Wireless Gecko
EFR32MG27 Errata

This document contains information on the EFR32MG27 errata. The latest available revision of this device is revision B.

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

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

Errata effective date: August, 2024.

1. Errata Summary

The table below lists all known errata for the EFR32MG27 and all unresolved errata of the EFR32MG27.

Table 1.1. Errata Overview

Designator	Title/Problem	Workaround Exists	Exists on Revision:	
			A	B
CUR_E305	Increased Leakage Current When PA00 Pin Voltage Is Above GND	Yes	X	X
DCDC_E303	DC-DC gets stuck in Boost Start-up if VBAT POR toggles after DVDD is powered up	Yes	X	—
EMU_E306	IOVDD Brown-Out Misdetection During Supply Ramp	Yes	X	X
EUSART_E303	EUSART Receiver Enters Lockup State when Using Low Frequency IrDA Mode	Yes	X	X
EUSART_E304	Incorrect Stop Bits Lock Receiver	Yes	X	X
IADC_E306	Changing Gain During a Scan Sequence Causes an Erroneous IADC Result	Yes	X	X
USART_E304	PRS Transmit Unavailable in Synchronous Secondary Mode	No	X	X

2. Current Errata Descriptions

2.1 CUR_E305 – Increased Leakage Current When PA00 Pin Voltage Is Above GND

Description of Errata
When PA00 voltage is above GND, up to approximately 3 mA of leakage current can be observed during chip reset and, on boost enabled OPNs, when VBAT is powered and BOOST_EN = 0.
Affected Conditions / Impacts
During chip reset or on boost enabled OPNs when VBAT voltage is present and boost is disabled, if the voltage on PA00 is raised above GND, there will be increased leakage current due to a low impedance path to GND.
Workaround
PA00 pin voltage should remain at GND during chip reset and with boost enabled OPNs, while VBAT is powered and boost is disabled.
Resolution
There is currently no resolution for this issue.

2.2 EMU_E306 – IOVDD Brown-Out Misdetection During Supply Ramp

Description of Errata
The IOVDD brown-out detector incorrectly reports a valid operating level when the IOVDD supply begins ramping after DVDD has reached the mininum operating level and the device has been released from reset.
Affected Conditions / Impacts
Because the IOVDD supply is fully decoupled from the DVDD supply, it is permissible for the DVDD supply to lead the IOVDD supply and thus allow the CPU to exit reset and begin executing code before IOVDD has reached a suitable minimum operating voltage for external logic. In such a configuration, the IOVDD brown-out detector cannot be immediately relied upon to detect a valid operating level because it will inadvertently show that IOVDD is valid over a nominal range of 0.5 V to 0.7 V. The duration of this misdetection and the specific voltage range over which it occurs vary depending on the ramp rate of IOVDD. Variation is also observed from device to device and over temperature. For slower ramps, the duration is extended and the range adheres more closely to 0.5 V to 0.7 V. For faster ramps, the duration is reduced but the range over which the misdetection occurs can shift to higher voltages. The brown-out detector will settle and report correctly within 1 ms of IOVDD reaching its steady-state level.
Workaround
For a system that might be subject to this condition, select one of the following two workarounds: 1. Use a power supply configuration in which IOVDD is tied to or ramps concurrently with DVDD. 2. Characterize the system's IOVDD ramp time and implement a software delay with some headroom (e.g. via Sleeptimer and with the underlying hardware timer clocked from the LFRCO) that must first elapse to account for the misdetection period and before proceeding with the initialization of GPIO pins. **Note:** The IADC cannot be used to monitor the IOVDD ramp because its supply input multiplexer will not be powered until IOVDD reaches a valid operating level.
Resolution
There is currently no resolution for this issue.

