Gecko Platform 4.4.2.0 GA

Gecko SDK Suite 4.4

April 10, 2024

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.

KEY FEATURES

Peripherals

- New API names introduced for low-level drivers (with compatibility layer for old names)

Services

- APIs for accessing value of tokens added to Token Manager

CPC

- NVM3 module, enabling Host access to Secondary device's non-volatile memory, released at production quality
- CPC Primary, for use with MCU Hosts, released at Experimental quality
- Numerous optimizations and performance improvements made

Security

- Mbed TLS upgraded (to version 3.5.0)

RAIL

- Several new features added for EFR32xG25 devices, including a new component for selecting modulations supported by the software modem
- Support added for several new PHYs, including Sidewalk PHYs on EFR32xG23 and EFR32xG28

Other Components

- Compilers upgraded (to GCC 12.2.1 and IAR 9.40.1)

These release notes cover SDK version(s):

Gecko Platform 4.4.2.0 released April 10, 2024
Gecko Platform 4.4.1.0 released February 14, 2024
Gecko Platform 4.4.0.0 released December 13, 2023

Contents

1 Upcoming GSDK Changes 3
2 CMSIS Device 4
3 Peripherals 5
4 Drivers 7
5 Services 8
6 CPC 10
7 Common 12
8 Middleware 13
9 Security 14
10 Operating System 16
11 Gecko Bootloader 18
12 MVP Math Library 19
13 Examples 20
14 Boards and External Devices 21
15 Other Gecko Platform Software Components 22
16 RAIL Library 23

1 Upcoming GSDK Changes

This release of the Gecko SDK (GSDK) will be the last with combined support for all EFM and EFR devices, except for patches to this version as needed. Starting in mid-2024 we will introduce separate SDKs:

- The existing Gecko SDK will continue with support for Series 0 and 1 devices.
- A new SDK will cater specifically to Series 2 and 3 devices.

The Gecko SDK will continue to support all Series 0 and 1 devices with no change to the long-term support, maintenance, quality, and responsiveness provided under our software policy.

The new SDK will branch from Gecko SDK and begin to offer new features that help developers take advantage of the advanced capabilities of our Series 2 and 3 products.

This decision aligns with customer feedback, reflecting our commitment to elevate quality, ensure stability, and enhance performance for an exceptional user experience across our software SDKs.

2 CMSIS Device

2.1 New Items

None

2.2 Improvements

None

2.3 Fixed Issues

Fixed in release 4.4.0.0

ID #	Description
1187465	Removed metadata indicating WiSUN support from devices that only support 2.4 GHz band.
1136027	Removed the support for AoA locator in xG27 device metadata configuration files (slcc) as this is not supported by the device.

2.4 Known Issues in the Current Release

None

2.5 Deprecated Items

None

2.6 Removed Items

None

3 Peripherals

3.1 New Items

Added in release 4.4.0.0

- Peripheral: Renamed APIs of low-level drivers in platform/peripheral(dcdc_coulomb_counter, etampdet, keyscan and sysrtc) from "sl_xxx" to "sl_hal_xxx". For example, sl_keyscan_init convert to sl_hal_keyscan_init. A compatibility layer is added so users can still use the old APIs in their application.
- em_timer: Added Reload-Start Sets Compare Output initial State rssCoist bit to TIMER_Init initialization structure.
- em_eusart: Added a watermark setting for EUART/EUSART in UART mode for users to configure the interrupt and status level of the transmit or receive FIFO.
- em_acmp: Added missing the EXTPx values to ACMP_Channel_TypeDef structure.
- em_eusart: Added auto UART TX delay transmission configuration.

3.2 Improvements

Changed in release 4.4.1.0

- em_iadc: Removed unnecessary assignment of value to uiAnaGain in em_iadc.c.

Changed in release 4.4.0.0

- em_vdac: Updated default settings for ondemand clock and refreshperiod, which were not the same as the default registers setting for VDAC_INIT_DEFAULT on series 2.
- em_cmu: Added Port check available for routing of the HF clock 0 and 1 to GPIO.

3.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1268728	em_burtc: Fixed an error in BURTC_Enable() and BURTC_Reset() which don't wait for write synchronization after writing to BURTC->EN on devices: xG21, xG22, and xG27.

