

# EFM<sup>®</sup>32

*... the world's most energy friendly microcontrollers*

## EFM32G200 Errata History

*F64/F32/F16*



This document describes known errata for all revisions of EFM32G200 devices.

# 1 Errata History

## 1.1 Errata Overview

Table 1.1 (p. 2) shows which erratum is applicable for each revision. The device datasheet explains how to identify chip revision, either from package marking or electronically.

In addition to the errata noted below, the errata for the ARM Cortex-M3 r2p0 ([www.arm.com](http://www.arm.com)) also applies to all revisions of this device.

**Table 1.1. Errata Overview**

Erratum ID	rev C	rev B	rev A
ACMP1		X	
ADC1			X
ADC2			X
ADC4		X	X
ADC5			X
ADC6		X	X
ADC8		X	
ADC10		X	
ADC11		X	X
ADC12		X	X
ADC13		X	X
ADC14		X	X
ADC15	X	X	X
BOD1			X
CMU1			X
CMU2			X
CMU3			X

Erratum ID	rev C	rev B	rev A
CMU4		X	X
CMU5			X
CMU6	X	X	X
CMU7		X	X
CMU8	X	X	X
CMU9	X	X	X
DAC1		X	X
DAC2		X	X
DAC3		X	X
DAC4		X	X
DAC5		X	X
DAC7		X	X
DAC8		X	X
EMU1		X	X
EMU2			X
EMU3	X	X	
EMU4	X	X	X
EMU5	X	X	X
EMU6	X	X	X
HFRCO1			X
I2C1			X
I2C2		X	X
LETIMER1		X	X
LEUART1			X

Erratum ID	rev C	rev B	rev A
LEUART2		X	X
LEUART3		X	
LFRCO1		X	X
LFXO2		X	X
PCNT1		X	X
RTC1	X	X	X
TIMER1		X	X
USART1	X	X	X
USART2		X	X
USART3		X	X
USART4		X	X
USART5		X	X
USART6		X	X
USART7		X	X
USART8		X	X
USART9		X	X
USART10		X	X
USART11		X	X
VCMP1		X	
VCMP2		X	X
WDOG1			X
WDOG2	X	X	X
WDOG3	X	X	X

## 1.2 EFM32G200 Errata Descriptions

**Table 1.2. EFM32G200 Errata Descriptions**

ID	Title/Problem	Effect	Fix/Workaround
ACMP1	<b>ACMP Mode</b> The ACMP only works in the low power reference mode.	The ACMP only works in the low power reference mode. When the low power reference mode is disabled (by not setting the LPREF bit in ACMPn_INPUTSEL), the ACMP does not work and its output is always 1.	When using the ACMP, put it in its low power reference mode by setting the LPREF bit in ACMPn_INPUTSEL (which is the default setting). In this mode, the power consumption in the reference buffer (VDD and bandgap) is lowered at the cost of accuracy.
ADC1	<b>ADC Temperature Sensor</b> The temperature sensor in the ADC does not work.	The temperature values read when sampling the temperature sensor in the ADC are not correct.	Do not use the ADC temperature sensor.
ADC2	<b>ADC SCANGAIN</b> SCANGAIN in ADCn_CAL affects the gain setting for single conversions.	When SCANGAIN and SINGLEGAIN in ADCn_CAL have different values, single conversions will be affected by the SCANGAIN value.	Configure SCANGAIN and SINGLEGAIN in ADCn_CAL to the same value. This requires the same reference to be used for both single and scan conversions.
ADC4	<b>ADC 1 Msample/s</b> 1 Msample/s is not achieved for default ADC bias settings.	At default ADC bias settings the ADC conversion results are wrong when running the ADC_CLK at 13 MHz, which is required to reach the 1 Msample/s performance. Under typical conditions wrong conversions have been observed for ADC_CLK speeds of 8 MHz and higher.	Increase the ADC performance by programming increased ADC bias, for example by using value 0xF0F for register ADCn_BIASPROG.
ADC5	<b>ADC Output</b> The ADC does not always sample a voltage at (or close to) the middle of its range correctly (e.g. when sampling 1.25V when using the 2.5V internal reference).	When the ADC is sampling voltages at (or close to) the middle of its range, the ADC output code can be off by a large value (e.g. returning value 1023 or 3072 instead of the expected value of 2048). This effect happens for all ADC reference selections.	Perform multiple (e.g. 3) ADC measurements for each ADC sample required and use the median value. Do not average the ADC results, throw away the 1023 or 3072 sample instead.
ADC6	<b>ADC Reference Settling</b> The ADC internal references, i.e. 1V25, 2V5 and VDD have a settling time of about 500 us.	Measurements done using one of the internal references will not be correct before the reference has settled. This effect appears when switching between references and when the references have been off between samples.	When using the internal references, set WARMUPMODE=3 in ADCn_CTRL and wait until the references have settled before taking the first sample.
ADC8	<b>ADC Temperature Sensor</b> does not work out of reset. The temperature sensor in the ADC does not work out of reset.	Temperature measurements done using the temperature sensor in the ADC will be wrong without the fix described below.	To enable the temperature sensor, set *0x400C6018 = 0x6. This fix can not be used at the same time as the fix for <i>CMU4</i>

