent conversion occurs. After the final conversion, which would be the fourth in this example, the DMA will service a final write request to CSEN_DMBASELINE. This final transfer can be (1) a dummy value if no further conversions are required, (2) the initial baseline value in the case where conversions are repeated in a loop, or (3) the initial baseline value for a new, yet-to-be-triggered series of conversions.
Resolution
There is currently no resolution for this issue.

2.7 CUR_E203 – Occasional Extra EM0/1 Current

Description of Errata
Occasionally, when exiting EM2, a low voltage oscillator sometimes continues to run and causes the device to draw an extra ~10 µA when in EM0 or EM1. This oscillator automatically resets when entering EM2 or EM3, so the extra current draw is not present in these modes.
Affected Conditions / Impacts
Systems using EM2 may occasionally see an extra ~10 µA of current draw in EM0 or EM1.
Workaround
There is currently no workaround for this issue.
Resolution
There is currently no resolution for this issue.

2.8 CUR_E204 – Extra EM4S Current When ANASW Set to 1

Description of Errata
After entering EM4S while analog peripherals are powered from DVDD (ANASW = 1 in the EMU_PWRCTRL register), the device will see an additional 30-300 µA of current consumption, depending on the AVDD voltage.
Affected Conditions / Impacts
Systems using EM4S will see an additional 30-300 µA of current draw if ANASW in EMU_PWRCTRL is set to 1.
Workaround
Firmware can clear ANASW to 0 before entering EM4S to reduce current consumption.
Resolution
There is currently no resolution for this issue.

2.9 DBG_E204 – Debug Recovery with JTAG Does Not Work

Description of Errata
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.
Affected Conditions / Impacts
The debug recovery sequence will not work when using the JTAG debug interface.
Workaround
Use the Serial Wire debug interface to implement the debug recovery sequence.
Resolution
There is currently no resolution for this issue.

2.10 EMU_E217 — EM4S Not Supported in 5V Sub-System Powered Devices at Temperatures Above 85°C

Description of Errata
When system uses 5V sub-system to power up the EFM chip, energy mode EM4 Shutoff is only supported up to 85C. When a system that uses the 5V sub-system to power up the EFM chip is in EM4 Shutoff at a temperature above 85C, the output voltage VREGO of the 5V sub-system will not stay above the required minimum AVDD voltage of 1.62V due to added leakage at high temperatures, which might be above the maximum 20uA current source capability of the 5V sub-system in EM4 Shutoff.
Affected Conditions / Impacts
Systems using the 5V sub-system and EM4S at high temperatures.
Workaround
There is currently no workaround for this issue. The recommendation for the 5V sub-system powered devices is to use EM4 Hibernate instead of EM4 Shutoff at temperatures above 85C.
Resolution
There is currently no resolution for this issue.

2.11 EMU_E220 – DECBOD Reset During Voltage Scaling After EM2 or EM3 Wakeup

Description of Errata
An infrequent, asynchronous and unrelated internal event can intermittently delay normal BOD state-machine transition sequencing during voltage scaling from VSCALE0 (1.0 Vdc) to VSCALE2 (1.2 Vdc) when emerging from EM2/EM3 to EM0. This delay can cause erroneous DECBOD resets on some devices.
Affected Conditions / Impacts
Systems operating with core voltage scaling can experience a decouple voltage brownout reset (DECBOD) when exiting EM2 or EM3.
Workaround

Systems that use core voltage scaling need to enter EM2 or EM3 via a RAM executed wait for interrupt instruction with interrupts disabled. Additionally, the EMU writes shown below should be added around EM2/EM3 entry and exit and after voltage scaling completes. This prevents the BOD state-machine transition signals from being delayed. This workaround adds 2.7 µs to the voltage scaling operation.
Note: This workaround is included in `em_emu.c` in the v2.7.4.0 or later of the Gecko SDK. It is recommended to workaround this issue by using the latest Gecko SDK version.
 	
`// Execute from RAM with interrupts disabled`
`*(uint32_t *) (EMU_BASE + 0x1A4) |= 0x1f << 10;`
`__WFI();`
`*(uint32_t *) (EMU_BASE + 0x14C) |=  0x01 << 31;`
`// Enable Interrupts and return to flash execution`
 	
`. . . .`
 	
`// After voltage scaling is complete`
`*(uint32_t *) (EMU_BASE + 0x14C) &= ~(0x01 << 31);`
`EMU->IFC = 0xFFFFFFFF;`

Resolution
There is currently no resolution for this issue.

2.12 I2C_E204 – I2C0 Does Not Meet Fast Mode Plus (Fm+) Timing at Voltage Scale Level 0

Description of Errata
I2C0 cannot meet Fast Mode Plus (Fm+) timing specifications when the device is operating at Voltage Scale Level 0 (`EMU_STATUS_VSCALE = VSCALE0`). Attempting to do so can result in false NACK/START/STOP conditions during communication.
Affected Conditions / Impacts
I2C0 is unable to support Fm+ timing at Voltage Scale Level 0 (`EMU_STATUS_VSCALE = VSCALE0`).
Workaround
There is no workaround for I2C0, and this erratum affects only I2C0 on a given device.
Resolution
There is currently no resolution for this issue.