2.3 EUSART_E303 — EUSART Receiver Enters Lockup State when Using Low Frequency IrDA Mode

Description of Errata
When low frequency IrDA mode is enabled (EUSART_IRLFCFG_IRLFEN = 1), the receiver can block incoming traffic if it receives either a… • 0 if EUSART_CFG0_RXINV = 0 or • 1 if EUSART_CFG0_RXINV = 1 …before… • the EUSART module is enabled (EUSART_EN_EN =1), • the receiver is enabled (EUSART_CMD_RXEN =1), and • the write to enable the receiver (RXEN = 1) has been synchronized (EUSART_SYNCBUSY_RXEN = 0).
Affected Conditions / Impacts
Incoming traffic will be blocked at the EUSART receiver and subsequent interrupts and status flags will not be set correctly.
Workaround

To avoid entering the lockup state, use one of the workarounds mentioned below:

- When the receiver (RX) input is routed through the PRS:

  Force the input to the IrDA demodulator to high by using the PRS before enabling EUSART. Keep it this way until the receiver has been enabled and EUSART_CMD_RXEN bit is synchronized. See the following code sequence for an example of how to do this:

```
// Output logic 0 through PRS Channel that is connected to EUSART RX GPIO
PRS->ASYNC_CH[0].CTRL = PRS_ASYNC_CH_CTRL_FNSEL_LOGICAL_ZERO |
                    PRS_ASYNC_CH_CTRL_SOURCESEL_GPIO | PRS_ASYNC_CH_CTRL_SIGSEL_GPIOPIN0;

// Select PRS as input to RX.
EUSART0->CFG1_SET = EUSART_CFG1_RXPRSEN;

// Enable EUSART to configure Rx
EUSART0->EN_SET = EUSART_EN_EN;

// Enable Rx
EUSART0->CMD = EUSART_CMD_RXEN;

// Wait until Rx enable is synchronized
while ((EUSART0->SYNCBUSY & EUSART_SYNCBUSY_RXEN) != 0U) {}

// Output EUSART RX pin through PRS Channel
PRS->ASYNC_CH[0].CTRL = (PRS->ASYNC_CH[0].CTRL & ~_PRS_ASYNC_CH_CTRL_FNSEL_MASK) |
                                    PRS_ASYNC_CH_CTRL_FNSEL_A;
```

  Note: EUSART_CTRL_RXINV = 1 in this workaround because the receiver input must be inverted for proper IrDA RZI operation.

- When the receiver (RX) input is not routed through the PRS:

  Force the input to the IrDA demodulator to high by using a GPIO pin other than the current EUSART RX pin before enabling the EUSART. Keep it this way until the receiver has been enabled and EUSART_CMD_RXEN bit is synchronized. See the following code sequence for an example of how to do this:

```
// Configure alternate GPIO (PA00) used for workaround to output 0
GPIO_PinModeSet(gpioPortA, 0, gpioModePushPull, 0);

// Route EUSART0 Rx to the alternate GPIO (PA00)
GPIO->EUSARTROUTE[0].ROUTEEN = (GPIO->EUSARTROUTE[0].ROUTEEN & ~GPIO_EUSART_ROUTEEN_RXPEN);
GPIO->EUSARTROUTE[0].RXROUTE = (gpioPortA << _GPIO_EUSART_RXROUTE_PORT_SHIFT) | (0 <<
_GPIO_EUSART_RXROUTE_PIN_SHIFT);
GPIO->EUSARTROUTE[0].ROUTEEN |= GPIO_EUSART_ROUTEEN_RXPEN;

// Enable EUSART0 to configure Rx
EUSART0->EN_SET = EUSART_EN_EN;

// Enable Rx
EUSART0->CMD = EUSART_CMD_RXEN;

// Wait until Rx enable is synchronized
while ((EUSART0->SYNCBUSY & EUSART_SYNCBUSY_RXEN) != 0U) {}

// Route EUSART Rx to EUSART_RX GPIO(EUSART_RX_PORT & EUSART_RX_PIN)
GPIO->EUSARTROUTE[0].ROUTEEN = (GPIO->EUSARTROUTE[0].ROUTEEN & ~GPIO_EUSART_ROUTEEN_RXPEN);
GPIO->EUSARTROUTE[0].RXROUTE = (EUSART_RX_PORT << _GPIO_EUSART_RXROUTE_PORT_SHIFT) | (EUSART_RX_PORT <<
_GPIO_EUSART_RXROUTE_PIN_SHIFT);
GPIO->EUSARTROUTE[0].ROUTEEN |= GPIO_EUSART_ROUTEEN_RXPEN;

// Disable alternate GPIO (PA00) used for workaround
GPIO_PinModeSet(gpioPortA, 0, gpioModeDisabled, 0);
```

  Note: EUSART_CTRL_RXINV = 1 in this workaround because the receiver input must be inverted for proper IrDA RZI operation.