ID	Title/Problem	Effect	Fix/Workaround
ADC10	<p><b>ADC VDD Reference Gives Half Resolution</b></p> <p>When using the internal VDD reference, the ADC resolution is reduced to 11 bits.</p>	Measurements done with the VDD reference will appear to have been divided by 2.	Double each measurement to give the measurements the correct amplitude, sacrificing one bit of resolution. This workaround will not be compatible with devices of later revisions where this erratum is corrected.
ADC11	<p><b>ADC References Doubled</b></p> <p>In single-ended mode, external and differential references are doubled internally.</p>	The reference doubling results in a decrease of the ADC resolution by one bit. As an example, when using an external reference of 1 V in single ended mode, a signal with values from 0 V to 1 V will result in adc codes from 0 to 2047 instead of the full 0 to 4095.	A temporary fix for the external references is to halve the reference voltage. This will give full resolution, but will not be compatible with devices of later revisions where this erratum has been corrected.
ADC12	<p><b>ADC Accuracy</b></p> <p>The ADC does not meet the specified accuracy of 11.7 effective bits. The ADC is monotonic and although within specification, the hit frequency for some codes such as 3071 and 3072 is such that DNL is distinctly different from the average DNL (but there are no missing codes). The gain of the 2XVDD reference is larger than 1.0.</p>	The ADC accuracy may vary depending on the ADC configuration and may in some cases be down to 10 effective bits. The DNL/INL anomalies will cause low level spurs in the output spectra. The gain error for the 2XVDD reference is over 20 LSBs and gain for this reference cannot be calibrated.	ADC accuracy can be increased by using hardware oversampling to increase resolution and/or by increasing the ADC bias current. The ADC oversampling rate can be programmed in the OVSSEL field of the ADCn_CTRL register; the oversampling can be enabled by using the OVS value in the RES field of the ADCn_SINGLECTRL or ADCn_SCANCTRL register. The ADC bias current can be programmed via the ADCn_BIASPROG register.
ADC13	<p><b>ADC Variability</b></p> <p>The PSRR, CMRR and variability over temperature for the ADC do not meet the specification.</p>	With a DC input voltage, the ADC output will vary depending on changes in VDD level, input common mode level, and temperature. The temperature related variation particularly applies to the 5VDIFF reference. For a DC level input, variability of the ADC output code over VDD is 2, 32, 7 LSBs for the VDD, 5VDIFF, and external references respectively. With external references, there is about 8 LSBs variation in ADC output code depending on the input common mode level.	With external references, adjust the common mode level of the differential external reference to a lower level. For the external differential references, performance is better with the lower external reference negative input level at 0V.
ADC14	<p><b>Incorrect ADC Calibration Register Reset Value</b></p> <p>The ADC calibration register (ADCn_CAL) are not updated with calibration values from production test during reset.</p>	The ADC will be uncalibrated out of reset and can show offset and gain error outside of specification.	Copy the gain and offset values for the selected ADC reference from the Device Information (DI) page in flash to the corresponding ADCn_CAL register fields before starting a conversion.
ADC15	<p><b>Incorrect ADC Temperature Sensor Calibration Data</b></p>	Devices with PROD_REV values of 9 or 10 does not have correct ADC temperature sensor reading stored in the ADC0_TEMP_0_READ_1V25 register of the Device Informa-	Instead of using the value stored in the Device Information table, use ADC0_TEMP_0_READ_1V25 = 0x906 and CAL_TEMP_0 = 0x19 when calculating the temperature.