2.13 I2C_E206 – Slave Holds SCL Low After Losing Arbitration

Description of Errata
If, while transmitting data as a slave, arbitration is lost, SCL is unintentionally held low for an indefinite period of time.
Affected Conditions / Impacts
The winner of arbitration cannot use the bus because SCL is never released.
Workaround
If the I^2C 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.
Resolution
There is currently no resolution for this issue.

2.14 I2C_E207 – I^2C Fails to Indicate New Incoming Data

Description of Errata
A race condition exists in which the I^2C fails to indicate reception of new data when both user software attempts to read data from and the I^2C hardware attempts to write data to the I2C_RXFIFO in the same cycle.
Affected Conditions / Impacts
When this race condition occurs, the RXFIFO enters an invalid state in which both I2C_STATUS_RXDATAV = 0 and I2C_STATUS_RXFULL = 1. This causes the I^2C to discard new incoming data bytes because RXFULL = 1 and would otherwise prevent user software from reading last byte written by the I^2C hardware to RXFIFO because RXDATAV = 0.
Workaround
User software can recognize and clear this invalid RXDATAV = 0 and RXFULL = 1 condition by performing a dummy read of the RXFIFO (I2C_RXDATA). This restores the expected RXDATAV = 1 and RXFULL = 0 condition. The data from this read can be discarded, and user software can now read the last byte written by the I^2C hardware to the RXFIFO (the byte which caused the invalid RXDATAV = 0 and RXFULL = 1 condition). No data will be lost as long as user software completes this recovery procedure (performing the dummy read and then reading the remaining valid byte in the RXFIFO) before the I^2C hardware receives the next incoming data byte.
Resolution
There is currently no resolution for this issue.

2.15 LES_E201 — LFPRESC Can Extend Channel Start-Up Delay

Description of Errata
Setting LESENSE_TIMCTRL_LFPRESC to a value other than DIV1 may delay channel start-up longer than the number of $LFACLK_{LESENSE}$ clock cycles specified by LESENSE_TIMCTRL_STARTDLY.
Affected Conditions / Impacts
Delaying channel start-up delays the subsequent excitation and measurement phases and may have an impact on the data returned by the LESENSE.
Workaround
If a channel start-up delay is used (LESENSE_TIMCTRL_STARTDLY > 0), LESENSE_TIMCTRL_LFPRESC must be set to DIV1.
Resolution
There is currently no resolution for this issue.

2.16 MSC_E201 – Invalid Data Cached After a Bus Fault

Description of Errata
The instruction cache is not flushed in the event of a bus fault. As a result, when an instruction fetch results in a bus fault, invalid data may be cached. When invalid data is cached due to a bus fault, the next time the instruction that caused the bus fault is fetched, the processor core will get the invalid cached data without any bus fault.
Affected Conditions / Impacts
This problem manifests itself in software that implements a bus fault handler because the processing of the bus fault does not invalidate the instruction cache.
Workaround
Set `MSC->CACHECMD.INVCACHE=1` upon entering the bus fault handler to invalidate the instruction cache.
Resolution
There is currently no resolution for this issue.

2.17 MSC_E202 – SRAM Does Not Support Prefetch When ECC is Enabled

Description of Errata
Prefetch cannot be used when the corresponding SRAM has ECC enabled.
Affected Conditions / Impacts
Erroneous write-back operations can lead to further corruption when 1-bit ECC errors occur. The optional bus fault upon 2-bit errors (`MSC->CTRL.RAMECCERRFAULTEN=1`) can cause the SRAM to generate invalid bus protocol responses, leading to unpredictable bus master (e.g. CPU or LDMA) behavior.
Workaround
Disable the SRAM prefetch feature by leaving `MSC->RAMCTRL.RAMPREFETCHEN=0` and `MSC->RAMCTRL.RAM1PREFETCHEN=0` to prevent this issue.
Resolution
There is currently no resolution for this issue.

2.18 MSC_E204 – Second Transaction of Back-to-Back SRAM Transactions is Ignored When First Transaction Causes a 2-bit ECC Bus Fault

Description of Errata
When SRAM ECC (via `MSC->ECCCTRL`) and bus fault upon 2-bit ECC error (`MSC->CTRL.RAMECCERRFAULTEN=1`) are enabled, the second transaction of back-to-back transactions is ignored if the first transaction caused a 2-bit ECC bus fault.
Affected Conditions / Impacts
Upon an ignored read transaction, the data read from SRAM is wrong. Upon an ignored write transaction, SRAM is not written.
Workaround
If back-to-back transactions to a specific SRAM slave (RAM0 or RAM1) are limited to be from the same master and that master is either the Cortex-M4 (when configured to only use precise busfault errors) or the LDMA, then the issue will not occur.
Resolution
There is currently no resolution for this issue.

