Gecko Platform 4.2.3.0 GA

Gecko SDK Suite 4.2
May 3, 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.

Machine Learning includes TensorFlow Lite Micro components, used to run neural network inference, and related preprocessing components.

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 4.2.3.0 released May 3, 2023
Gecko Platform 4.2.2.0 released March 8, 2023
Gecko Platform 4.2.1.0 released February 1, 2023
Gecko Platform 4.2.0.0 released December 14, 2022

KEY FEATURES

- Support for EFR32xG25 devices
- Support for MGM240Dx modules
- Support for BGM24x modules
- Improved CPC UART drivers reliability
- Many CPC improvements
- Android support for CPCd

Security:

- Updates the Mbed TLS library to version 3.2.1, with support for CBAP.
- Updated ITS driver (version 3) for faster performance.
- Released software support for TrustZone, GA quality

Contents

1 CMSIS Device 2
2 Peripherals 4
3 Drivers....................6
4 Services 7
5 CPC 9
6 Common....................11
7 Middleware....................12
8 Security....................13
9 Operating System 17
10 Gecko Bootloader 18
11 Machine Learning 19
12 Examples....................20
13 Boards and External Devices....................21
14 Other Gecko Platform Software Components....................22
15 RAIL Library....................23

1 CMSIS Device

1.1 New Items

Added in release 4.2.3.0

- Added support for EFR32xG21 Rev. C and Rev. D devices

Added in release 4.2.2.0

- Added support for the following OPNs: EFR32BG24A010F1024GJ42-B, EFR32BG24B410F1536IJ42-B

Added in release 4.2.0.0

- Added support for EFR32xG25 devices
- Added support for EFR32xG21 Rev. C devices
- Added support for the BGM220SC23HNA module
- Added support for the MGM240SA22VNAx module
- Added support for the MGM240SD22VNAx module
- Added support for the BGM240SA22VNAx module
- Added support for the BGM240SB22VNAx module
- Added support for the BGM241SD22VNAx module

1.2 Improvements

Added in release 4.2.1.0

- The control bit DONESET in the LDMA_CH_CTRL register was renamed into DONEIEN to be consistent with other EFR32xG2x devices

1.3 Fixed Issues

Fixed in release 4.2.0.0

ID #	Description
1059733	ARMCC assembly startup files for Series 0 and Series 1 devices have been removed from GSDK since they have not been supported since GSDK v2.7
1064407	The PLL0, which provides an accurate 48 MHz clock to the USB, has been renamed for USBPLL0 in the EFR32xG25 header files.
1021831	Added EMU_RSTCAUSE_SETAMPER bit in device headers of EFR32xG21, EFR32xG23, EFR32xG24 and EFR32xG24 devices. This bit is asserted following a reset due to a tampering event detection.
1015647	Fixed the EXT_IRQ_COUNT macro value to match the number of external NVIC internal interrupt sources for EZR32HG, EZR32LG and EZR32WG devices.
1059733	ARM Startup files for Series 0 and Series 1 devices have been removed from GSDK since they have not been supported since GSDK v2.7

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 4.2.2.0

- Added a CMU_HFXOCoreBiasCurrentCalibrate in em_cmu to allow starting an optimization of the Core Bias Current of HFXO.

Added in release 4.2.0.0

- Added support for EFR32xG21 Rev. C in peripheral drivers (support for IADC internal voltage reference)
- Added support for the ETAMPDET peripheral
- The CMU_CLOCK_SELECT_SET macro was added as an alternative to the function CMU_ClockSelectSet, and may be used to reduce code size.
- Added new DisSyncOut field to TIMER_Init_TypeDef structure. This field can be used to disable the Timer's ability to start/stop/reload other timers.
- Added APIs to managed EFP interrupt in the EMU driver.
- Added APIs to access the HFXO CTune value in the CMU driver.