ID	Title/Problem	Effect	Fix/Workaround
	The ADC temperature sensor calibration value stored in the DI page is not correct.	tion Page, and using this value for calculating the temperature will yield wrong results.	These values are gathered from production data, and will give an accuracy where 3x the standard deviation correspond to 5.2 degrees celsius.
BOD1	<b>BOD Threshold</b> The Brown-Out Detector (BOD) threshold voltage is calibrated to a too high value.	The high BOD threshold voltage may create sporadic BOD resets while the EFM32 is running in Energy Mode 2. Also, the BOD may cause a reset at higher voltages than specified as the threshold voltage in the Electrical Characteristics.	Download Development Kit Board Support Library and Example Code (rev 1.1.1 or later) and include <code>efm32_chip.h</code> in your project. In the start of the application code, call <code>void CHIP_Init(void);</code> . This procedure will re-program the device to a safe BOD threshold.
CMU1	<b>Peripheral Clocks Active In EM2/EM3 During Debug</b> When a debug session has been active since the last reset, EM1 is entered when trying to enter EM2 or EM3.	The device cannot enter EM2/3 if a debug session has been entered since the last reset. When attempting to go to either EM2 or EM3, the system goes to EM1, and the peripheral clocks, which should have been turned off in EM2/EM3 keep going. This is only an issue when debugging a system.	If the debugger is running, clear <code>HFPERCLKEN</code> in <code>CMU_HFPERCLKDIV</code> before going to EM2/EM3 and set it when going back to EM0.
CMU2	<b>LFRCO/HFRCO Frequency Change during EM2/3</b> RCO oscillator frequency can become unstable on transitions between EM2/3 and EM0.	When switching between EM0 and EM2/3, the following events can happen occasionally: <ul style="list-style-type: none"> <li>• The frequency of LFRCO becomes off by up to 14%</li> <li>• The frequency of HFRCO becomes off by up to 6%</li> </ul> The frequency will be off for a shorter or longer period.	Make this line of code part of your startup code, typically in the start of <code>main()</code> : <code>*(volatile unsigned int*) 0x400C600CUL = 0x00020100;</code> As a result of this workaround, the current consumption in EM2/3 will go up by 450 nA. This fix is not compatible with devices of later revisions where this erratum has been corrected.
CMU3	<b>Wrong RCO Frequency</b> The HFRCO, AUCHFRCO and LFRCO oscillators has wrong frequency when running with default settings.	The oscillator frequency has not been programmed with correct calibration values, and the frequencies are not within the expected frequency ranges.	The oscillator frequency can be calibrated in the Clock Management Unit, which is described in the CMU chapter of the EFM32G Reference Manual.
CMU4	<b>Energy Mode Transitions Cause HFRCO Overshoot</b> Transitions between energy modes may cause an overshoot in the HFRCO frequency	When switching between energy modes, there is a slight chance that HFRCO will temporarily overshoot its configured frequency and in some cases cause a BOD reset. This overshoot may be up to 50%, and may take the system out of its allowed operating conditions by having a system clock higher than 32 MHz. Note that when the MSC is configured to use zero waitstates when accessing flash, the maximum core frequency is 16 MHz.	To fix issue, set <code>*0x400C6018 = 0xC201</code> . This gives better HFRCO stability at the cost of an increased current consumption in EM0 and EM1 of 10 uA. This fix is not compatible with the fix for <code>ADC8</code> . This fix is not compatible with devices of later revisions where this erratum has been corrected.
CMU5	<b>AUXHFRCO Active in EM2/EM3</b> AUXHFRCO is not disabled automatically when entering EM2/EM3.	If AUXHFRCO is running while in EM2/EM3, and the <code>EMVREG</code> bit in <code>EMU_CTRL</code> is cleared. This may result in an unstable system.	Disable AUXHFRCO by writing a 1 to <code>AUXHFRCODIS</code> in <code>CMU_OSCENCMD</code> , before going to EM2/EM3. When waking up, enable the AUXHFRCO again if needed by writing a 1 to <code>AUXHFRCOEN</code> in <code>CMU_OSCENCMD</code> .

ID	Title/Problem	Effect	Fix/Workaround
CMU6	<b>LFXO Digital External Mode</b>  LFXO ready flags are never set when LFXO is configured in Digital External Clock mode.	When LFXOMODE in CMU_CTRL is set to DIGEXTCLK the LFXORDY flag in CMU_STATUS and CMU_IF will not be set when the number of cycles set in LFXOTIMEOUT in CMU_CTRL has elapsed. Thus polling of this flag will not work. However, the clock propagates as normal. It is only the flag that is not set.	To detect that the clock has propagated through the ripple counter, write to any Asynchronous Register in any Low Energy peripheral and wait for SYNCBUSY for that register field to go low. Remember to enable the LE core clock and the clock for the LE peripheral you choose. For example, write 0xA5 to RTC_COMP0 and wait for COMP0 in RTC_SYNCBUSY to go low.
CMU7	<b>Disabled Low Frequency Clocks</b>  Disabled Low Frequency Clocks tick once every second.	The LFA and LFB clocks tick with a frequency equal to the source clock divided by 32768 when the corresponding enable bit in CMU_LFACLKEN0/CMU_LFBCLKEN0 is not set. So, for example, if the RTC is enabled and RTC in CMU_LFACLKEN0 is 0, the RTC will tick once every second. Notice that it is not possible to write to a Low Energy Peripheral when the clock to the peripheral is not enabled. Thus, the effects of this issue is only seen when the clock to an active Low Energy Peripheral is turned off.	Disable the Low Energy Peripheral before disabling the clock to the Low Energy Peripheral to avoid unexpected behaviour.
CMU8	<b>LFxCLKEN write</b>  First write to LFxCLKEN can be missed.	For devices with PROD_REV < 15, enabling the clock for LFA/LFB after reset and then immediately writing LFACLKEN/LFBCLKEN, may cause the write to miss its effect.	For devices with PROD_REV < 15, make sure CMU_SYNCBUSY is not set before writing LFACLKEN/LFBCLKEN. Can temporarily switch to HFCORECLKLEDIV2 to speed up clearing synchbusy.
CMU9	<b>LFXO configuration incorrect</b>  LFXO configuration incorrect.	For devices with PROD_REV < 15, the default value for LFXOBOOST in CMU_CTRL are wrong.	On devices with PROD_REV < 15, change LFXOBOOST to 0.
DAC1	<b>DAC Sample-Hold</b>  When the DAC is in sample/hold mode, the DAC output is not correctly held, but drifts faster than specified.	The DAC output starts drifting in the order of 10 mV/us after two DAC clock cycles.	Put the DAC in continuous mode by setting the CONVMODE field in the DACn_CTRL register to CONTINUOUS. The DAC channels will then drive their outputs continuously with the data in the DACn_CHxDATA registers. This mode will maintain the output voltage and refresh is therefore not needed. As the DAC cores are not turned off between samples in continuous mode, the power consumption is somewhat increased compared to sample/hold mode.
DAC2	<b>DAC Enabling</b>  DAC conversions done closely after enabling the DAC channel are incorrect.	The DAC output takes about 600 us (under typical conditions) to settle after a DAC channel has been enabled via setting field CH0EN in DACn_CH0CTRL (or CH1EN in DACn_CH1CTRL for channel 1). The effect is most visible for the 1.25V and 2.5V internal references.	After enabling a DAC channel, wait 600 us before programming the channel data (via DACn_CH0DATA, DACn_CH1DATA, or DACn_COMBDATA).
DAC3	<b>DAC Ringing</b>	When applying large steps to the DAC, ringing can be observed on the output (up to 100 mV peak-to-peak depending on load). The oscillations will last no longer than 1 us	Filter on output or don't apply large steps.