2.19 RMU_E202 – External Debug Access Not Available After Watchdog or Lockup Full Reset

Description of Errata
When a reset is triggered in full-reset mode, a debugger will not be able to read AHB-AP or ARM core registers.
Affected Conditions / Impacts
Systems using the full reset mode for watchdog or lockup resets will see limited debugging capability after one of these resets triggers.
Workaround
There are three possible workarounds: • Software should configure peripherals to either LIMITED or EXTENDED mode if full debugger functionality is needed after a watchdog or lockup reset. • When using FULL reset mode, appending at least 9 idle clock cycles to the last debug command will allow the transaction to complete. • A power cycle or hard pin reset will restore normal operation.
Resolution
There is currently no resolution for this issue.

2.20 TIMER_E202 — Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode

Description of Errata
When the TIMER is configured to operate in quadrature decoder mode with the overflow interrupt enabled and the counter value (TIMER_CNT) reaches the top value (TIMER_TOP), the overflow interrupt is requested continuously even if the interrupt flag (TIMER_IF_OF) is cleared. Similarly, if the underflow interrupt is enabled and the counter value reaches zero, the underflow interrupt is requested continuously even if the interrupt flag (TIMER_IF_UF) is cleared. The interrupt can be cleared only after the counter value has incremented or decremented so that the overflow or underflow condition no longer applies.
Affected Conditions / Impacts
Because the counter is clocked by its CC0 and CC1 inputs in quadrature decoder mode and not the prescaled HFPERCLK, overflow and underflow events remain latched as long as TIMER_CNT remains at the value that triggered the overflow or underflow condition. Until the counter is no longer in the overflow or underflow condition, it is not possible to clear the associated interrupt flag.
Workaround

Short of disabling the relevant interrupts, the simplest workaround is to manually change TIMER_CNT so that the overflow or underflow condition no longer exists. Insert the following or similar code in the interrupt handler for the timer in question (TIMER0 in this case) to do this:
`uint32 intFlags = TIMER_IntGet(TIMER0);`
 	
`if((intFlags & TIMER_IF_OF) && (TIMER0->CNT == TIMER0->TOP))`
`  TIMER0->CNT = 0;`
 	
`if((intFlags & TIMER_IF_UF) && (TIMER0->CNT == 0x0))`
`  TIMER0->CNT = TIMER0->TOP;`
 	
It may be necessary for firmware to account for this adjustment in calculations that include the counter value.

Resolution
There is currently no resolution for this issue.

2.21 USART_E201 — USART DMA Transactions Fail with Slow Peripheral Clocks

Description of Errata
USART DMA transactions will fail when the USART peripheral clock is slower than the DMA clock and IGNORESREQ is cleared to 0.
Affected Conditions / Impacts
Systems will not be able to use the DMA with a USART running from a slow clock when IGNORESREQ is cleared to 0.
Workaround
There is currently no workaround for this issue.
Resolution
There is currently no resolution for this issue.

2.22 USART_E204 — IrDA Modulation and Transmission of PRS Input Data

Description of Errata
If the USART IrDA modulator is configured to accept input from a PRS channel, the incoming data stream will not be transmitted because the required clock from the baud rate generator is never enabled.
Affected Conditions / Impacts
It is not possible for the USART IrDA modulator to directly transmit data from a source other than the USART's own transmitter. The USART_IRCTRL_IRPRSEN bit should remain at its reset state of 0.
Workaround
Assuming the data to be sent via the PRS is also data that could be received by the EFM32/EFR32 USART, then the data can be received using the USART's PRS RX feature (USART_INPUT_RXPRS = 1), stored in RAM (e.g., using DMA), and then transmitted with IrDA mode enabled. In cases where IrDA operation is transmit-only, the PRS RX data can be received on the same USART doing the transmission. If IrDA operation is bidirectional, then another USART must be used to receive the PRS data. If the data to be sent is in some other format (e.g., pulses from a timer output), then there is no direct way to transmit it using the IrDA modulator. It would be necessary to capture the data in some other way and reformat it as serial data timed according to the clock generated by the USART.
Resolution
There is currently no resolution for this issue.

2.23 USART_E205 — Possible Data Transmission on Wrong Edge in Synchronous Mode

Description of Errata
If the USART is configured to operate in synchronous mode with... 1. USART_CLKDIV_DIV = 0 (clock = $f_{HFPERCLK} \div 2$) 2. USART_CTRL_CLKPHA = 0 3. USART_TIMING_CSHOLD = 1 ...and 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 chip select hold time (USART_TIMING_CSHOLD = 1), 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.
Affected Conditions / Impacts
Reception of each data bit by the slave is tied to a specific clock edge, thus the late transmission by the master of the first bit of a word may cause the slave to receive the incorrect data, especially if the data setup time for the slave approaches or exceeds one half the shift clock period.
Workaround
Because there is no way to specifically time a write to the transmit FIFO such that it does not occur when the USART state machine changes state, use one of the following workarounds to avoid the risk for data corruption described above: • 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 operations in both the CLKPOL = CLKPHA = 0 and CLKPOL = CLKPHA = 1 modes.
Resolution
There is currently no resolution for this issue.