To exit the lockup state, disable the EUART and force the input to the IrDA demodulator to 1 before re-enabling the EUART by using steps mentioned above.

Resolution

There is currently no resolution for this issue.

2.4 EUSART_E304 — Incorrect Stop Bits Lock Receiver

Description of Errata
When low frequency IrDA mode is enabled (EUSART_IRLFCFG_IRLFEN = 1), the receiver can block incoming traffic if it receives either a… • 0 if EUSART_CFG0_RXINV = 0 or • 1 if EUSART_CFG0_RXINV = 1 …when it is expecting a stop bit.
Affected Conditions / Impacts
Incoming traffic will be blocked at the EUSART receiver. Subsequent interrupts and status flags will not be set correctly.
Workaround

To avoid receiver lock-up in the application firmware caused by formatting errors in the received data, change the receiver GPIO pin routing to force the input to the IrDA demodulator to 1 for the anticipated period of time during which such data can be received.

To exit the lockup state, disable the EUSART and force the input to the IrDA demodulator to 1 before re-enabling the EUSART by using one of the workarounds mentioned below:

- When the receiver (RX) input is routed through the PRS:

  Force the input to the IrDA demodulator to high by using the PRS before enabling EUSART. Keep it this way until the receiver has been enabled and EUSART_CMD_RXEN bit is synchronized. See the following code sequence for an example of how to do this:

```
// Output logic 0 through PRS Channel that is connected to EUSART RX GPIO
PRS->ASYNC_CH[0].CTRL = PRS_ASYNC_CH_CTRL_FNSEL_LOGICAL_ZERO |
                        PRS_ASYNC_CH_CTRL_SOURCESEL_GPIO | PRS_ASYNC_CH_CTRL_SIGSEL_GPIOPIN0;

// Select PRS as input to Rx
EUSART0->CFG1_SET = EUSART_CFG1_RXPRSEN;

// Enable EUSART to configure Rx
EUSART0->EN_SET = EUSART_EN_EN;

// Enable Rx
EUSART0->CMD = EUSART_CMD_RXEN;

// Wait until Rx enable is synchronized
while ((EUSART0->SYNCBUSY & EUSART_SYNCBUSY_RXEN) != 0U) {}

// Output EUSART RX through PRS Channel
PRS->ASYNC_CH[0].CTRL = (PRS->ASYNC_CH[0].CTRL & ~_PRS_ASYNC_CH_CTRL_FNSEL_MASK) |
                        PRS_ASYNC_CH_CTRL_FNSEL_A;
```

  Note: EUSART_CTRL_RXINV = 1 in this workaround because the receiver input must be inverted for proper IrDA RZI operation.

- When the receiver (RX) input is not routed through the PRS:
  Force the input to the IrDA demodulator to high by using a GPIO pin other than the current EUSART RX pin before enabling the EUSART. Keep it this way until the receiver has been enabled and EUSART_CMD_RXEN bit is synchronized. See the following code sequence for an example of how to do this:

```
// Configure alternate GPIO (PA00) used for workaround to output 0
GPIO_PinModeSet(gpioPortA, 0, gpioModePushPull, 0);

// Route EUSART0 Rx to the alternate GPIO (PA00)
GPIO->EUSARTROUTE[0].RXROUTE = (gpioPortA << _GPIO_EUSART_RXROUTE_PORT_SHIFT) | (0 <<
_GPIO_EUSART_RXROUTE_PIN_SHIFT);

// Enable EUSART0 to configure Rx
EUSART0->EN_SET = EUSART_EN_EN;

// Enable Rx
EUSART0->CMD = EUSART_CMD_RXEN;

// Wait until Rx enable is synchronized
while ((EUSART0->SYNCBUSY & EUSART_SYNCBUSY_RXEN) != 0U) {}

// Route EUSART Rx to EUSART_RX GPIO(EUSRT_RX_PORT & EUSART_RX_PIN)
GPIO->EUSARTROUTE[0].RXROUTE = (EUSART_RX_PORT << _GPIO_EUSART_RXROUTE_PORT_SHIFT) | (EUSART_RX_PIN <<
_GPIO_EUSART_RXROUTE_PIN_SHIFT);

// Disable alternate GPIO (PA00) used for workaround
GPIO_PinModeSet(gpioPortA, 0, gpioModeDisabled, 0);
```