ID	Title/Problem	Effect	Fix/Workaround
	Ringing effects can also be observed on the DAC output.		
DAC4	<b>DAC Sample-Hold/Sample-Off</b> When a sample/refresh is done while in the sample-hold and sample-off modes, a dip in the DAC output voltage occurs.	The voltage-dip causes noise	Use the DAC in continuous mode by setting the CONVMODE field in the DACn_CTRL register to CONTINUOUS.
DAC5	<b>DAC startup</b> When enabling a DAC channel, there may be a transient on the channel output. This transient may be up to 800 mV and last about 1us on an unloaded DAC.	The DAC output is incorrect a short while after enabling the DAC channel.	To prevent the transient on the DAC output, make sure the DAC output is disabled by clearing OUTMODE in DACn_CTRL when enabling a DAC channel.
DAC7	<b>DAC Accuracy</b> The DAC does not meet the specified accuracy of 11.5 effective bits. The DAC linearity does not meet the specification for all input codes.	The DAC accuracy may vary depending on the DAC configuration and may in some cases be down to 9 effective bits (which is primarily caused by the SFDR without external filtering being limited to about 55 dB in case of the 1V25 reference). For some input codes, e.g. 1024, 2048, 3072, an increase/decrease of one input code will result in an output change equal to up to 3 input codes.	No workaround.
DAC8	<b>Incorrect DAC Calibration Register Reset Value</b> The DAC calibration register (DACn_CAL) are not updated with calibration values from production test during reset.	The DAC will be uncalibrated out of reset and can show offset and gain error outside of specification.	Copy the gain and offset values for the selected DAC reference from the Device Information (DI) page in flash to the corresponding DACn_CAL register fields before starting a conversion.
EMU1	<b>EM Transition Brown Out</b> In rare situations, transitioning between energy modes may cause a Brown Out (BO)	When switching between energy modes, there is a chance that the system will experience a BO. In that case, the system will be reset and the error condition can be detected by reading the RMU_RSTCAUSE register, which will then show that an internal BO was the reason for the reset.	To fix issue, set *0x400C6020  = 0x6000. This prevents the BO, but results in an increase of current consumption in EM0 and EM1 by about 4%. This fix is not compatible with devices of later revisions where this erratum has been corrected.
EMU2	<b>DMA Clock EM2/EM3</b> When the DMA clock is disabled, the EFM32 is not able to go to Energy Modes 2 or 3.	The DMA will prevent the system to go to EM2/EM3 as long as the DMA clock is disabled.	Make sure the DMA clock is enabled when going to EM2/EM3. The DMA clock can be enabled in the CMU.