2.24 USART_E206 — Additional SCLK Pulses Can Be Generated in USART Synchronous Mode

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

2.25 WDOG_E201 – Clear Command is Lost Upon EM2 Entry

Description of Errata
If the device enters EM2 while the clear command is still being synchronized, the watchdog counter may not be cleared as expected.
Affected Conditions / Impacts
If the watchdog counter is not cleared as expected, the device can encounter a watchdog reset.
Workaround
Wait for WDOG_SYNCBUSY_CMD to clear before entering EM2. Note that WDOG can be clocked from one of the low-frequency clock sources and will require additional time to enter EM2 when implementing this workaround.
Resolution
There is currently no resolution for this issue.

2.26 WTIMER_E201 — Continuous Overflow and Underflow Interrupts in Quadrature Counting Mode

Description of Errata

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

Affected Conditions / Impacts

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

Workaround

Short of disabling the relevant interrupts, the simplest workaround is to manually change WTIMER_CNT so that the overflow or underflow condition no longer exists. Insert the following or similar code in the interrupt handler for the timer in question (WTIMER0 in this case) to do this:

```
uint32 intFlags = TIMER_IntGet(WTIMER0);

if((intFlags & WTIMER_IF_OF) && (WTIMER0->CNT == WTIMER0->TOP))
  WTIMER0->CNT = 0;

if((intFlags & WTIMER_IF_UF) && (WTIMER0->CNT == 0x0))
  WTIMER0->CNT = WTIMER0->TOP;
```

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

Resolution

There is currently no resolution for this issue.

3. Resolved Errata Descriptions

This section contains previous errata for EFM32GG11 devices. Several revision A part numbers are available at Engineering Status, which is denoted with revision code X. For the purposes of this document, revision “A/X” refers to both revision A and revision X part numbers.

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

3.1 ADC_E224 – ADC Warm-Up Ready Can Cause IDAC, ACMP, or CSEN to Not Function

Description of Errata
The IDAC, ACMP, or CSEN modules use 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 IDAC, ACMP, or CSEN modules to never finish warming up. The ADC is not affected by this issue and will always be available after it is enabled.
Affected Conditions / Impacts
Systems using the IDAC, ACMP, or CSEN modules in conjunction with the ADC can see intermittent failures where these modules do not operate.
Workaround
To work around this issue, enable the IDAC, ACMP, or CSEN modules before enabling the ADC. This will ensure the handshaking logic between the ADC and other modules functions correctly.
Resolution
This issue is resolved in revision B devices.

3.2 ADC_E225 – Using the ADC in High-Accuracy Bias Mode Will Force All Analog Peripherals to High-Accuracy Bias Mode

Description of Errata
Using the ADC in high-accuracy bias mode (GPBIASACC in ADCn_BIASPROG cleared to 0) forces all other analog peripherals into high-accuracy bias mode. These peripherals will then draw additional current. The data sheet current consumption numbers are current specified with this additional current consumption included. When the updated devices are available, the device data sheet will be updated with the reduced current consumption specifications.
Affected Conditions / Impacts
Systems using the ADC in high-accuracy bias mode may see higher current consumption than expected when other analog peripherals not using high-accuracy bias mode are in use.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.3 ADC_E228 – Limited ADC Sampling Frequency in EM2

Description of Errata
ADC FIFO overflows occur at frequencies that are much lower than the ADC's maximum theoretical sampling rate.
Affected Conditions / Impacts
ADC sampling frequency is reduced in EM2.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.4 BU_E201 — Extra Current from BUVDD to VREGVDD in BUMODE when Pulling Main Supply Low

Description of Errata
While in Backup mode, pulling AVDD/IOVDD/VREGVDD/DVDD to 0 V and keeping BUVDD constant causes extra current from BUVDD to VREGVDD. This current starts occurring when AVDD = 0.6 V and is worst when AVDD = 0 V (~250 µA) and depends on the load on the VREGVDD pin (including the device itself).
Affected Conditions / Impacts
In most cases, even when replacing a battery, the supply should not drop below 0.6 V. If the supply does drop below this threshold, there will be some additional current draw depending on the load on the VREGVDD pin.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.5 BU_E202 — Extra Current from DVDD to GND in BUMODE when Pulling BUVDD Low

Description of Errata
While in Backup mode, pulling BUVDD low (but still above the BOD threshold) and keeping the main supply (AVDD/IOVDD/VREGVDD/DVDD) constant causes extra current from DVDD to GND. For example, with DVDD = 3.79 V, and BUVDD = 2 V, there is a current draw of ~128 µA current from DVDD to ground.
Affected Conditions / Impacts
Systems that enter Backup mode may see some additional current draw from DVDD to GND if BUVDD drops.
Workaround
For systems that do not use the DC-DC converter, there is currently no workaround for this issue. For systems that do use the DC-DC converter, configure the chip to drive DVDD with the DC-DC converter (either BYPASS or regulation mode). When the device enters Backup mode, the DC-DC converter is turned off, leaving DVDD floating and removing the current leakage from DVDD to ground.
Resolution
This issue is resolved in revision B devices.

