Gecko Platform 4.3.1.0 GA
Gecko SDK Suite 4.3
July 26, 2023

The Gecko Platform provides infrastructure support for applications developed with higher-level protocols, and it provides an interface with the underlying hardware. It is composed of the following modules:

CMSIS Device is a vendor-independent hardware abstraction layer for the Cortex®-M processor series.

Peripherals provides a complete peripheral API for all Silicon Labs EFM32, EZR32 and EFR32 MCUs and SoCs.

Drivers is the Gecko Platform driver library for EFM32, EZR32 and EFR32 on-chip peripherals. Drivers are typically DMA-based and use all available low-energy features.

Services includes common services such as NVM3 and Power Manager.

CPC (Co-Processor Communication) provides a library to communicate between two processors using a serial link. CPC is used by the ACP & RCP solutions

Common components are used throughout the SDKs.

Middleware includes the Capacitive Sensing Firmware Library and the GLIB graphics library, along with Micrium OS stacks like CAN/CANopen, File System, Networking and USB Device and Host.

Security includes mbed TLS and other security services.

Operating System includes Micrium OS Kernel as well as other things related to Operating Systems such as a CMSIS-RTOS2 layer.

The **Gecko Bootloader** is a code library configurable through Simplicity Studio's IDE to generate bootloaders that can be used with a variety of Silicon Labs protocol stacks. The Gecko Bootloader can be used with EFM32 and EFR32 Series 1 and later devices.

MVP Math Library includes *Real* and *Complex* Matrix and Vector operations using the Matrix Vector Processor available on EFR32xG24. The library is an alternative to CMSIS-DSP for Matrix and Vector math operations.

Examples are example applications illustrating platform functionality.

Boards and External Devices cover supported hardware.

Other Gecko Platform Components regroups changes to documentation, project building and configuration, as well as any other aspects related to Gecko Platform.

RAIL (Radio Abstraction Interface Layer) provides a customizable radio interface layer that supports proprietary or standards-based wireless protocols. RAIL use by application protocols such as Silicon Labs Zigbee or Silicon Labs Connect is managed through the stack library. Direct RAIL use is exposed through the Flex SDK.

These release notes cover SDK version(s):
Gecko Platform 4.3.1.0 released July 26, 2023
Gecko Platform 4.3.0.0 released June 7, 2023

KEY FEATURES

- Support for EFR32xG27 devices
- Power Manager update for optimized return from EM2
- CPC support for multiple SPI clients
- CPC remote peripheral (GPIO) access
- License change (to open-source zlib) for various platform files

Contents

1 CMSIS Device

1.1 New Items

Added in release 4.3.1.0

- Added support for the following OPNs:
    - EFR32MG24A010F768IM40, EFR32MG24A020F768IM40
    - EFR32BG24B310F1536IJ42
- Added support for the EFR32SG23 family.
- Added support for the EFR32SG28 family.

Added in release 4.3.0.0

- Added support for EFR32xG27 family.
- Added support for EFR32xG28 family.
- Added support for the following OPNs:
    - EFR32FG25A221F1152IM56, EFR32FG25A211F1152IM56
    - EFM32PG28B310F1024IM68, EFM32PG28B210F1024IM68, EFM32PG28B300F512IM68, EFM32PG28B200F512IM68

1.2 Improvements

Changed in release 4.3.0.0

Added support for float16_t for DSP.

Moved CMSIS DSP from platform/CMSIS/DSP to util/third_party/cmsis_dsp.

1.3 Fixed Issues

Fixed in release 4.3.0.0

ID #	Description
1105018	Made changes in pin_config.h defines to avoid MISRA warnings caused due to multiple definitions of macros.

1.4 Known Issues in the Current Release

None

1.5 Deprecated Items

None

1.6 Removed Items

None

2   Peripherals

2.1   New Items

Added in release 4.3.0.0

- em_cmu: Added a new API CMU_HF_ClockPrecisionGet to retrieve the clock precision of HF clocks.
- em_cmu: Added a new API to launch an optimization of the Core Bias Current of HFXO (CMU_HFXOCoreBiasCurrentCalibrate).
- em_system: Added a new API to get the base address of SRAM (SYSTEM_GetSRAMBaseAddress).

2.2   Improvements

Changed in release 4.3.0.0

- em_cmu: Wait on HFXO_STATUS_SYNCBUSY bit after clearing HFXO_CTRL_FORCEEN
- em_cmu: Added support for LFXO in CMU_OscillatorTuningSet/ CMU_OscillatorTuningGet
- em_lcd: Added support for Sextaplex and Octaplex mode (4x28 and 8x24).