ID	Title/Problem	Effect	Fix/Workaround
EMU3	<p><b>EM4 current</b></p> <p>In EM4 the device may consume 700nA instead of 20nA.</p>	<p>If EM4 is issued within a 10µs-12µs window after the 1kHz RC oscillator rising edge transition the device will permanently consume 700nA.</p>	<p>There two possible workarounds for this issue.</p> <p>The first workaround is using the WDOG to identify the rising edge transition and add a delay before going into EM4. Write on the WDOG_CTRL register (for instance WDOG-&gt;CTRL=WDOG_CTRL_CLKSEL_ULFRCO) and wait for the SYNCBUSY to be released. The release of the SYNCBUSY happens on a rising edge transition of the 1Khz clock. After that insert a number of __NOP(); to cause a delay of 20µs (12µs plus margin). The number of __NOP(); will depend on the processor frequency. After the delay EM4 can be entered safely. Note: to implement this workaround the WDOG can not be locked, otherwise the registers will not be written.</p> <p>The second workaround is by outputting the ULFRCO on a pin (CMU_CLK0) using CMU_CTRL and CMU_ROUTE registers. That pin should then be configured as push pull with interrupt enable on rising edge, so the device can go to EM2 while it waits for the ULFRCO rising edge transition. When the interrupt occurs clear it and add a number of __NOP(); before entering EM4, as described in the first workaround. Note: the pin used to output the ULFRCO should be driven by an external source.</p>
EMU4	<p><b>Sequencing of Analog and Digital Power</b></p> <p>Power-on Reset might fail if power is applied to IOVDD_x or VDD_DREG before AVDD_x</p>	<p>The device might lock up if power is applied to IOVDD_x or VDD_DREG pins before AVDD_x pins during power up. This lock-up state can be exited by removing power to the device followed by a power up sequence according to what is described in the workaround.</p>	<p>Make sure that the power on the AVDD_x pins ramp earlier or at the same time as the power on IOVDD_x and VDD_DREG during power up. Practical schematic recommendations for this workaround are given in the EFM32 Application Note "AN0002 Hardware Design Considerations".</p>
EMU5	<p><b>Debug unavailable during DMA processing from EM2</b></p> <p>The debugger cannot access the system processing DMA request from EM2.</p>	<p>DMA requests from the LEUART can trigger a DMA operation from EM2. While waiting for the DMA to fetch data from the respective peripheral, the debugger cannot access the system. If such a DMA request is not handled by the DMA controller, the system will keep waiting for it while denying debug access.</p>	<p>Make sure DMA requests triggered from EM2 are handled.</p>
EMU6	<p><b>SWO line pulled low in EM2</b></p> <p>SWO pulled low in EM2.</p>	<p>The SWO line is pulled low in EM2. This can be interpreted as garbage by an outside observer.</p>	<p>Before entering EM2, disable pin-enable by clearing SWOPEN in GPIO_ROUTE, and set SWO pin output high. After exiting EM2, the SWO pin should be re-enabled.</p>
HFRCO1	<p><b>HFRCO Calibration</b></p>	<p>The HFRCO frequency will be outside the expected frequency range when applying the calibration value from the device information page.</p>	<p>The oscillator frequency can be calibrated in the Clock Management Unit, which is described in the CMU chapter of the EFM32G Reference Manual.</p>

ID	Title/Problem	Effect	Fix/Workaround
	The Device Information page does not contain calibration values for the 1 MHz, 7 MHz, 11 MHz and 21 MHz frequency band.		
I2C1	<b>I<sup>2</sup>C RX Overflow</b>  If reception of a byte by the RX shift register is completed while there is still a byte in the RX buffer, the byte in the shift register is silently discarded.	If a received byte is acknowledged before it is read out of the RXDATA, all new bytes received before the read operation are discarded. A new byte is not discarded if the read operation is performed before the new byte is fully received.	Make sure to read the RX buffer before the reception of the next byte completes. One way to ensure this is to always read a received byte before acknowledging it.
I2C2	<b>I<sup>2</sup>C Disabled After EM2/EM3</b>  If the USART0 clock is disabled, the I2C will not work when waking up from EM2/EM3	When waking up from EM2/EM3 with the USART0 clock disabled, the I2C module will be in a disabled state until the USART0 clock has been enabled again.	Make sure the USART0 clock is enabled when using the I2C. Alternatively, enable the USART0 clock for a short while after exiting EM2/EM3.
LETIMER1	<b>Buffered Top Value</b>  CNT is updated with LETIMERn_COMP0 instead of LETIMERn_COMP1 when REPO goes to zero.	In Buffered Mode, both CNT and LETIMERn_COMP0 shall be updated with new LETIMERn_COMP1 value when REPO goes to zero. Instead, CNT gets the previous LETIMERn_COMP0 value, while LETIMERn_COMP0 is correctly updated. Thus, the first period in the next repeat-sequence is the previous top value (LETIMERn_COMP0) and not the new top value (LETIMERn_COMP1) as one would expect.	No workaround.
LEUART1	<b>LEUART + DMA</b>  EM2 cannot be entered when transmitting the last byte using LEUART and DMA.	When using the LEUART with DMA in EM2, TXDMAWU in LEUARTn_CTRL must be cleared when the DMA has no more data to transmit. Otherwise the LEUART will keep the system awake waiting for data from the DMA. The way to do this is to clear TXDMAWU in the DMA DONE interrupt for the channel feeding the LEUART with data. In this device revision, the DMA DONE interrupt will not trigger a wakeup from EM2, and software will thus not be able to clear TXDMAWU immediately when a transmission has been completed, causing the system to be awake more than necessary.	Use the TX complete interrupt (TXC) in the leuart to clear TXDMAWU, or clear TXDMAWU in the DMA DONE interrupt and make sure the TXC interrupt is triggered. The system will then be awake with a higher power consumption while the last byte is transmitted by the LEUART, but will be allowed to go back to EM2 once TXDMAWU has been cleared.
LEUART2	<b>LEUART Baudrate</b>  The LEUART baudrate generator should have an integral baudrate error no larger than +- 0.5 clock cycles. However, the error may be up to 1 clock cycle.	For some baudrate settings, the jitter will be higher than +- 0.5 clock cycles, and the average baudrate value will also not be as expected. For ~9600 b/s @ 32.768 kHz oscillator frequency, CLKDIV=0x268 and CLKDIV=0x270 for instance gives significantly different baudrates. For lower baudrates there should be no problem when using a 32 kHz oscillator.	For 9600 b/s @ 32.768 kHz, use CLKDIV=0x270, which gives a baudrate of ~9534 b/s. This workaround will not be compatible with devices of later revisions where this erratum has been corrected.