3.6 CMU_E203 – Peak Detector May Not Trip

Description of Errata
When PEAKDETTHR in HFXOCTRL1 is set to 4 or higher, the peak detector may not trip when the oscillating amplitude is higher than the target, which will cause calibration failure.
Affected Conditions / Impacts
If the device is not calibrated correctly (See CMU_E204), IBTRIMXOCORE will end up at a higher value than expected. Consequently, the oscillating amplitude and current consumption will also be higher than expected.
Workaround
PEAKDETTHR should be configured to be less than 4 to avoid potential peak detection failure. See CMU_E204 for guidance on how to program IBTRIMXOCORE.
Resolution
This issue is resolved in revision B devices.

3.7 CMU_E204 – Initial Oscillator Calibration

Description of Errata
On power-up, a one-time calibration of oscillator amplitude is performed using the bias value specified by IBTRIMXOCORE in the HFXOSTEADYSTATECTRL register. The HFXO may not be able to start or settle at the target amplitude with the default value of IBTRIMXOCORE when crystal ESR is extraordinarily high.
Affected Conditions / Impacts
This typically manifests itself when performing a Pierce oscillator robustness test using a resistor that results in the HFXO seeing a series resistance of 3 or 5 times the maximum ESR specified for the crystal. The artificially inflated ESR may prevent the HFXO from starting or settling at the target amplitude with the default value of IBTRIMXOCORE.
Workaround
Three register fields can be modified to enhance oscillator start-up: 1. Increase the value of the IBTRIMXOCORE field in the HFXOSTEADYSTATECTRL register from the default value of 0x100 to 0x1ff. 2. Increase the value of the STEADYSTATETIMEOUT field in the HFXOTIMEOUTCTRL register from the default value of 0x04 to 0x08. 3. Decrease the value of the PEAKDETTHR field (bits [14:12]) in the previously undocumented HFXOCTRL1 register (at offset 0x28 in the CMU address space) from the default value of 0x04 to 0x02. The following table summarizes the affected register fields on revision A/X and revision B devices:

Table 3.1. CMU HFXO Register Field Defaults

Offset	Register	Bit Field	Default Value	
			Revision A/X	Revision B
0x28	HFXOCTRL1	PEAKDETTHR	0x4	0x2
0x34	HFXOTIMEOUTCTRL	STEADYTIMEOUT	0x4	0x8

Note that a user-specified value for IBTRIMXOCORE is going to be correlated with crystal frequency, ESR, and load capacitance, so its default value is not changing on revision B devices.
Resolution
This issue is resolved in revision B devices.

3.8 CORE_E205 – SRAM Slave (RAM0, RAM1) ECC Related Busfaults Cannot be Blocked for Non-Word Accesses

Description of Errata
SRAM slave (RAM0, RAM1) ECC related busfaults cannot be blocked for non-word accesses
Affected Conditions / Impacts
Blocking of DMEM ECC related bus faults (upon detecting 2-bit errors) by setting `MSC->CTRL.RAMECCERRFAULTEN=0` does not work for non-word (i.e. byte, halfword) accesses. When limiting DMEM access to word accesses, the `MSC->CTRL.RAMECCERRFAULTEN` control functions as it should.
Workaround
There is no workaround for this issue.
Resolution
This issue has been reclassified as MSC_E203.

3.9 EMU_E214 – Device Erase Cannot Occur if Voltage Scaling Level is Too Low

Description of Errata
The device erase logic does not check the Voltage Scale Level prior to attempting a device erase. If using Voltage Scale Level 0 (1 V), the device may not be able to erase the flash. This results in a potentially unlockable device if operating at Voltage Scale Level 0 (1 V).
Affected Conditions / Impacts
It is possible that the flash is only partially erased when performing the operation at Voltage Scale Level 0 (1 V). If this results in the debug lock bit not clearing, a locked part doesn't unlock after the partial erasure (which it is intended to do), and the part remains locked. If subsequent erasures continue to fail, the part would remained locked.
Workaround
The voltage should be set to Voltage Scale Level 2 (1.2 V) before executing the device erase. For systems that don't lock the debug interface, the user can follow the debug recovery procedure to halt the CPU before it has a chance to execute code in software to avoid the code scaling the voltage. The device erase can then be executed at Voltage Scale Level 2 (1.2 V) (the power-on default voltage of the device). For systems that do lock the debug interface, firmware can implement a mechanism whereby it can voltage scale or unlock debug access if its defined authentication method is passed.
Resolution
This issue is resolved in revision B devices.