  Note: EUSART_CTRL_RXINV = 1 in this workaround because the receiver input must be inverted for proper IrDA RZI operation.

Resolution

There is currently no resolution for this issue.

2.5 IADC_E306 – Changing Gain During a Scan Sequence Causes an Erroneous IADC Result

Description of Errata
Differences in the ANALOGGAIN setting within multiple IADC_CFGx groups during a scan sequence introduces a transient condition that may result in an inaccurate IADC conversion.
Affected Conditions / Impacts
The result of the IADC scan measurement may not match the expected result for the voltage present on the pin during the conversion.
Workaround
Both 1 and 2 shown below must be implemented. 1. If there is a difference in the ANALOGGAIN setting between IADC_CFGx groups during a scan sequence, the IADC_SCHEDx clock prescaler must also change to an appropriate setting. This forces a warmup state (5 µs delay) in between ANALOGGAIN changes. Note that the same IADC_SCHEDx clock prescaler value may be an appropriate setting for both ANALOGGAIN settings, but to force the warmup delay, the IADC_SCHEDx must have different values. 2. The first and last entry of a scan group should use IADC_CFG0, which is the default configuration of the IADC at the start and end of a scan conversion sequence. If CONFIG1 is used at the start and end of the scan group, erronous IADC results may occur.
Resolution
There is currently no resolution for this issue.

2.6 USART_E304 — PRS Transmit Unavailable in Synchronous Secondary Mode

Description of Errata
When the USART is configured for synchronous secondary operation, the transmit output (MISO) is not driven if the signal is routed to a pin using the PRS producer (e.g., SOURCESEL = 0x20 and SIGSEL = 0x4 for USART0).
Affected Conditions / Impacts
Systems cannot operate the USART in synchronous secondary mode if the PRS is used to route the transmit output to the RX (MISO) pin. Operation is not affected in main mode when the transmit output is routed to the TX (MOSI) pin using the PRS producer nor is operation affected in any mode when the GPIO_USARTn_RXROUTE and GPIO_USARTn_TXROUTE registers are used.
Workaround
There is currently no workaround for this issue.
Resolution
There is currently no resolution for this issue.

3. Resolved Errata Descriptions

This section contains previous errata for EFR32MG27 devices.

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 DCDC_E303 – DC-DC gets stuck in Boost Start-up if VBAT POR toggles after DVDD is powered up

Description of Errata
The boost DC-DC enters Boost Start-up mode when the VBAT rail drops below the VBAT POR threshold (approximately 0.6V) after initially powering up DVDD to 1.8 V.
Affected Conditions / Impacts
When the boost DC-DC gets stuck in Boost Start-up mode, the DVDD supply regulates at approximately 2.4V instead of 1.8V, supporting only 3 mA load instead of 25 mA load current.
Workaround
When the boost DC-DC is stuck in Boost Start-up mode, the following steps should be performed to resume normal operation: 1. Shutdown the boost DC-DC by clearing BOOSTENCTRL bit in the EMU_BOOSTCTRL register 2. Toggle the BOOST_EN pin input from logic high to logic low, deactivating the boost DC-DC converter 3. Toggle the BOOST_EN pin input from logic low to logic high, re-activating the boost DC-DC converter
Resolution
This issue is resolved on revision B devices.

4. Revision History

Revision 0.3

August, 2024

- Added CUR_E305, DCDC_E303, EUSART_E303 and EUSART_E304.

Revision 0.2

August, 2022

- Updated for device revision B.

Revision 0.1

May, 2022

- 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

www.silabs.com