ID	Title/Problem	Effect	Fix/Workaround
LEUART3	<b>LEUART RXOF</b> In rare situations RX overflow interrupt can be set despite RX data is not lost.	RXOF is set one cycle too early with the consequence that if RX buffer is read in the same internal clock cycle as RXOF is set, the frame causing RX data is actually loaded into the RX buffer. Thus, the overflow interrupt does not guarantee that data has been lost.	Consider RXOF as an indication that an overflow might have occurred
LFRCO1	<b>LFRCO Frequency</b> The frequency of the LFRCO changes with up to 30% between EM0/EM1 and EM2/EM3.	Calibrating the LFRCO for a given frequency in EM0/EM1 will not guarantee the same frequency in EM2/EM3.	Use LFXO if an accurate clock frequency is important.
LFXO2	<b>LFXO Temperature Sensitivity</b> LFXO may not start.	On some devices the LFXO may not start, on others the LFXO may stop when temperature approaches -40C. In the latter case the LFXO will start up again when the temperature rises.	Place a resistor in parallel with the LFXO crystal. The resistor should be approximately 50 MOhm.
PCNT1	<b>PCNT0 TOP Register</b> The reset value of the TOP register of PCNT0 is incorrect.	When counting downwards, the pulse counter underflows to an incorrect value. When counting upwards, no interrupt flag is set when counting beyond the maximum value of 0xFF.	Before enabling PCNT0, write the desired top value to the TOPB register. Then load this value into the TOP register by setting LTOPBM in the CMD register.
RTC1	<b>RTC PRS output</b> The RTC PRS output might cause false triggers	If the RTC is selected as a PRS producer there might occur glitches which will accidentally cause false triggers.	Do not use the RTC as a PRS producer, instead use one of the other timer sources (e.g. TIMER0).
TIMER1	<b>TIMER Up/down Mode</b> In up/down mode TIMERN_CCV and TIMER_TOP are updated on both overflow and underflow.	Up/down mode may generate pulses that are not centered around the TIMERN_TOP value.	Correct PWM operation can be ensured by updating TIMERN_CCx_CCVB after the timer overflow flag (TIMERN_IF_OF) is set. When using the DMA overflow/underflow trigger to update TIMERN_CCx_CCVB, the CCV samples in the source buffer must duplicated once. For example, if the CCV sequence is {0x24, 0x100, 0x99}, the buffer the DMA reads from must be {0x24, 0x24, 0x100, 0x100, 0x99, 0x99}. The first sample is written on the underflow event and the second (duplicate) is written on the overflow event. The DMA work-around is not compatible with rev C fix.
USART1	<b>U(S)ART Double Buffer</b> Transmission control through TX-DATAX and TXDOUBLEX does not work with data double buffering.	When a frame is loaded into the transmission shift register, transmission control bits are always taken from outer buffer element. If only one frame is in the U(S)ART buffer, the content of the buffer elements is equivalent, and transmission control bits work as specified. If two frames are in the buffer however, the control bits for the frame in the outer buffer are used for transmitting the frame in inner buffer. This is not a problem for	If using transmission control bits in registers TXDATAX or TXDOUBLEX make sure there are not more than one frame in the U(S)ART buffer at a time, or that the control bits are equal. When TXBL in U(S)ARTn_CTRL is cleared, the TXBL status and interrupt flags in U(S)ARTn_STATUS and U(S)ARTn_IF respectively tell when the buffer is empty. When using transmission control bits, a single frame can then be loaded into the USART for transmission.