3.10 GPIO_E202 – GPIO Mode PUSHPULLALT Not supported for High Speed Priority Locations of Alternate Functions

Description of Errata
High speed priority locations of alternate functions are not enabled if GPIO mode is set to PUSHPULLALT.
Affected Conditions / Impacts
High speed priority locations of alternate functions can only be used if GPIO MODEn is set to PUSHPULL. This implies that alternate port controls (DINDISALT, SLEWRATEALT and DRIVESTRENGTHALT) cannot be used for the pins associated with high speed priority locations of alternate functions.
Workaround
There is currently no work around for this issue.
Resolution
This issue is resolved in revision B devices.

3.11 I2C_E202 – Race Condition Between Start Detection and Timeout

Description of Errata
There is a race condition where the Bus Idle Timeout counter may clear the busy status of the I2C bus after a start condition.
Affected Conditions / Impacts
Software may attempt another I2C start if it thinks the bus is idle. This may disrupt the I2C bus. After the Bus Idle Timeout feature has triggered, it will not detect another idle condition.
Workaround
Software can wait for any of the following conditions before starting an I2C transaction: • The received address match interrupt indicates that the I2C bus is busy. Software should serve this transaction and proceed accordingly. Software can ignore the wrong busy status. • The SSTOPIF interrupt flag indicates that the I2C bus has returned to the idle state. • A defined, system-dependent amount of time to wait after bus activity to ensure that the bus is in idle state.
Resolution
This issue is resolved in revision B devices.

3.12 I2C_E203 – I2C Received Data Can be Shifted

Description of Errata
If SDA falls between detection of the start condition and the first rising edge of SCL, the I2C state machine clears the start condition that was just detected, causing the state machine counter to count the rising edge of SCL earlier than it was detected. This causes the received data to be out of sync and the acknowledge phase to occur one SCL clock cycle earlier than expected, thus corrupting the integrity of the I2C bus. There are two ways in which the falling condition on SDA can potentially happen: • In multi-master systems, one master initiates a start condition and then drives SDA high shortly before another master drives SDA low to indicate a start condition. • In a single master system, if SDA is high from the last bit of the previous transaction, the master initiates a start condition and then drives SDA low because the MSB of the new address is low.
Affected Conditions / Impacts
I2C operation in slave mode or multi-master mode.
Workaround
This depends on whether the system is multi- or single-master. There is no workaround for multi-master cases. In a single-master system, the state of SDA may not change unless a new address is being sent, such that the falling condition on SDA would not be observed. Whether or not this is the case is dependent on the implementation of the particular I2C master.
Resolution
This issue is resolved in revision B devices.

3.13 I2C_E205 – Go Idle Bus Idle Timeout Does Not Bring Device to Idle State

Description of Errata
When the I2C is operating as a slave, if the bus idle timeout is active (`I2Cn_CTRL_BITO != 0`) and the go idle on bus timeout feature is enabled (`I2Cn_CTRL_GIBITO = 1`), the bus idle interrupt flag (`I2Cn_IF_BITO`) sets upon timeout, but the receiver does not enter the idle state.
Affected Conditions / Impacts
The I2C receiver needs to detect a START condition to recover from the bus idle timeout state. If there is other, undefined activity on the bus after the timeout, the receiver will not recover as expected.
Workaround
The `I2Cn_CTRL_EN` bit can be toggled from 1 to 0 and back to 1 again to resume normal operation. Alternatively, a START condition issued by any other master on the bus (including the EFM32/EFR32 device) will reset the receiver and return it to normal operation.
Resolution
This issue is resolved in revision B devices.

3.14 LCD_E201 — LCD Boost Mode Does Not Work

Description of Errata
It is not recommended to use the LCD boost mode on affected revisions of the device.
Affected Conditions / Impacts
Systems should not use the LCD boost mode feature.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.15 MSC_E203 – ECC-Related Bus Faults Cannot be Blocked for Non-Word Accesses

Description of Errata
Bus faults for non-word accesses cannot be blocked when the corresponding SRAM has ECC enabled (via `MSC->ECCCTRL`).
Affected Conditions / Impacts
Blocking SRAM ECC-related bus faults (upon detecting 2-bit errors) by writing `MSC->CTRL.RAMECCERRFAULTEN=0` does not work for non-word (i.e. byte and halfword) accesses. When SRAM ECC is enabled, `MSC->CTRL.RAMECCERRFAULTEN=0` works as expected for word-only accesses.
Workaround
There is no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.16 RMU_E203 – AVDD Ramp Issue

Description of Errata
The device may not properly start during power-on or restart when a voltage droop (brown out) occurs on AVDD. The failure is intermittent. For example configurations and waveforms that are more likely to result in this issue, see the following Knowledge Base article: http://community.silabs.com/t5/32-bit-MCU-Knowledge-Base/RMU-E203-AVDD-Ramp-Issue/ta-p/197340 To detect this failure state, place a GPIO toggle at the beginning of `main()` in the device firmware. When this failure occurs, the pin will not be toggling as expected, as the device is not executing any code.
Affected Conditions / Impacts
Systems may intermittently see the device fail to start, reset, or respond. The current draw of the device in this state is ~100 µA and DECOUPLE will be fully powered (~1.2 V). The device will not execute any code in this state.
Workaround