2.3   Fixed Issues

Fixed in release 4.3.0.0

ID #	Description
1131889	em_eusart: Fixed an array-bounds warning in em_eusart.c
1114232	em_lesense: fixed LESENSE_DecoderStateAllConfig and LESENSE_DecoderStateConfig to account for the correct number of states for Series 2 devices.

Fixed in release 4.3.0.0

ID #	Description
1127003	Fixed Buck DCDC IPKVAL and DRVSPEED default settings for Series 2 devices.
1106525	For Series 2 devices, added a timeout to EMU_DCDCModeSet function to avoid deadlock when called from an interrupt context during EM4 shutdown process.
1083378	For Series 2 devices, disabled Interrupts while entering EM4 to prevent modification to DCDC Mode. If DCDC is in regulation mode while trying to reach the EM4 state, the power state transition will be aborted, and the DCDC_IF_EM4ERR condition will be set.
1119860	Updated IADC_init function to removed an implicit conversion from double to float compiler warning.
412632	Fix ADC_Init so it does not overwrite SINGLEDMAWU in CTRL register when ADC in configured in ASYNC mode.
1079268	Fixed CMU_ClockSelectGet() to return the right value when the EM01GRPACLK clock is disabled instead of cmuSelect_Error.

2.4   Known Issues in the Current Release

None

2.5   Deprecated Items

None

2.6   Removed Items

None

3 Drivers

3.1 New Items

None

3.2 Improvements

Changed in release 4.3.0.0

The platform/driver/mvp source catalog has been moved to the new position platform/compute where the content has been split into the component, driver, math, and nn catalogs. The rationale behind this change is to have all the software that supports hardware accelerators under the platform/compute catalog. In the new structure, the math library is placed in its own directory and it was natural to separate the CMSIS-NN compatible library functions used by Tensorflow Lite Micro from the driver. The MVP driver, nn and math libraries are now supported by new slc-components and the mvp.slcc component has been updated to bring in all the other MVP components by using only this component. This will ensure application compatibility with the previous file structure when using the mvp-component. (Release note updated in release 4.3.1.0)

3.3 Fixed Issues

Fixed in release 4.3.0.0

ID #	Description
1105632	Fixed an implicit unsigned to signed casts warning in the DMADRV_AllocateChannel() function.
1099270	Fixed a bug in the SPIDRV_DeInit function, where the allocated/routed GPIOs were not reset.
1095676	Fixed I2CSPM to send 9 clocks during the i2c bus recovery sequence.
1095611	Fixed the sl_spidrv_init_instances function of the SPIDRV component to correct the EUSART Chip Select control mode (control mode set to spidrvCsControlApplication).
854128	Change default flow control settings in the UARTDRV component to uartdrvFlowControlHwUart.

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

The SLEEP and RTCDRV drivers from platform/emdrv were removed.

4 Services

4.1 New Items

Added in release 4.3.0.0

- Power Manager: Added a new power manager component power_manager_deepsleep_blocking_hfxo_restore. This component improves power consumption while waiting for HFXO to be ready coming from EM2 sleep. This is only compatible with series 2 devices that do not have the SYSRTC peripheral (EFR32xG21, EFR32xG22 and EFR32xG27) and thus do not have the capacity to startup HFXO while being in EM2.

4.2 Improvements

Changed in release 4.3.0.0

- Power Manager: When the sleeptimer component is configured to use the SYSRTC, the Power Manager uses PRS signaling rather than timers to initialize the HFXO. This ultimately reduces the HFXO startup time from ~600 µs to ~250 µs.
- Power Manager: The power manager SL_POWER_MANAGER_LOWEST_EM_ALLOWED macro used to define the lowest energy mode has been deprecated. Two mutually exclusive components have been created, power_manager_deeplseep (present by default) and power_manager_no_deeplseep to replace this configuration in order to reduce code size. Migration from the configuration to the component is automatic when upgrading the GSDK.
- Clarified the semantics of optional command-line (CLI) arguments in header comments.
- IOStream now automatically configures the VCOM peripheral on the WSTK/WPK devices for seamless communication.

4.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1167174	HFXO Manager: HFXO default startup is now 600 µs for when either PRS or a timer is used to start the HFXO. There is no longer the distinction between 300 µs & 600 µs startup for when PRS used vs when timer used to start HFXO. 300 µs HFXO default startup value could cause issues related to radio operation during the first EM2 wake up events.
1157958	HFXO manager: now checks if the SYSRTC peripheral is used by the sleeptimer instead of if the SYSRTC peripheral is present on this IC for Series 2 devices, fixing issues if another RTC timer was used by the sleeptimer.
1152110	Power Manager: Fixed a bug in the power manager component preventing a project not containing the power manager component to be upgraded to a newer GSDK. The bug was causing the upgraded project to be invalid.
1122968	MPU: Fixed `sl_mpu_disable_execute` to return an error when defining a region that overlaps a region already defined in the MPU to prevent a hard fault to occur on ARMv8-M architecture.