Fixed in release 4.4.1.0

ID #	Description
1245232	em_acmp: Fix missing ACMP input for EFR32xG24 devices. VDAC output have been added as valid ACMP positive or negative input selection sources.

Fixed in release 4.4.0.0

ID #	Description
1219359	em_eusart: Fixed missing support for C++.
1207190	em_core: Fixed missing dependency between components: when interrupt max disablement time was enabled, the cycle_counter driver must be added to the project. Otherwise, the compilation fails. To avoid this dependency the feature of cycle_counter is reimplemented in the em_core component to measure interrupt disable time.
1198594	em_msc: Fixed a bug in MSC_LockGetLocked function with GSDK 4.3.1 returning an invalid lock state leading to crash.

1160824	em_cmu: Added a return value to the CMU_HFXOCTuneSet() function. The application can check this value to see if the initialization parameter is valid or not.
1015203	em_lesense: Fixed the width of the structure for the storeCntRes bitfield in the LESENSE module.
177981	em_i2c: Fixed I2C_CR_MAX which affects the I2C clock equation.
1210216	em_iadc: Fixed high speed mode handling on devices family that has this mode, by correcting the maximum clock frequency computation.
1182892	Fixed missing C++ support in platform/emlib (added missing C linkage directive in headers).

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

None

4 Drivers

4.1 New Items

None

4.2 Improvements

Changed in release 4.4.1.0

- spidrv: Added conditions in validation script to check consistency in bitrate calculation.

Changed in release 4.4.0.0

- spidrv: Updated the eusart validation script to check the valid bitrate.

4.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1257321	Spidrv: Fixed HFXO clock validation error in SPI driver components.

Fixed in release 4.4.0.0

ID #	Description
1202645	Fixed missing USART/EUSART configuration for Si446x Radio component in Simplicity Studio pin tool GUI.
1177314	Added a condition to check the width of timers in PWM driver. Verifies the timers as 32-bit or 16-bit wide and compares with the given values.
1182892	Fixed missing C++ support in platform/emdrv (added missing C linkage directive in headers).

4.4 Known Issues in the Current Release

None

4.5 Deprecated Items

None

4.6 Removed Items

None

5 Services

5.1 New Items

Added in release 4.4.0.0

- Token Manager: Token specific APIs have been added to access the value of tokens.

5.2 Improvements

Changed in release 4.4.0.0

- In IOStream retarget STDIO, the assert on error status has been removed for a more adaptable behavior.
- Device Init DPLL: Added a validation script to correlate HFXO and DPLL Frequency and display a warning if they are incompatible.
- Sleeptimer: Updated the API function sl_sleeptimer_get_remaining_time_of_first_timer() to be more generic so that applications can use it to know how much time is left before the next wakeup event regardless of the Sleeptimer object type.

5.3 Fixed Issues

Fixed in release 4.4.2.0

ID	Description
1175116	hfxo_manager: Fixed HFXO_IRQn priority to be defined depending on CORE_ATOMIC_BASE_PRIORITY_LEVEL. HFXO's Interrupt priority was hardcoded to 2. This could possibly prevent the HFXO_IRQn from being serviced on time when CORE_ATOMIC_BASE_PRIORITY_LEVEL was changed in an application.
1256806	device init: Corrected Auto band RFFPLL settings for 928 MHz band.

Fixed in release 4.4.1.0

ID #	Description
1218121	device_init: added Pin Tool annotations for device_init_lfxo component. User can configure pin assignment in Simplicity Studio Pin Tool for LFXO.

Fixed in release 4.4.0.0

ID #	Description
1207966	Power Manager: Fix for DPLL that is not re-enabled after exiting from EM2 on the series 2 EFR32xG27. On EFR32xG27, the DPLL is disabled automatically when entering EM2, EM3. But exiting EM2, EM3 will not re-enable the DPLL automatically. Therefore, the software needs to re-enable the DPLL upon EM2 or EM3 exit. The local function dpllState() (em_emu.c) is called from EMU_Save() and EMU_Restore() to handle the DPLL state before entering EM2 or EM3 and after exiting EM2 or EM3. dpllState()-related code was not activated for EFR32xG27.
1220224	Device Init DPLL: Updated the default DPLL mode to cmuDPLLLockMode_Phase. Configured this way the frequency error will tend to 0, but the overshoot may be more important than with cmuDPLLLockMode_Freq. The previous default configuration (cmuDPLLLockMode_Freq) was causing some issues with certain applications requiring very precise frequency (Proprietary protocols or BLE).
1182892	Fixed missing C++ support in platform/service (added missing C linkage directive in headers).