This issue can be resolved with a hardware workaround where an external circuit holds the reset pin low during power-on or brown out until AVDD reaches 1.8 V. For brown out, the reset pin must be configured to hard reset mode. This can be accomplished as part of the firmware image programmed to the device (lock bits area) or using the following code:
 	
`// Clears the CLW0 bit to enable Hard reset`
`void enable_hardreset()`
`{`
`  uint32_t value;`
`  uint32_t newvalue;`
`  value = *(uint32_t *)0xFE041E8;`
`  newvalue = value & ~(1 << 2);`
`  MSC_WriteWord((uint32_t *)0xFE041E8, &newvalue, 4);`
`}`
 	
There is currently no software workaround for all potential failure mechanisms. The software workaround included in the Knowledge Base article will prevent failure in some scenarios. See the Knowledge Base article for more information:
 	
http://community.silabs.com/t5/32-bit-MCU-Knowledge-Base/RMU-E203-AVDD-Ramp-Issue/ta-p/197340

Resolution
This issue is resolved in revision B devices.

3.17 RTCC_E205 – Wrap Event Can Be Missed

Description of Errata
The RTCC main counter can miss a CC1 wrap event (CCV1TOP bitfield in the RTCC_CTRL register set to 1) if one of the following registers are written in the same cycle as the wrap event: RTCC_CTRL, RTCC_CNT, RTCC_TIME, RTCC_DATE, RTCC_PRECNT, RTCC_IFC, RTCC_IFS, RTCC_CCx_CCV, RTCC_CCx_CTRL, RTCC_CCx_TIME, RTCC_CCx_DATE, RTCC_CMD, RTCC_RETx_REG.
Affected Conditions / Impacts
Systems using the CC1 wrap event feature may miss events if an affected register is written immediately before a wrap occurs.
Workaround
There are two workarounds to this issue: • Do not use the CC1 wrap event feature (CCV1TOP in RTCC_CTRL should be cleared to 0). • Alternatively, do not write to any of the affected registers when the counter is about to wrap. This means that firmware must check that RTCC_CNT is not close to RTCC_CC1_CCV before writing the register.
Resolution
This issue is resolved in revision B devices.

3.18 SDIO_E201 — Internal Pull-up Resistors not Enabled

Description of Errata
Internal on-chip pull-up resistors on SDIO CMD/DAT lines do not get enabled even though corresponding GPIO pin modes are configured as PUSHPULL/PUSHPULLALT, GPIO_Px_DOUT of the corresponding pins are set to '1' and involved pins are enabled in SDIO->ROUTEPEN.
Affected Conditions / Impacts
Without pull-up resistors on SDIO CMD/DAT lines, SDIO will not function correctly.
Workaround
The hardware workaround is to connect external pull-up resistors of 10~100k ohm to SDIO CMD/DAT lines.
Resolution
This issue is resolved in revision B devices.

3.19 USB_E201 — USB PHY Short Circuit Failure

Description of Errata
The USB PHY internal terminating resistors cannot withstand a continuous short circuit between D+ and ground or D- and ground as defined by the USB specification for an extended period of time. These types of short circuits might occur due to a damaged or defective USB cable. When this failure occurs, the USB interface suffers physical damage. As a Device with this failure condition, the USB interface will stall, disconnect, or no longer enumerate as a Device. As a Host, the USB interface will fail enumeration or continuously send start-of-frame packets.
Affected Conditions / Impacts
USB peripheral on affected revisions will still function and can be used for development, but it is not recommended to use devices affected by this issue for production.
Workaround
There is currently no workaround for this issue.
Resolution
This issue is resolved in revision B devices.

3.20 USB_E202 — USB Regulator Does Not Read Tuning Register R5VOUTLEVEL After Power Loss and Restore

Description of Errata
The output voltage of the 5V sub-system does not follow the programmed value in EMU->R5VOUTLEVEL.OUTLEVEL after the 5V sub-system power loss (i.e. both VBUS and VREGI are disconnected) and subsequent power restoration (i.e. one of the inputs VBUS and VREGI is connected to power).
Affected Conditions / Impacts
When the 5V sub-system power loss (i.e. both VBUS and VREGI are disconnected) and subsequent power restoration (i.e. one of the inputs VBUS and VREGI is connected to power) happens, EMU->R5VOUTLEVEL.OUTLEVEL keeps showing the value which was set by software before the power loss but the actual voltage at the 5V regulator output VREGO gets reset to ~2.4V and stays at that voltage level.
Workaround
To workaround this issue, software should:1. Enable R5VREADY interrupt by setting EMU->IEN.R5VREADY to '1'.2. Each time EMU->IF.R5VREADY interrupt is detected, wait for EMU->R5VSYNC.OUTLEVELBUSY to go low and write desired value to EMU->R5VOUTLEVEL.OUTLEVEL bitfield inside corresponding interrupt handler.
Resolution
This issue is resolved in revision B devices.