Fixed in release 4.3.0.0

ID #	Description
1109520	Fix issue in Sleeptimer related to integration with Power Manager Early wake-up feature. This issue appeared on FreeRTOS-based applications. An abnormally high HFXO startup time was observed because of the issue leading to the application entering EM1 sleep instead of EM2.
1139472	Fixed a bug in Series 0 where the get_write_ptr function computed an incorrect value.
1130712	Fixed an issue in the deinitialization process of IOStream-UART where the power requirement was not properly removed, resulting in an incorrect power state.

4.4　Known Issues in the Current Release

None

4.5　Deprecated Items

None

4.6　Removed Items

None

5 CPC

5.1 New Items

Added in release 4.3.0.0

- Implemented a significant modification that enables the sharing of the SPI bus among multiple CPC secondary devices, sensors, and external peripherals.

  This enhancement empowers CPCd to utilize the native chip select (CS) functionality, eliminating the reliance on a GPIO pin for bus access control. Consequently, an update is required for the Linux Host system to accurately map the CS GPIO to a SPI CS. This modification significantly improves the flexibility and efficiency of SPI bus utilization, enabling seamless communication and coordination among CPC devices and connected peripherals.

  It is important to note that this update may necessitate adjustments in the Linux Host system to ensure proper mapping. Additionally, please consider the compatibility implications with previous CPC versions and carefully plan an appropriate upgrade strategy to ensure a smooth transition. As a result of these changes, the protocol version has been incremented to accommodate the modifications.

- Added the CPC GPIO Expander component, a valuable addition that empowers users to discover and harness the GPIO pins available on the device. With this feature, the Host system gains effortless access to the GPIO capabilities provided by the CPC device. The CPC GPIO Expander component streamlines the process of exposing and utilizing GPIO pins. On a Linux Host, a dedicated kernel driver can be installed, enabling seamless access to the GPIOs present on the device, treating them as standard GPIOs. This integration allows users to interact with the CPC GPIOs using familiar Linux GPIO APIs, simplifying the development and integration of GPIO-based functionalities.

5.2 Improvements

Changed in release 4.3.1.0

- Fixed potential overflow when application version string is too long.
- Introduced a new script to enhance the configuration validation process for projects that utilize a VCOM-based interface. The script specifically focuses on validating the setting SL_BOARD_VCOM_ENABLE to ensure it is correctly enabled when working with a VCOM interface.
- Removed dead code from UART driver.

Changed in release 4.3.0.0

- Implemented a significant security enhancement that excludes 'ack' and the header's CRC from the authenticated data when the security feature is enabled. Consequently, to accommodate these changes, the protocol version has been upgraded. Please note that this protocol version upgrade may result in a break in compatibility with previous CPC versions.
- Implemented major improvements in the UART driver, specifically for cases where Hardware Flow control is not available, to recover from an invalid CRC. This enhancement introduces robust error handling mechanisms that allow the driver to detect and recover from CRC errors, ensuring the integrity of data transmission. These improvements significantly enhance the reliability and error resilience of the UART communication, improving the overall performance and correctness of the system.
- Enhanced libcpc library by adding pkg-config support, enabling build systems to seamlessly locate the library during compilation.
- Added a default initialization value for the re-transmission (re-tx) timeout.
- Incorporated a behavior in CPCd where it drops any U/I frames while awaiting the secondary device's reset reason. This modification enhances the daemon's capability to ignore frames from previous sessions, mitigating the risk of undefined behaviors that may occur when handling such frames.
- Enhanced buffer management to prevent deadlock scenarios.
- Implemented API protection in the libcpc library to prevent unsafe calls and ensure proper usage. As a result, it is now required to close all libcpc handles before invoking the 'cpc_restart' function.
- Removed the CMake target group 'target_test' from the project. This target was specifically designed for internal testing.

5.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1141564	Resolved an issue where the initialization between CPCd/Secondary would fail at runtime if the request for the application version couldn't be acknowledged.
1150872	Addressed an issue causing negative RTO (retransmit timeout) calculation.