5.4 Known Issues in the Current Release

None

5.5 Deprecated Items

None

5.6 Removed Items

None

6 CPC

6.1 New Items

Added in release 4.4.0.0

- First GA-quality release of CPC NVM3 module, which serves as a software component crafted to empower a Host device in leveraging the Non-Volatile Memory (NVM3) software of a Secondary device for the purpose of persistent data storage.
- Introducing the experimental version of CPC Primary, enabling communication between an MCU and a CPC Secondary device.
- Divided *sl_cpc_open_user_endpoint()* into two distinct APIs: *sl_cpc_user_endpoint_init()* and *sl_cpc_listen_endpoint()*. Note that the use of *sl_cpc_open_user_endpoint()* is slated for deprecation. Transitioning to the new APIs is recommended for future implementations.
- Separated *sl_cpc_close_endpoint()* into two distinct APIs: *sl_cpc_terminate_endpoint()* and *sl_cpc_free_endpoint()*. It is important to note that the use of *sl_cpc_close_endpoint()* is marked for deprecation. Transitioning to the new APIs is recommended for future implementations.

6.2 Improvements

Changed in release 4.4.2.0

- Enhanced CPCd logging by eliminating redundant and misleading data (re-sync mentions in the UART driver traces).

Changed in release 4.4.1.0

- CPCd now provides a helpful message and exits when presented with an invalid input argument.
- Enhanced the firmware upgrade functionality of CPC-Primary for improved performance and reliability.
- Achieved a minor performance enhancement to CPC-Secondary.
- The CPC-Secondary SPI driver has been upgraded to utilize IRQ line for flow control when RX buffers are exhausted. This enhancement effectively prevents the dropping of CPC acknowledgments and minimizes unnecessary re-transmissions.
- Added the CPC journal component to enables efficient logging of CPC events.
- Improved the CPCd EZSP firmware upgrade driver for enhanced speed and verbosity.

Changed in release 4.4.0.0

CPC has evolved to a connection-oriented protocol for endpoints, streamlining the determination of the connection status with remote endpoints. Notably, CPCd maintains backward compatibility with the previous version (Protocol V4), ensuring a smooth transition for existing implementations.

- In CPCd, eliminated dependency on sysfs or libgpiod for GPIO control.
- Restricted Libcpc APIs from being called in the reset callback.
- Enhanced the libcpc reset callback to consistently detect CPCd exits, ensuring detection even when the process is not blocked on a file descriptor or hasn't called a CPC API previously.
- Removed the limitation of one application process/libcpc per CPCd instance, providing increased flexibility and scalability.
- Adjusted CPC-GPIO-Expander Bridge dependency to reference the Rust libcpc bindings for CPCd 4.3.0.
- Optimized back pressure in CPC-GPIO-Expander Bridge for improved performance.
- Removed the multi-instance limitation for CPC-GPIO-Expander, allowing for more flexible and scalable usage.
- Enhanced CPC-GPIO-Expander logging to distinguish between different instances, providing clearer and more detailed information.
- Reduced CPC Secondary RAM Usage.
- Improved *sl_cpc_write()* to automatically block until resources are available (when using a kernel), eliminating the need for the application to handle the SL_STATUS_NO_MORE_RESOURCE error.

6.3 Fixed Issues

Fixed in release 4.4.1.0

ID #	Description
1251099	Resolved a memory leak, in CPC-Secondary, concerning the rx buffer handle that would occur during resynchronization when utilizing UART without hardware flow control.
1245850	Addressed an issue on CPCd where in re-transmission failed to occur during the shutdown of the remote endpoint.
1237395	Resolved a segmentation fault issue that occurred in CPCd when encountering a critical error with musl libc.