ID	Title/Problem	Effect	Fix/Workaround
		frames consisting of more than 9 bits, since these large frames occupy both the inner and outer buffer elements.	
USART2	<b>U(S)ART RXOF</b> In rare situations RX overflow interrupt can be set despite RX data is not lost.	RXOF is set one cycle too early with the consequence that if RX buffer is read in the same internal clock cycle as RXOF is set, the frame causing RX data is actually loaded into the RX buffer. Thus, the overflow interrupt does not guarantee that data has been lost. In addition, this frame will have its lsb cleared when transmitting lsb first, and its msb cleared if msb is transmitted first	On RX overflow, the last frame received may contain a bit error. Disregard this frame.
USART3	<b>U(S)ART Slave TXUF Causes Shift</b> Slave TXUF may take TX out of sync with master.	When in sync slave mode, an underflow may take the slave TX out of sync with the master clock, and data received at the master will be shifted.	Make sure the TX does not underflow when in slave mode.
USART4	<b>U(S)ART Autotri One Cycle Late</b> The AUTOTRI feature enables output one cycle late.	When output enable is controlled by AUTOTRI, the output will be enabled one clock cycle after the first data is output. The largest effect of this will occur when the U(S)ART clock is close to the U(S)ART baudrate.	If the timing of AUTOTRI is not sufficient, use the TXTRIEN/TXTRIDIS commands to enable and disable the USART output.
USART5	<b>U(S)ART Fractional Baudrate</b> Fractional baudrate is wrong for divisors 1.25, 1.50 and 1.75.	The fractional baudrate generated by the U(S)ART is wrong for the divisors 1.25, 1.50 and 1.75. This corresponds to the CLKDIV values 0x40, 0x80 and 0xC0.	Avoid using the erroneous divisors.
USART6	<b>U(S)ART Slave TX Tristate</b> TXTRIEN/TXTRIDIS do not work in half-duplex synchronous slave mode.	In slave mode, the TXTRIEN and TXTRIDIS commands have no effect on the enabled state of the MISO output. The MISO output is controlled solely by whether the slave is selected by the master or not. This affects half-duplex communication when using the USART in synchronous slave mode.	To explicitly control output-enable in slave mode, use GPIO to control the mode of the MISO pin.
USART7	<b>U(S)ART TXC</b> TXC may be set even though data is in the TX buffer.	If data is written to the U(S)ART at the same time as the previous transmission completes, the TXC status flag may be set even though new data has been written to the USART and the USART starts transmission of this data.	Qualify the TXC status with the status of TXBL to know whether the U(S)ART is idle and empty.
USART8	<b>U(S)ART Slave TX Data Required Early</b> For CLKPHA=0, slave TX data is required too early.	When operating with CLKPHA=0, and the slave TX is empty when CS is asserted or on the last edge of a frame, the slave underflows. In this case, no data will be clocked out on the next frame.	Make sure the slave TX has data in time or use CLKPHA=1 in synchronous slave mode.
USART9	<b>U(S)ART Slave TXUF artifacts</b>	When underflowing, a U(S)ART slave may give a pulse on MISO on every setup-edge of the SPI clock. The underflow	Make sure the slave TX has data in time.

ID	Title/Problem	Effect	Fix/Workaround
	MISO toggling and TXUF set multiple times on slave TX underflow.	interrupt flag may also be set multiple times during a frame where the slave underflows.	
USART10	<b>U(S)ART Slave TX Data Lost</b> For CLKPHA=0 a frame is cleared from TX buffer on CS deassert.	When operating as a slave with CLKPHA=0, a frame is cleared from the TX buffer when CS is asserted by the SPI master.	Use CLKPHA=1, or make sure to not write more data to the slave TX than what is intended to transmit during a transmission.
USART11	<b>U(S)ART RX DMA Request After TX</b> For CLKDIV less than 0x100 RX DMA double requests come after the TX DMA double request.	When operating with LOOPBK=1 writing and reading 16-bits at a time to the USART using DMA on a loaded system, this will result in TX requests being handled before RX requests, which may result in TX transmitting frames too fast for RX to handle, leading to RX overflows	Keep the system load under control and handle the U(S)ART overflows
VCMP1	<b>VCMP Mode</b> The VCMP only works in the low power reference mode.	The VCMP only works in the low power reference mode. When the low power reference mode is disabled (by not setting the LPREF bit in VCMP_INPUTSEL), the VCMP does not work and its output is always 1.	When using the VCMP, put it in its low power reference mode by setting the LPREF bit in VCMP_INPUTSEL (which is the default setting). In this mode, the power consumption in the reference buffer (VDD and bandgap) is lowered at the cost of accuracy.
VCMP2	<b>VCMP Current</b> The current consumption of the VCMP is too high in low power reference mode.	When the VCMP is enabled in low power reference mode, the current consumption is higher than specified. The current is the same independent of whether the low power reference mode is enabled or not.	No workaround.
WDOG1	<b>WDOG in EM2/EM3</b> When the watchdog (WDOG) triggers a reset while the EFM32 is in EM2 or EM3, the resulting behaviour is undefined.	After a watchdog reset from EM2/EM3, the EFM32 may go directly to hard fault or may not start at all.	Do not use the watchdog in EM2/EM3 by disabling WDOG before entering EM2/EM3. Note that EM2RUN and EM3RUN bits cannot be used (see WDOG2 Erratum description).
WDOG2	<b>WDOG does not freeze in EM2/EM3</b> The WDOG keeps running in EM2 and EM3 even though EM2RUN and EM3RUN bits in WDOG_CTRL are programmed to 0.	If the WDOG is enabled when entering EM2 or EM3, a WDOG reset will occur unless the system wakes up from EM2/EM3 (by interrupt) and clears the WDOG timer before the WDOG times out.	Disable WDOG before entering EM2/EM3 by writing EN bit in WDOG_CTRL to 0. This requires that the WDOG configuration is unlocked (LOCK bit in WDOG_CTRL = 0). If WDOG configuration is locked, the WDOG will remain enabled in EM2/EM3 and the system must wake up the device from EM2/EM3 and clear WDOG before WDOG times out.
WDOG3	<b>WDOG EM2 detection with LFXO digital/sine input</b> The WDOG will mistake EM2 is EM3 if using LFXO with digital or sine input	When the WDOG is using LFXO with digital or sine input as a clock source, it will mistake EM2 for EM3. The EM2RUN and EM3RUN bits of WDOG_CTRL will behave accordingly.	When using LFXO with digital/sine input, EM3RUN must be set to keep the WDOG running in EM2.