2.2 Improvements

Changed in release 4.2.0.0

- Added documentation for EMU_DCDCXxxx() functions
- Improved Eusart_Rx function when to account for Rx FIFO watermark

2.3 Fixed Issues

Fixed in release 4.2.0.0

ID #	Description
1043460	Fixed IADC_calcTimebase function to use the correct clock source. Time base calculation is now based off the correct clock source CLK_SRC_ADC rather than previous CLK_CMU_ADC. IADC is initialized with the correct time base and it is no longer required to manually set the correct time base value after initializing the IADC.
1037225	Fixed an issue where the EM4WU GPIO pin mode was configured as InputPull instead of InputPullFilter when configuring the EM4WU External Interrupt.
1033757	Device can still execute instructions after executing Enter_EM4() function is addressed with a new EMU_EnterEM4Wait() function. It waits after an EM4 entry request and makes sure no other code can run between the request and the moment the CPU is actually shut down.
1020645	Fixed EMU_BoostExternalShutdownEnable() API, BOOSTENCTRL bit logic was reversed.
1019940	Fixed EMU_RamPowerDown() end parameter behavior.
845272	Fixed a bug where the high-accuracy IADC offset correction calculation was wrong in the IADC initialization routine.
845232	Fix issue where the interrupts of the EFP would still be enabled after de-initializing the EFP driver.
1022566	Removed dmadrv_baremetal sample application for BRD2204C because it does not use usart0 peripheral for vcom.

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

None

3.3 Fixed Issues

Fixed in release 4.2.2.0

ID #	Description
1090509	Simple RBG PWM LED: Change type of the level parameter from uint8_t to uint16_t to prevent overflow errors.
1095611	SPIDRV: fixed sl_spidrv_init_instances() to select the right Chip Select control configuration for EUSART peripheral.

Fixed in release 4.2.0.0

ID #	Description
811009	Implemented KEYSCAN_E301 fix for EFR32xg2x devices (fix for Keyscan interrupts not cleared on unused rows).

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

None

4 Services

4.1 New Items

Added in release 4.2.0.0

- Added a new Device Initialization RFFPLL service to handle RFFPLL peripheral initialization.
- Added new API to iostream: sl_iostream_vprintf()

4.2 Improvements

Changed in release 4.2.3.0

- Removed direct call from IOStream context and added an API for sleep preparation for UART based instances in IOStream.

Changed in release 4.2.1.0

- RFFPLL configuration now defaults to use band *AUTO* band rather than defaulting to Band *9xx MHz* in the device_init_rffpll component. This allows picking up the RFFPLL configuration parameters from the radio configurator.
- Added a new API to get the sleeptimer's clock source's accuracy.

Changed in release 4.2.0.0

- Added support for 64-bit timestamps in sleeptimer wall clock APIs.
- Added the ability for sleeptimer service to use high frequency TIMER/WTIMER peripherals.
- sl_device_init_hfxo() will now use the CMU_HFXOINIT_EXTERNAL_SINE default settings if the "Device Init: HFXO" component's mode configuration is set to "External sine wave".
- Changed IOStream Uart interfaces reception to use DMA instead of receiving byte per byte.

4.3 Fixed Issues

Fixed in release 4.2.3.0

ID #	Description
1123649	Fixed an issue where dmadrv component would not report IOStream UART dependent in Simplicity Studio.
1124490	Fixed a corner case in IOStream that would cause the read_rx_buffer function to never return when reading data after the buffer was filled up.

Fixed in release 4.2.2.0

ID #	Description
1106720	Fixed incorrect size calculation for the buffer in the DMA Complete IRQ computes in IOStream UART implementations.
1098946	Fixed a segmentation fault when using power manager after de-initializing an IOStream instance.

Fixed in release 4.2.1.0

ID #	Description
1078350	Addressed an issue related to increased interrupt latency caused by the integration of IOstream UART DMA.
1092999	Resolved various static analysis issues identified in the IOstream and CLI components.
1081342	Corrected an issue where initial SWO data sent immediately after initializing SWO would become corrupted.

Fixed in release 4.2.0.0

ID #	Description
852037	Periodic sleeptimers no longer drift over time when using milliseconds as the time base.

1019822	Fixed sl_sleeptimer_get_datetime() so it returns time in Standard Time to match sl_sleeptimer_set_datetime() which sets the timezone.
1073997	Fixed sl_sleeptimer_convert_unix_time_to_ntp() to return an error when invalid time parameter is passed.
1068039	Fixed internal HFXO Manager settings applied when enabling SL_HFXO_MANAGER_SLEEPY_CRYSTAL_SUPPORT.
818102	Updated critical section used in Power Manager to explicitly use PRIMASK. This fixes an issue where the CPU would not wake up from sleep if em_core critical section are configured to use BASEPRI.
937666	Fixed IOStream Flush UART functionality for EUSART peripheral.
1065570	Fixed a bad usage of an enum during the preprocessor into the IOStream UART component.