Fixed in release 4.4.0.0

ID #	Description
1219851	Resolved issues with CPCd on big-endian platforms.
1219510	Addressed potential segmentation fault in the UART validator of CPCd.
1212501	Fixed a race condition in cpc-gpio-expander/driver that occurs when a gpio chip is being removed while there is an ongoing gpio request.
1037832	Resolved a libcpc case where, if no reset_callback was provided to cpc_init, the application process would forcefully exit upon secondary reset.
1225166	Resolved, in CPC-GPIO-Expander, a concurrency issue that could lead to driver panics during exit cleanup or bridge initialization.
1218657	Capped SPI Bus frequency at 4 MHz to prevent peripheral incompatibility, especially when the bootloader utilizes a USART and CPC EUSART.

6.4 Known Issues in the Current Release

None

6.5 Deprecated Items

Deprecated in release 4.4.0.0

- *sl_cpc_open_user_endpoint()* is now deprecated. Transition to using *sl_cpc_terminate_endpoint()* and *sl_cpc_free_endpoint()* for future implementations.
- *sl_cpc_close_endpoint()* is now deprecated. Transition to using *sl_cpc_user_endpoint_init() and sl_cpc_listen_endpoint()* for future implementations.

6.6 Removed Items

None

7 Common

7.1 New Items

None

7.2 Improvements

Changed in release 4.4.0.0

- Improved MISRA compliance in platform/common, platform/emdrv, platform/emlib and platform/service.

7.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1266276	Fixed wrong assert statement in linkerfile_base.ld.jinja resulting in the error message "internal storage is excessing the flash size" being continuously displayed when the two components "Flash Storage Support" and "OTA Simple Storage Module" were present in the Zigbee Light examples.

Fixed in release 4.4.0.0

ID #	Description
1182892	Fixed missing C++ support in platform/common (added missing C linkage directive in headers).

7.4 Known Issues in the Current Release

None

7.5 Deprecated Items

None

7.6 Removed Items

Removed in release 4.4.0.0

- Hardware/driver/efp: Removed set regulator APIs that are marked as deprecated.
- Hardware/kit/common: Removed Energy Profiler SWO setup API.
- Uartdrv: Removed deprecated UART driver instance initialization structure.
- CMU, I2C, CRYPTO, WDOG: Removed APIs that are marked as deprecated.

8 Middleware

8.1 New Items

None

8.2 Improvements

None

8.3 Fixed Issues

None

8.4 Known Issues in the Current Release

None

8.5 Deprecated Items

None

8.6 Removed Items

None

9 Security

9.1 New Items

Added in release 4.4.2.0

- Added a configuration option (SL_VSE_BUFFER_TRNG_DATA_DURING_SLEEP) for optimizing power consumption of xG22 and xG27 devices using PSA Crypto for random number generation. When enabling this option (which is disabled by default), random bytes generated by the TRNG hardware will be buffered during EM2/EM3 sleep. This can result in fewer necessary startups of the TRNG, saving power consumption.

Added in release 4.4.0.0

- Updated the Mbed TLS library to version 3.5.0.
  - Various fixes and updates. See the official changelog at https://github.com/Mbed-TLS/mbedtls/releases/tag/mbedtls-3.5.0.
  - Added SLC components for enabling EC J-PAKE and PBKDF2 in PSA Crypto.
- In the SE Manager API, added functions for entering and exiting active mode on HSE based devices (xG21, xG23, xG24, xG25, xG28)