3.21 VDAC_E201 — VDAC Output Drives All APORT Buses Simultaneously

Description of Errata
When VDACn_OPAx_OUT.APORTOUTEN is set, the VDACn/OPAx will drive all its connected APORT buses (BUSAY, BUSBY, BUSCY, and BUSDY) instead of only the APORT bus selected via APORTOUTSEL. However, the VDACn/OPAx APORT request signals do correspond to the programmed APORTOUTSEL value and therefore, the APORT conflict registers (OPAxAPORTCONFLICT in VDACn_STATUS, OPAxAPORTCONFLICTIF in VDACn_IF, and VDACn_APORTCONFLICT) will not reflect potential conflicts on the erroneously driven APORT buses. If any other peripherals (or other VDAC channel or other OPAMP) are using either of the above APORT buses at the same time, this can lead to contention on the APORT bus and possible high current consumption.
Affected Conditions / Impacts
Systems attempting to use multiple APORT buses and the VDAC may see contention.
Workaround
If none of the other APORT clients (e.g., ADC, ACMP, OPAx input muxes, etc.) use BUSAY, BUSBY, BUSCY, and BUSDY, then no problem exists and the potential simultaneous driving of these buses by VDACn/OPAx can be ignored. Alternatively, the VDACn/OPAx can be configured to use direct connections of its main or alternative outputs to certain pins, thereby bypassing the APORT. Direct output connections can be enabled by programming MAINOUTEN=1 and/or ALTOUTEN=1 (while keeping APORTOUTEN=0) in the VDACn_OPAx_OUT register. The device data sheet lists the available main output and alternative output connections to pins per VDAC output or OPAMP.
Resolution
This issue is resolved in revision B devices.

3.22 VDAC_E202 — PRS Outputs Not Generated when Interrupt Flag is Set

Description of Errata
The conversion done (CD) PRS outputs from the DAC are tied to the interrupt flags. As long as the interrupt flag is set, no PRS output will be generated. When the first conversion done (CD) event occurs, the VDAC will set the interrupt flag and generate one PRS pulse. As long as the interrupt flag is set, any new conversion done events will not generate a new PRS pulse. After software clears the flag, the next conversion done event will generate a PRS pulse. Clearing the interrupt flag itself will not generate a pulse. Any CD event that occurs while the flag is set will be ignored.
Affected Conditions / Impacts
Systems attempting to use the DAC PRS outputs should ensure the interrupt flags are cleared.
Workaround
Firmware should clear the conversion done flag immediately after entering the interrupt service routine. This will allow the next conversion done event to generate a PRS pulse.
Resolution
This issue is resolved in revision B devices.

4. Revision History

Revision 0.6

September, 2020

- Added I2C_E207, USART_E206 and WDOG_E201.

Revision 0.5

April 2020

- Added EMU_E220, LES_E201, TIMER_E202, USART_E204, USART_E205 and WTIMER_E201.
- Wording clarified for I2C_E204, and I2C_E205
- Migrated to new errata document format

Revision 0.4

August, 2018

- Updated for device revision B.
- ADC_E224, ADC_E225, ADC_E228, BU_E201, BU_E202, CMU_E203, CMU_E204, CORE_E205/MSC_E203, EMU_E214, GPIO_E202, I2C_E202, I2C_E203, , LCD_E201, RMU_E203, RTCC_E205, SDIO_E201, USB_E201, USB_E202, VDAC_E201, and VDAC_E202 resolved and moved to .
- Added ADC_E228, CSEN_E201, CSEN_E202, and I2C_E206.
- CORE_E203, CORE_E204, CORE_E205, and CORE_E206 reclassified as MSC_E201, MSC_E202, MSC_E203, and MSC_E204 with clarified wording.
- Wording of CMU_E204 clarified, specifically, with a more thorough description of the impact and workaround.
- RTCC_E203 has never been present on EFM32GG11; its mention below is incorrect and does not apply.

Revision 0.3

March, 2018

- Updated for device revision A.
- Updated the workaround in RMU_E202.
- Added CMU_E203, CMU_E204, CORE_E203, CORE_E204, CORE_E205, CORE_E206, EMU_E217, GPIO_E202, I2C_E202, I2C_E203, , , and SDIO_E201.

Revision 0.2

July, 2017

- Added ADC_E224, ADC_E225, BU_E201, BU_E202, CUR_E204, DBG_E204, RMU_E202, RMU_E203, RTCC_E203, and USB_E201.
- Updated the LCD_E201 resolution.
- Changed revision A to revision A/X.

Revision 0.1

February, 2017

- Initial release.

Simplicity Studio

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

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

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

http://www.silabs.com