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 4.2.1.0

- The bind argument (-b) can now be used together with restart argument (-r) to restart cpcd after a successful bind as well as restart cpcd if it was previously bound with a secondary.

Added in release 4.2.0.0

- New CPC Protocol version; CPCd and Secondary must be updated.
- Encryption can be disabled per endpoint; dictated by the secondary.
- Added new functionality to retrieve libcpc the firmware application version and the library version.
- Added new asynchronous event notification mechanism to libcpc.
- Added new API in libcpc to set endpoint options with type safety.
- libcpc's wrapper in Rust.
- libcpc instances are limited to one per task.

5.2 Improvements

Changed in release 4.2.2.0

- Removed the default values for SPI Chip Select, Interrupt, Reset, and Wake-Up pins. These pins must now be defined in the config file if they are needed.
- Wake-Up pin is no longer exported when SPI mode is selected unless it is explicitly set in the config file.
- Updated the timestamp format to be the same in cpcd & libcpc.
- Reduced the default number of user endpoints to 0 to reduce memory usage; User endpoint can be configured from 0 to 10.

Changed in release 4.2.1.0

- Improved handling for missing bootloader in firmware update mode.
- Added functionality to the CPC daemon for automatically generating a plaintext binding key if one is not present before binding.

Changed in release 4.2.0.0

- Improved CPC Secondary UART drivers' reliability and recovery when the hardware flow control is disabled.
- libcpc had a few changes when errors occur. Instead of returning -1 and setting "errno", it now returns negative values; negative values of "errno".
- Few improvements to cpcd cmake file; how to retrieve mbed TLS library and default compiler flags.
- Allow overriding socket directory / config path via CMAKE.
- Reduced minimum I-Frame retransmit timeout which also increase overall performance.
- Enhanced the tracing of the CPCd EZSP driver, used to send a firmware update to a CPC secondary using the Ember bootloader via SPI.
- CPCd can run on latest version of Android.
- Improved error reporting on bind failure.
- CPCd default binding key file location has been modified for "~/.cpcd/binding.key".
- Prevent clients from connecting if CPCd is not started in normal mode.
- CPCd displays a warning if run with root privileges.
- Improved CPCd's error handling when the bootloader interface is not present, and a firmware upgrade is requested.
- Improved CPCd config parser to allow comments on the same line.

5.3 Fixed Issues

Fixed in release 4.2.2.0

ID #	Description
1081547	Fixed memory leak when querying for secondary app version in Python and Rust libcpc implementations.
1092830	Fixed potential fatal on cpcd exit.
1092843	Fixed race condition when multiple processes/threads init/deinit event endpoints.
1095587	Fixed an issue that prevented the UART driver to recover from a sync loss if the contents of a payload had a HDLC flag within the last 7 bytes.
1105518	Fixed an issue where a packet could be retransmitted twice when it was not needed.

Fixed in release 4.2.1.0

ID #	Description
1085742	Resolved a crash issue in the secondary caused by receiving a packet before the kernel had started.

Fixed in release 4.2.0.0

ID #	Description
839674	Fixed in libcpc, the state returned by cpc_get_endpoint_state() after cpc_endpoint_close() has been called.
1040127	CPC security failed to initialize on EFR32MG13 and MG14 series parts. To work around this issue, mbedtls_entropy_adc() has been added as entropy source for these parts.
1057885	Fixed in libcpc, an issue that prevented users to call cpc_restart() again once it failed.
1064508	Fixed Secondary UART-EUSART driver with Hardware Flow control disabled and the security enabled.
1066686	Fixed a potential socket leak into cpc_init() when the daemon was not started.
1068063	Fixed an issue where the CS signal would not be de-asserted after successfully sending an SPI to the secondary device.
1066273	Fixed an issue where a retransmit from the secondary could cause an endpoint to close because the decryption would fail.
1067736	Fixed race condition with security endpoint when the client is actively connecting.

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

None

6.2 Improvements

Changed in release 4.2.0.0

- Linker sections are sorted by alignment to improve memory usage

6.3 Fixed Issues

Fixed in release 4.2.1.0

ID #	Description
1092995	Dedicated RAM section for SEGGER_RTT moved from the .bss section in the GCC linker script to the .data section. SEGGER_RTT section in .bss was causing .bss to be considered as initialized data and increased flash memory usage.

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

Changed in release 4.2.0.0

- Updated GLIB components to allow creation or use of a custom display implementation.
- Added configuration file to GLIB to reduce the memory footprint.