9.2 Improvements

None

9.3 Fixed Issues

Fixed in release 4.4.0.0

ID #	Description
1184675, 1184678	Fixed vulnerabilities in TrustZone secure services that allowed zero-length buffers to bypass parameter validation for IO buffers (IOVECs), allowing an attacker to potentially overwrite secure memory. Affected secure services are PSA Attestation service (Vault High only), SE Manager and NVM3. Affected devices are xG21, xG22, xG23, xG24, xG25, xG27, xG28. All GSDK versions with TrustZone support are affected.
1211537	For all PSA cipher drivers, in the sli_psa_validate_pkcs7_padding function, improved the implementation of time constant PKCS#7 padding handling for AES-CBC through PSA Crypto. Affected devices are xG1, xG11, xG12, xG13, xG14, xG21, xG22, xG23, xG24, xG25, xG27, xG28. Affected SDK versions are 4.3.2 and earlier.
1214862	Fixed an issue where the use of Link Time Optimization (LTO) could result in faulty code being generated for xG27 devices using PSA Crypto. The faulty code would result in a hard fault being triggered during the platform initialization routine. Affected device is xG27. Affected SDK versions are 4.3.2 and earlier.
1175222	In the EMLIB module, em_crypto.c (for series-1), renamed variables with the same name in the function CRYPTO_SHA_1 function, which caused GCC to report a build error when called with option -Werror=shadow. The issue is not known to cause any different behavior. Affected are series-1 based devices, xG1, xG11, xG12, xG13, xG14. Affected SDK versions are 4.3.2 and earlier.
1202804	In the CRYPTO PSA driver (for series-1), renamed variables with the same name in the static function sli_ccm_starts which caused GCC to report a build error when called with option -Werror=shadow. The issue is not known to cause any different behavior. Affected are series-1 based devices, xG1, xG11, xG12, xG13, xG14. Affected SDK versions are 4.3.2 and earlier.
691903	In the SE Manager GCM API, renamed variables with the same name in the function sl_se_gcm_multipart_starts, which caused GCC to report a build error when called with option -Werror=shadow. The issue is not known to cause any different behavior. Affected are HSE based devices, xG21, xG23, xG24, xG25, xG28. Affected SDK versions are 4.3.2 and earlier.

9.4 Known Issues in the Current Release

None

9.5 Deprecated Items

None

9.6 Removed Items

None

10 Operating System

10.1 New Items

None

10.2 Improvements

None

10.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1268061	Fixed HTTP server header parsing invalid pointer dereference vulnerability.

Fixed in release 4.4.1.0

ID #	Description
1199232	Micrium TCP-IP: Addressed Denial of Service (DoS) vulnerabilities by fixing issues related to ICMP and ICMPv6 parsing.
1118273	Micrium TCP-IP: Resolved a potential remote code execution vulnerability stemming from a heap overflow in the Micrium HTTP server.

Fixed in release 4.4.0.0

ID #	Description
1186138	Micrium OS: Fixed missing dependency on micriumos_common_lib in micriumos_common_core when compiling without optimization
1118273	Micrium OS: Fixed memory leak in cmsis_os2.c when creating new task.

10.4 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 a message to current task when called with a null pointer as argument.	N/A

10.5 Deprecated Items

None

10.6 Removed Items

None

11 Gecko Bootloader

11.1 New Items

None

11.2 Improvements

ID #	Description
830108	AES CTR STREAM BLOCK CFG component is now a part of the bootloader. The size of the stream block counter can be set by configuring this component. This is configurable, with 1, 2, 4, and 8-block options available. Decrypting GBL data takes less time as block size increases. The potential consequences of this improvement are as follows: Currently "pre-computing" additional keystream bytes for CTR and putting them in a RAM-accessible struct within the bootloader parser. Previously, it could only save as much as 16 bytes of keystream, but now that number may grow to 128. The key is not exposed, thus it is still difficult to forecast the next keystream, but the more bytes are stored, the greater risk becomes.

11.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1266811	Fixed compatibility issue of secure bootloader (standalone bootloaders) with internal storage and common storage component.

Fixed in release 4.4.1.0

ID #	Description
1251568	Added compatible bootloader examples for EFR32MG1B parts.

Fixed in release 4.4.0.0

ID #	Description
1210242	Updated the bootloader storage size in bootloader-storage-internal-single-1536k sample app for MG24 in UG489.

11.4 Known Issues in the Current Release

None

11.5 Deprecated Items

None

11.6 Removed Items

None

12 MVP Math Library

12.1 New Items

None

12.2 Improvements

None

12.3 Fixed Issues

None

12.4 Known Issues in the Current Release

None

12.5 Deprecated Items

None

12.6 Removed Items

None

13 Examples

13.1 New Items

None

13.2 Improvements

Changed in release 4.4.0.0

- Coulomb application: Improved API coulomb_calibrate() to check for the calibration completion after calling coulomb_calibrate.