Fixed in release 4.3.0.0

ID #	Description
1125380	Resolved a warning related to sl_cpc_iostream cast compatibility with the IAR compiler.
1117787	Addressed a stack smashing issue that occurred when the compiler option was activated.
1105518	Resolved an issue where packets were being unnecessarily retransmitted twice, resulting in redundant transmissions.
1093597	Addressed a critical bug in the UART/SPI driver that could lead to frame corruption during transmission. Previously, the bug caused invalid pointers within the frame when the TX complete callback was invoked, resulting in corruption.
1105185	Resolved an issue where a reject frame was incorrectly interpreted as a valid acknowledgement (ack).
1107259	Fixed the CPC IOStream bridge to enable external connections for enhanced connectivity and communication capabilities.

5.4 Known Issues in the Current Release

None

5.5 Deprecated Items

None

5.6 Removed Items

None

6 Common

6.1 New Items

Many platform components license have changed to use the ZLib license.

6.2 Improvements

None

6.3 Fixed Issues

None

6.4 Known Issues in the Current Release

None

6.5 Deprecated Items

None

6.6 Removed Items

None

7   Middleware

7.1   New Items

None

7.2   Improvements

None

7.3   Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1159242	Fixed current page size calculation before writing to flash.

Fixed in release 4.3.0.0

ID #	Description
748986	Fixed bug #61333 (https://savannah.nongnu.org/bugs/?61333) in lwIP module.

7.4   Known Issues in the Current Release

None

7.5   Deprecated Items

None

7.6   Removed Items

None

8 Security

8.1 New Items

Added in release 4.3.1.0

- Added new evaluation-level components for enabling specific RSA algorithms in Mbed TLS and PSA Crypto. The following config options can now be enabled through the SLC components provided in the GSDK: MBEDTLS_PKCS1_V15, MBEDTLS_PKCS1_V21, PSA_WANT_ALG_RSA_PKCS1V15_CRYPT, PSA_WANT_ALG_RSA_OAEP, PSA_WANT_ALG_RSA_PKCS1V15_SIGN, PSA_WANT_ALG_RSA_PSS, MBEDTLS_KEY_EXCHANGE_ECDHE_RSA_ENABLED, and MBEDTLS_RSA_NO_CRT. Please refer to mbedtls/include/mbedtls/mbedtls_config.h and mbedtls/include/mbedtls/config_psa.h for further information on these config options.

Added in release 4.3.0.0

- Added functionality for setting/getting the most secure lifetime of a key in PSA Crypto based on device capabilities. See `sl_psa_crypto.h` for more details.
- Added a key derivation function (PBKDF2-CMAC) support for PUF-based built-in key for xG27.
- In the SE Manager function `sl_se_init_otp` , added an option to configure the tamper block to remain active in EM2 for xG23 and later devices.
- Added evaluation quality support for using Secure Engine as an endpoint over CPC.
- Added threading support for the PSA Crypto library and drivers:
  - PSA Crypto Key Slot Management
    - supports importing keys to be used by reference (keyid) in crypto functions.
  - PSA ITS driver
    - for permanent secure key storage.
  - PSA TRNG driver for Series-1
    - called by PSA APIs like `psa_generate_key` and `psa_generate_random`.

8.2 Improvements

Changed in release 4.3.1.0

- Disabled code execution in the NVM3 region in the TrustZone NonSecure applications.

Changed in release 4.3.0.0

Removed disabling of code execution in the NVM3 region for TrustZone NonSecure apps. In other words, code execution is now possible in the NVM3 region in the NonSecure app, but not in the Secure library. This is a work-around for a potential situation of multiple overlapping MPU regions, which is not supported by the Cortex-M33 and may cause faults. A fix that disables code execution in the NVM3 region in NonSecure apps is planned in a GSDK 4.3.x patch release.

8.3   Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1154777	Fixed a bug in the PSA Crypto AEAD driver for Series-1 SoCs/Modules where the driver would potentially return an incorrect 'tag_length' for the 'psa_aead_finish()' function. The value of 'tag_length' would depend on the value passed in the 'tag_size' argument instead of the actual length of the tag. The bug affected both AES-CCM and AES-GCM modes of operation. The driver would never allow a 'tag_size' less than the tag length specified by the algorithm chosen, so this bug is not considered a vulnerability.
1148593	Corrected the behavior of the PSA Crypto Cipher driver on HSE devices (xG21, xG23, xG24, xG25, xG28) to make sure that calls to psa_cipher_encrypt_setup()/psa_cipher_decrypt_setup() return PSA_ERROR_NOT_SUPPORTED if the current configuration does not include support for the selected cipher algorithm.
1167059	Fixed a bug in the PSA ITS driver (version 1 and 2) that can potentially lead to inaccessibility of some keys when there are more than 16 keys (stored in NVM3 by the ITS driver). The ITS driver version 3 is not affected. Affected SDK versions are 4.3.0 and earlier (versions including PSA ITS version 1 and 2 drivers). Affected devices are EFx32xG1 and EFR32xG2.