## 2 Revision History

### 2.1 Revision 1.50

January 12th, 2012

Added ADC15.

Added CMU6.

Added CMU8.

Added CMU9.

Added EMU5.

Added WDOG3.

### 2.2 Revision 1.40

November 17th, 2010

Added EMU4.

Added ADC14.

Added DAC8.

Updated BOD1 description to new CMSIS naming.

Updated CMU2 description.

Updated CMU4 description.

Updated CMU5 description.

Updated EMU1 description.

Updated EMU2 description.

Updated ADC10 description.

Updated LEUART2 description.

## **2.3 Revision 1.30**

October 26th, 2010

Added EMU3 and RTC1.

## **2.4 Revision 1.20**

August 31st, 2010

Updated with chip revision C information.

Added WDOG2.

## **2.5 Revision 1.00**

April 23rd, 2010

Initial Version.

## A Disclaimer and Trademarks

### A.1 Disclaimer

*Energy Micro AS 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 Energy Micro 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. Energy Micro 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. Energy Micro 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 must not be used within any Life Support System without the specific written consent of Energy Micro. 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. Energy Micro products are generally not intended for military applications. Energy Micro 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.*

### A.2 Trademark Information

Energy Micro, EFM32, EFR, logo and combinations thereof, and others are the registered trademarks or trademarks of Energy Micro AS. ARM, CORTEX, THUMB are the registered trademarks of ARM Limited. Other terms and product names may be trademarks of others.

## B Contact Information

### B.1 Energy Micro Corporate Headquarters

Postal Address	Visitor Address	Technical Support
Energy Micro AS P.O. Box 4633 Nydalen N-0405 Oslo NORWAY	Energy Micro AS Sandakerveien 118 N-0484 Oslo NORWAY	support.energymicro.com Phone: +47 40 10 03 01

**www.energymicro.com**

Phone: +47 23 00 98 00

Fax: + 47 23 00 98 01

### B.2 Global Contacts

Visit **www.energymicro.com** for information on global distributors and representatives or contact **sales@energymicro.com** for additional information.

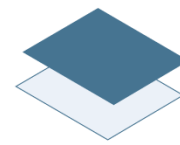
Americas	Europe, Middle East and Africa	Asia and Pacific
<a href="http://www.energymicro.com/americas">www.energymicro.com/americas</a>	<a href="http://www.energymicro.com/emea">www.energymicro.com/emea</a>	<a href="http://www.energymicro.com/asia">www.energymicro.com/asia</a>

# Table of Contents

- 1. Errata History ..... 2
  - 1.1. Errata Overview ..... 2
  - 1.2. EFM32G200 Errata Descriptions ..... 5
- 2. Revision History ..... 15
  - 2.1. Revision 1.50 ..... 15
  - 2.2. Revision 1.40 ..... 15
  - 2.3. Revision 1.30 ..... 16
  - 2.4. Revision 1.20 ..... 16
  - 2.5. Revision 1.00 ..... 16
- A. Disclaimer and Trademarks ..... 17
  - A.1. Disclaimer ..... 17
  - A.2. Trademark Information ..... 17
- B. Contact Information ..... 18
  - B.1. Energy Micro Corporate Headquarters ..... 18
  - B.2. Global Contacts ..... 18

## List of Tables

1.1. Errata Overview .....	2
1.2. EFM32G200 Errata Descriptions .....	5



**ENERGY**<sup>®</sup>  
*micro*

*Energy Micro AS  
Sandakerveien 118  
P.O. Box 4633 Nydalen  
N-0405 Oslo  
Norway*

*[www.energymicro.com](http://www.energymicro.com)*