13.3 Fixed Issues

Fixed in release 4.4.0.0

ID #	Description
1216731	Increased timeout iterations in i2cspm_baremetal to 500 iterations so that the sensor at 400kHz does not time out.
1043701	Configured the RESETn Pin with open drain mode for the standalone programmer in the dci_swd_programming example.

13.4 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.

13.5 Deprecated Items

None

13.6 Removed Items

None

14 Boards and External Devices

14.1 New Items

Added in release 4.4.0.0

Added support for the following new OPN's

- BRD4198B
- BRD4199B
- BRD4406A
- BRD4411A

14.2 Improvements

None

14.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1249898	Added support for board OPN BRD4328B.

14.4 Known Issues in the Current Release

None

14.5 Deprecated Items

None

14.6 Removed Items

None

15 Other Gecko Platform Software Components

15.1 New Items

Added in release 4.4.0.0

- Updated GCC compiler to 12.2.1
- Updated IAR compiler to 9.40.1

15.2 Improvements

None

15.3 Fixed Issues

None

15.4 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
	An issue is detected when updating the GCC toolchain from version 10.3 to 12.2, which increases the RAM usage by about 400 bytes in some cases when using stdio.	--

15.5 Deprecated Items

None

15.6 Removed Items

None

16 RAIL Library

16.1 New Items

Added in release 4.4.2.0

None

Added in release 4.4.1.0

- Added support for setting the Whitening and CRC initial values at runtime that are defined by the existing PHY's radio configuration.

Added in release 4.4.0.0

- Added support for a new assert, which will be thrown if the loaded PHY is not supported by the software-defined modem on the EFR32xG25.
- 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 on the EFR32xG23 and EFR32xG28 chips.
- Added an assert RAIL_ASSERT_INVALID_XTAL_FREQUENCY on the EFR32xG1x and EFR32xG2x chips, to fire when a radio config that is loaded is incompatible with a device due to the defined crystal frequency of the config not matching with the device's crystal frequency.
- Added the RAIL_TX_REPEAT_OPTION_START_TO_START option to measure the delay between repeated transmits from the start of TX to start of TX instead of the default from end of TX to start of TX.
- Added support for GCC 12.2.1 and IAR 9.40.1 compilers.
- Added support for Fast Channel Switching PHYs on the EFR32xG24.
- Added support for RAIL_IEEE802154_SUPPORTS_G_MODESWITCH on the EFR32xG28.
- Added support for the IEE802154 2.4 GHz coherent PHYs via RAIL_SUPPORTS_IEEE802154_BAND_2P4 on the EFR32xG28.
- Added a new RAIL_RxOptions_t option to enable collision detection on the EFR32xG25. Once enabled, when a collision with a strong enough packet is detected, the demod will stop the current packet decoding and try to detect the preamble of the incoming packet.
- Added support of RAILTEST for the MGM240x modules.
- Added support for channel masks during Wi-SUN mode switch in RAILTEST application on the EFR32xG25.
- Added support of 802.15.4 IMM-ACK with OFDM and OQPSK modulations (FCS 4 bytes only) on the EFR32xG25.
- Updated the RAIL Library component to automatically include the "RAIL Utility, Built-in PHYs Across HFXO Frequencies" component to better support different HFXO frequencies on supported parts.

16.2 Improvements

Changed in release 4.4.2.0

None

Changed in release 4.4.1.0

None

Changed in release 4.4.0.0

- Updated the default PTI rate to 3200000 bps on the EFR32XG25.
- No longer include rail_chip_specific.h and rail_features.h (anything dependent on em_device.h) when SLI_LIBRARY_BUILD is defined. This will allow the user to build their radio code in a way that depends on RAIL generically but is not chip-specific. When doing this the code cannot depend on things that are inherently chip-specific and are still in those files like RAIL_RF_PATHS, RAIL_NUM_PA, or any of the compile-time RAIL_SUPPORTS_xxx defines. Code will need to call appropriate runtime APIs or split itself between the generic and chip specific portions and build them separately.
- To better support chip-agnostic builds RAIL_TxPowerMode_t is now a superset representing all possible PAs across all chips. Any code depending on the number of or consecutive ordering of chip-supported PAs will likely need to be updated.
- To better support chip-agnostic builds RAIL_CalValues_t and subordinate RAIL_IrCalValues_t have grown to encompass the superset of fields needed on all chips, affecting all chips except EFR32xG25.
- To better support chip-agnostic builds RAIL_TransitionTime_t and therefore RAIL_StateTiming_t have grown to a superset type, affecting EFR32xG1.
- To better support chip-agnostic builds RAIL_FIFO_ALIGNMENT is now universally 32-bit, affecting EFR32xG1x and EFR32xG21, but is still only enforced on chips actually requiring that alignment.
- Updated the default power curves for the 10dBm High Power PA on EFR32xG24.