8.4   Known Issues in the Current Release

None

8.5   Deprecated Items

None

8.6   Removed Items

None

9 Operating System

9.1 New Items

None

9.2 Improvements

None

9.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1154620	Fixed an issue in MicriumOS where a HardFault exception occurs when the scheduler is configured in Round Robin mode and the last task in the application deletes itself.

9.4 Known Issues in the Current Release

ID #	Description	Workaround
1108940	When osThreadTerminate() is called from the task that needs to be terminated a memory leakage occurs.	Call the OsThreadTerminate from another task specifying correct thread_id to terminate the designated task.
1166857	OSTaskQPost doesn't send message to current task when called with a null pointer as argument.	N/A

9.5 Deprecated Items

None

9.6 Removed Items

None

10 Gecko Bootloader

10.1 New Items

None

10.2 Improvements

None

10.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1161217	Added Fix in parser_parseApplicationInfo() function to fix the leakage from secure area

Fixed in release 4.3.0.0

ID #	Description
1105352	Fixed OTA DFU App component issues with default optimization.
1107186	Fixed Project Creation fails for BG24 Bootloader SoC Internal Storage (single image on 1 MB device).
1111035	Fixed bootloader SPI error.
1138508	Fixed format of Bootloader Upgrade Location Base configuration option in the Bootloader Core component.
1130159	Fixed issue to enable clock for bootloader SPI Controller EUSART Driver.
1124579	Fixed Bootloader - SoC Bluetooth AppLoader OTA DFU Secure part fails to build.
1122767	Fix added to support to build Trust Zone apps using workspaces.

10.4 Known Issues in the Current Release

None

10.5 Deprecated Items

None

10.6 Removed Items

None

11 MVP Math Library

11.1 New Items

Added in release 4.3.0.0

The new Math Library accelerates Real and Complex Matrix and Vector operations using the Matrix Vector Processor available on EFR32xG24 (see parts with MVP available) operating on IEEE-754 half-precision floating point data and 8-bit integers, while offloading CPU. For heavy workloads it will improve overall performance and efficiency significantly.

The library is an alternative to CMSIS-DSP for Matrix and Vector math operations.

Category	Function
Vector Math	sl_math_mvp_complex_vector_conjugate_f16
	sl_math_mvp_complex_vector_dot_product_f16
	sl_math_mvp_complex_vector_magnitude_squared_f16
	sl_math_mvp_complex_vector_mult_real_f16
	sl_math_mvp_complex_vector_mult_f16
	sl_math_mvp_vector_abs_f16
	sl_math_mvp_vector_add_f16
	sl_math_mvp_vector_add_i8
	sl_math_mvp_clamp_i8
	sl_math_mvp_vector_clip_f16
	sl_math_mvp_vector_copy_f16
	sl_math_mvp_vector_dot_product_f16
	sl_math_mvp_vector_fill_f16
	sl_math_mvp_vector_mult_f16
	sl_math_mvp_vector_negate_f16
	sl_math_mvp_vector_offset_f16
	sl_math_mvp_vector_scale_f16
	sl_math_mvp_vector_sub_f16
Matrix Math	sl_math_mvp_complex_matrix_mult_f16
	sl_math_mvp_matrix_add_f16
	sl_math_mvp_matrix_mult_f16
	sl_math_mvp_matrix_scale_f16
	sl_math_mvp_matrix_sub_f16
	sl_math_mvp_matrix_transpose_f16
	sl_math_mvp_complex_matrix_transpose_f16
	sl_math_mvp_matrix_vector_mult_f16

Types

Note that this library mainly supports *float16* input/output.

11.2 Improvements

Changed in release 4.3.0.0

The platform/driver/mvp source catalog has been moved to the new position platform/compute where the content has been split into the component, driver, math, and nn catalogs. See Drivers > Improvements for more information.

11.3 Fixed Issues

None

11.4 Known Issues in the Current Release

None

11.5 Deprecated Items

None

11.6 Removed Items

None

12 Examples

12.1 New Items

None

12.2 Improvements

None

12.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1106289	Increased the heap size for the usb_device_cdc_acm_micriumos, usb_device_composite_micriumos and usb_device_hid_micriumos IAR applications, application was configured with a too small heap, preventing it from working correctly.

12.4 Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on https://www.silabs.com/products/software.

ID #	Description	Workaround
664803	Se_manager and psa_crypto sample apps do not work correctly in Simplicity Studio 5's launch console.	In the launch console, change the line terminator selection to None.