16.3 Fixed Issues

Fixed in release 4.4.2.0

ID #	Description
1248013	Fixed an issue with IEEE 802.15.4 High Data Rate (HDR) packets such that PTI now properly indicates they have a 2-byte 802.15.4 PHY header.
1255347	Fixed an issue with RAIL_SetTxFifo() setup that could trip RAIL_ASSERT_FAILED_UNEXPECTED_STATE_TX_FIFO when the prior TX FIFO was larger and contained more data than the new, smaller TX FIFO can hold. Any data in the old TX FIFO should have been ignored.
1271435	Fixed an issue where RAIL_WriteTxFifo() could write outside the TX FIFO when called without resetting the FIFO in dynamic multiprotocol applications.

Fixed in release 4.4.1.0

ID #	Description
1241800	Added the missing pa_dbm_mapping_table.py script to the release package. This script is used to help create power tables on the EFR32xG25 part.
1242723	Fixed an issue to exit from the critical section while performing multiprotocol PA operations on the EFR32xG25.
1243727	Improved the CCA fiability on the EFR32xg23, EFR32xg25 and EFR32xg28 chips.

Fixed in release 4.4.0.0

ID #	Description
1079816	Fixed a race condition on the 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.

ID #	Description
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.
1153679	Fixed an issue in "RAIL Utility, Coexistence" component on the EFR32xG24 where a GRANT signal pulse less than 100us might result in the radio not being properly placed in hold off after GRANT is deasserted.
1156980	Fixed an issue with channel hopping on the 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.
1175684	Fixed an issue with the RAIL_IDLE form of RAIL_Idle() and RAIL_STOP_MODE_PENDING form RAIL_StopTx() during the LBT/CSMA phase of a transmit which could previously hang. Pending LBT/CSMA and scheduled transmits are now stopped or idled with RAIL_EVENT_TX_BLOCKED triggered, except for idle mode RAIL_IDLE_FORCE_SHUTDOWN_CLEAR_FLAGS.
1183040	Updated all PHYs built into RAIL on Series 2 platforms using the latest radio calculator to reduce observed high reference spurs.
1184982	Fixed an issue with RAIL_StartAverageRssi() that could cause execution to hang in interrupt context just before raising RAIL_EVENT_RSSI_AVERAGE_DONE. This was primarily an issue on the EFR32xG21 platform. Note that it is possible, though unlikely, that RAIL_GetAverageRssi() might still return RAIL_RSSI_INVALID after the average RSSI period has finished.
1184982	Fixed an issue with RAIL_StartAverageRssi() that caused it to mistakenly idle the radio in the newly-activated protocol if a dynamic protocol switch occurred during the averaging operation of the suspended protocol.
1188083	Fixed an issue where RAIL_Idle() could hang waiting for the radio to idle when using IEEE 802.15.4 fast RX channel switching.
1190187	Fixed an issue where idling the radio before a Scheduled Receive window ends could cause a subsequent packet that should be silently filtered and rolled back to instead be received with RAIL_RX_PACKET_READY_CRC_ERROR.
1201506	Fixed an issue in multiprotocol applications where the wrong sync word would be used if two protocols used the same radio configuration and only one of those protocols set a custom sync word with RAIL_ConfigSyncWords() API.