12.5 Deprecated Items

None

12.6 Removed Items

None

13 Boards and External Devices

13.1 New Items

Added in release 4.3.1.0

- Added support for the following new OPNs
- BRD2705A, BRD2905A, BRD4115B

Added in release 4.3.0.0

- Added support for the following new OPNs
    - BRD4110B, BRD4111B, BRD2602A, BRD4400C, BRD4401C, BRD2900A, BRD2901A, BRD2902A, BRD2903A, BRD2904A, BRD2503B, BRD2506A

13.2 Improvements

None

13.3 Fixed Issues

None

13.4 Known Issues in the Current Release

None

13.5 Deprecated Items

None

13.6 Removed Items

None

14 Other Gecko Platform Software Components

14.1 New Items

None

14.2 Improvements

None

14.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
1173756	Fixed a warning not being displayed correctly when generating a project with SLC and the project contains the freeRTOS component but does not contain the NVIC device initialization component (except when the device has a Cortex-M0+ core).

Fixed in release 4.3.0.0

ID #	Description
1077816	Added APIs to fetch FLASH base address in both internal and external flash scenarios. Made external flash address configurable.
1083908	Added preprocessor macros to separate the use of internal flash(MSC) and SPI in iot_flash. The file is now an auto generated file.

14.4 Known Issues in the Current Release

None

14.5 Deprecated Items

None

14.6 Removed Items

None

15 RAIL Library

15.1   New Items

Added in release 4.3.1.0

- Added new RAIL_GetAutoAckFifo() API and allow NULL for RAIL_WriteAutoAckFifo() or RAIL_IEEE802154_WriteEnhAck() ackData parameter, which gives applications direct access to the AutoAck FIFO to construct Ack packets in pieces.
- Added support for antenna selection through the applicable RAIL_RxOptions_t and RAIL_TxOptions_t values when using OFDM on the EFR32xG25.
- Added a new "RAIL Utility, SFM Sequencer Image Selection" component to allow selection of modulations supported by EFR32XG25 software modem (SFM). These changes can save considerable flash space by reducing the set of modulations to just those that are needed.
- Added support for Sidewalk PHYs for EFR32xG23 and EFR32xG28 chips.
- Added a new component to switch between Coexistence, Antenna Diversity and FEM Utilities for EFR32xG21 and EFR32xG24 chips supporting 15.4 Fast Channel Switching feature.

Added in release 4.3.0.0

- Added support for a new RX_DIRECT_SYNCHRONOUS_MODE_DATA RAIL RX data source to capture direct mode data in sync with the configured bit rate for the PHY. This requires a PHY that supports this mode of capture from the Radio Calculator and is only currently supported on the EFR32xG23.
- Added new RAIL_EnableCacheSynthCal function to enable the radio sequencer to cache calibration values instead of recalculating them on every RX and TX event. This allows you to lower the minimum transition time for most RAIL_StateTiming_t transitions in typical cases.
- Added a new RAIL_RX_OPTION_FAST_RX2RX which will force the radio sequencer to immediately transition to RXSEARCH to get ready to receive the next packet while still processing the previous one. This will minimize the RX to RX state transition time. This is only supported on chips that have RAIL_SUPPORTS_FAST_RX2RX set to true.
- Added RAIL_PacketTimeStamp_t::packetDurationUs field which is currently set only on EFR32xG25 for received OFDM packets.
- Added RAIL support for the MGM240L lighting modules.
- Added the new RAIL_WMBUS_Config API to allow configuring WMBUS and simultaneous M2O RX of T and C mode packets.
- Added a new API RAIL_SetTxFifoAlt() which provides a new start offset parameter to specify where the data begins in the TX FIFO.
- Added support for RAIL_IEEE802154_SupportsRxChannelSwitching on the EFR32xG21. This is also still supported at an alpha quality level on the EFR32xG24.
- Added RAIL_IEEE802154_SetRxToEnhAckTx() to allow IEEE 802.15.4 stacks to specify a different rxToTx state transition turnaround time for Enhanced ACKs, which generally need more time to construct and secure. Immediate ACKs will continue to use the rxToTx time specified in RAIL_IEEE802154_Config_t::timings.
- Added Tx packet duration information for EFR32xG22 and newer chips.

15.2   Improvements

Changed in release 4.3.1.0

- Updated the RAIL_PacketTimeStamp_t::packetDurationUs field on the EFR32xG25 to be populated for non-OFDM packets.

Changed in release 4.3.0.0

- Added support for PHY-specific RSSI offsets on the EFR32xG27 and EFR32xG28 platforms.
- Updated Packet Trace on the EFR32xG25 and EFR32xG28 when using the Wi-SUN protocol to have a more informative PHY identifier and to support the whole channel number range.
- Fixed the RAIL_PA_CURVES_2P4_LP power curves on the EFR32xG24 to better match characterization data.
- Corrected the sign of the frequency error reported by RAIL_GetRxFreqOffset() when using OFDM on the EFR32xG25 to match how this was handled for other modulations (e.g., Freq_error=current_freq-expected_freq).
- Added new RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE to trigger RAIL_EVENT_ZWAVE_BEAM on all beam frames.
- Added RAIL_ZWAVE_GetBeamHomeIdHash() to retrieve the beam frame's HomeIdHash when handling that event and made sure that the HomeIdHash byte is now present on PTI for Z-Wave beam frames even when NodeId does not match.

- Fixed an issue on EFR32ZG23 where multiple beam frames were lumped together on PTI as one large beam chain.
- Adjusted channel power restrictions for the 802.15.4 PHYs on new xGM210 modules.
- Added separate curves when the 20 dBm PA is used at 3.3 V and 1.8 V for EFR32xG24.
- Increased EFR32ZG13 and EFR32ZG14 Z-Wave long-range beam detect time to improve FLiRS performance.

15.3 Fixed Issues

Fixed in release 4.3.1.0

ID #	Description
707731	Fixed an issue when using RAIL_BLE_SetNextTxRepeat() that would cause an incorrect Protocol Config ID to appear in the packet trace for repeated transmits.
1079816	Fixed a race condition on EFR32xG22 and later during RX channel hopping or duty-cycling where frame detection occurring close to when a hop should happen could leave the radio stuck in reception but not receiving anything, with the only remedy being to idle the radio.
1088439	Fixed an issue which would cause the incorrect antenna to be reported for a received packet on the EFR32xG25 when using OFDM and antenna diversity.
1141539	Fixed an issue to improve CCA behavior for regional certification test suites for EFR32xG25 chips.
1153679	Fixed an issue in "RAIL Utility, Coexistence" component where a GRANT signal pulse less than 100 µs might result in the radio not being properly placed in hold off after GRANT is deasserted.
1156980	Fixed an issue with channel hopping on EFR32xG22 and later where use of RAIL_RX_CHANNEL_HOPPING_OPTION_RSSI_THRESHOLD can prevent the timed RX channel hopping modes including RX duty-cycling from timing out properly.
1167235	Fixed an issue with RAIL_SupportsFastRx2Rx() where it used to return an incorrect value on supported platforms.

Fixed in release 4.3.0.0

ID #	Description
824355	Fixed an issue in IEEE802.15.4 MAC address filtering when receiving small OFDM packets.
832743	Clarified use of RAIL_SetNextTxRepeat() must be prior to initiating a transmit operation via API call and fixed an issue where it did not properly return an error when called while a transmit operation was in progress.
1055824	Fixed an issue with low-side synth injection (negative IF) on proprietary 2.4GHz PHYs when using EFR32xG22 and newer chips. This fix requires regenerating the PHY with the latest version of the Radio Configurator to work.
1058480	Fixed an RX FIFO corruption on EFR32xG25 that occurred when receiving/sending certain OFDM packets using FIFO mode.
1082274	Fixed an issue on the EFR32xG22, EFR32xG23, EFR32xG24, and EFR32xG25 chips that could cause the chip to lock up if the application attempts to re-enter EM2 within ~10 µs after wake-up and hits a <0.5 µs timing window. If hit, this lockup requires a power-on reset to restore normal operation to the chip.
1083615	Fixed an issue for certain ramp time and power level combinations on the EFR32xG21 where the PA ramp would stop one power level short of the desired output level.
1090336	Fixed an issue in the "RAIL Utility, Protocol" component where BLE would be required to select a Zigbee PHY.
1090512	Fixed an issue in the "RAIL Utility, PA" component where certain functions would attempt to use the RAIL_TX_POWER_MODE_2P4GIG_HIGHEST macro even though they didn't support it. This would result in undefined behavior previously, but will now correctly error.
1090728	Fixed a possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO issue on EFR32xG12 with RAIL_IEEE802154_G_OPTION_GB868 enabled for a FEC-capable PHY which can happen when aborting a packet at frame detection, for instance by idling the radio.
1092769	Fixed an issue when using Dynamic Multiprotocol and BLE Coded PHYs where a transmit could underflow depending on what protocol was active when the PHY and syncword were loaded.
1096663	Fixed a compilation error in "RAIL Utility, Coexistence" component when the Coexistence WiFi TX GPIO is enabled.