16.4 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	

16.5 Deprecated Items

Note:

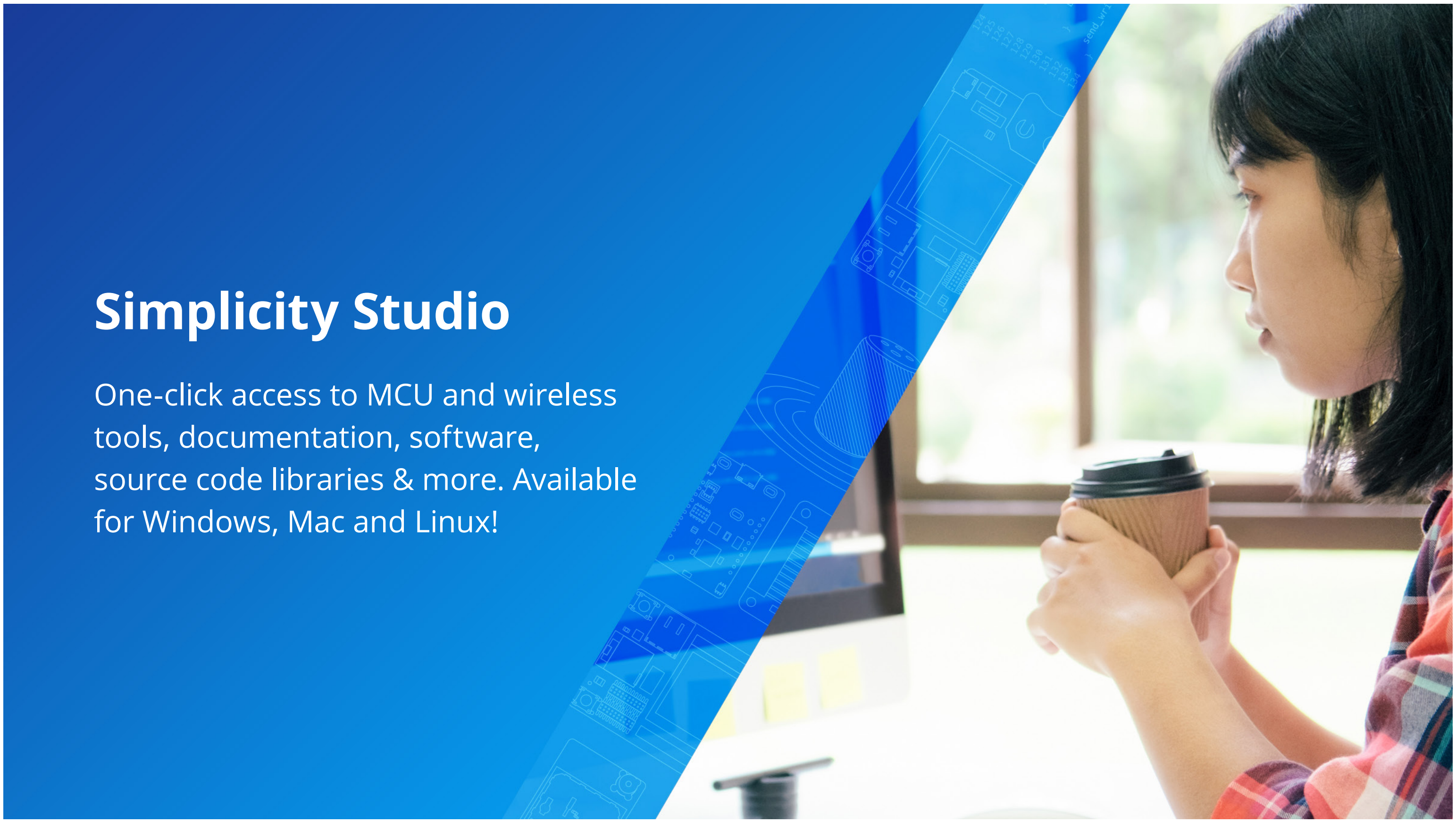
- The RAIL 2.x API is planned for deprecation in the 24Q4-GA release (December 2024). At that time, the new RAIL 3.0 API will be released for all supported chips along with a RAIL 2.x compatibility layer and migration guide.
  - The goal of this new API is to evolve what we currently have to get rid of some unused features, add better support for concurrent

listening use cases, and simplify channel and PA configurations.

- We hope the migration is straight forward and simple for the majority of customers, but there may be manual help needed in some cases which we will do our best to document to ease this transition.

16.6 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

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications. **Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project**

Trademark Information

SILICON LABS

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

www.silabs.com