ID #	Description
1096665	Fixed a compilation issue in "RAIL Utility, Coexistence" component when the SL_RAIL_UTIL_COEX_WIFI_TX_PORT is defined.
1103966	Fixed an unexpected Rx packet abort on the EFR32xG25 when using the Wi-SUN OFDM option4 MCS0 PHY.
1104033	Fixed an issue in the RAIL_ZWAVE_ReceiveBeam function so that it idles the radio regardless of whether a beam is detected on the EFR32ZG23.
1104441	Fixed an issue with the "RAIL Utility, Coexistence" component counters for Zigbee that could prevent them from ticking as expected depending on how things are linked.
1105134	Fixed an issue when switching between certain PHYs that could cause the first received packet to be reported as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS. This issue could potentially impact EFR32xG22 and newer chips.
1105529	Fixed an issue on EFR32xG22 and later platforms when using a FrameType decoding PHY where a bad frame type packet was mis-reported as RAIL_RX_PACKET_ABORT_ABORTED instead of the proper RAIL_RX_PACKET_ABORT_FORMAT.
1109574	Fixed an issue on EFR32xG22 and newer chips where a radio sequencer assert could cause the application to hang in an ISR rather than report the assert via RAILCb_AssertFailed().
1118063	Fixed issue with recent RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE on EFR32xG13 and EFR32xG14 where the NodeId of the promiscuous beam was not properly recorded for RAIL_ZWAVE_GetBeamNodeId(), causing it to report 0xFF.
1126343	Fixed an issue on EFR32xG24 when using the IEEE 802.15.4 PHY where the radio could become stuck when doing an LBT transmit if a frame is received during the CCA check window.
1134223	Fixed an issue when using "RAIL Utility, Coexistence" component where the request line is left asserted after TX is aborted following a sync detect.
1135418	Fixed incorrect RAIL_RxPacketInfo_t::filterMask on received 802.15.4 Beacon frames, which now reflects which PanId and address the Beacon's Source PanId and Source Address match, if any. Note that RAIL generally accepts all Beacons so the filterMask can be 0x00.
1138522	Fixed an issue on the EFR32xG25 for SUN FSK PHYs where receiving a packet after calling RAIL_IEEE802154_Init() but before configuring RAIL_IEEE802154_ConfigGOptions could break reception.
1140569	Fixed a rare timing issue on EFR32xG24 where an ACK timeout might cause the next packet to be received as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS.
1150779	Fixed the 15.4 channel configurations on the MGM240PA32 and MGM240PB32 modules to use the correct frequency for channel 26.

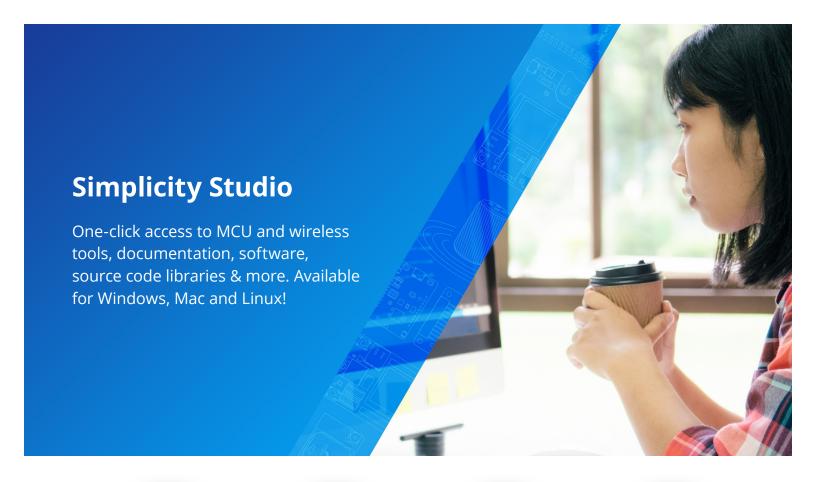## 15.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
	Using direct mode (or IQ) functionality on EFR32xG23 requires a specifically set radio configuration that is not yet supported by the radio configurator. For these requirements, reach out to technical support who could provide that configuration based on your specification	
641705	Infinite receive operations where the frame's fixed length is set to 0 are not working correctly on the EFR32xG23 series chips.	
732659	On EFR32xG23: • Wi-SUN FSK mode 1a exhibits a PER floor with frequency offsets around ± 8 to 10 KHz • Wi-SUN FSK mode 1b exhibits a PER floor with frequency offsets around ± 18 to 20 KHz	

15.5 Deprecated Items

None

15.6 Removed Items

None

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

www.silabs.com