7.3 Fixed Issues

Fixed in release 4.2.0.0

ID #	Description
1015841	Fixed C++ compiler error with Glib.

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 4.2.2.0

- Released GA quality software support for TrustZone
  - Added Fault Injection hardening of the secure main function in the TrustZone Secure Key Library. The secure main function is responsible for initializing TrustZone on the Cortex-M33, which is crucial for secure operation. The Fault Injection hardening supports 3 profiles (low, medium and high), which enable different measures of hardening at the cost of additional code size for higher levels. See documentation in sl_fault_injection_hardening_cfg.h, which is available in Simplicity Studio accompanying the Fault Injection Hardening component.
  - Improved parameter validation in secure library services.
  - Partial defeaturing of the MSC secure service. Non-Secure applications are no longer allowed arbitrary writes to flash memory using the MSC service.
  - In the bootloader, retain a portion of RAM that the Non-Secure application might need after a softreset. The previous bootloader version cleaned the entire RAM except for the reset reason word when entering the application to make sure that no bootloader traces of RAM was left.
  - The TrustZone Secure Key Library requires the following SE / VSE firmware versions:
    - Version 1.2.14 is required for EFR32xG21 and EFR32xG22
    - Version 2.2.1 is required for EFR32xG23, EFR32xG24 and EFR32xG25

Added in release 4.2.0.0

- Updated the Mbed TLS library to version 3.2.1 with support for Certificate-Based Authentication & Pairing (CBAP).
- Added support for the TrustZone PSA initial attestation service. The Secure Key Library PSA attestation private key is stored in ITS with the UID 0xFFFFFFFFFFFFFFFF for Vault Mid devices. Hence, this UID should not be used for other purposes when the attestation service is enabled.
- Added RTOS support for the TrustZone Secure Key Library (TZ SKL) and TrustZone Gecko Bootloader. The TrustZone NSC interfaces support only one single thread to enter the secure libraries at any time by using a mutex, implemented with the CMSIS RTOS2 API. That is, there is no multi-thread concurrency allowed inside the secure libraries. The NonSecure application can run multiple threads concurrently. Currently FreeRTOS and MicriumOS are supported.
- Added ETAMPDET tamper source to SE manager for xG25.
- Added sl_se_get_tamper_reset_cause() function to SE Manager, available for Secure Vault High devices. The function lets the user find out if the last reset was caused by a Tamper event and which Tamper source caused the reset. Requires SE firmware version 2.2.1.
- sl_se_get_otp_version() is now available for VSE devices (as well as HSE devices). Requires VSE firmware version 2.2.1 / 1.2.14.
- Added support for non-blocksize input for AEAD multipart (fixed in GSDK version 4.1.1.0)
- Added support for partially overlapping input buffers for AEAD multipart (fixed in GSDK version 4.1.1.0)
- Added driver support for CCM multipart on xG21 devices (fixed in GSDK version 4.1.1.0)

8.2 Improvements

Changed in release 4.2.0.0

- Added a new version of the ITS driver (v3) in order to optimize the key look up time, especially for applications with more than 100 keys. ITS V3 uses simple PRNG with UID as a seed to create a hash-map-like structure. Note that NVM3 ID range was changed from region starting at 0x83100 to region ending with 0x870ff. We provide an upgrade path for the devices that already use the ITS driver. Note that after upgrading to ITS driver V3, SL_PSA_ITS_USER_MAX_FILES cannot be changed. Also note that upgrade can take some time (more than 1 min for 1000 files). New config options were added:
  - SL_PSA_ITS_SUPPORT_V2_DRIVER - This can be set to 0, which removes unused code, if the device has not used the ITS driver before in GSDK 4.1.x and earlier or the keys have already been migrated to v3.
  - SL_PSA_ITS_SUPPORT_V3_DRIVER - This can be set to 1 if the device has not used the ITS driver before in GSDK 4.1.x and earlier. This configuration option will add support for V3 driver, and provide an upgrade path from V1/V2. For devices that are not able to upgrade to ITS driver V3, we recommend disabling it, and using the previous version of the ITS driver.

  The downgrade path of the ITS driver to an earlier version is not provided.

- PSA Crypto components have now been updated to reflect the addition of Curve25519 support on xG21A and xG23A devices. Users that depend on this functionality and are running SE firmware versions older than 1.2.10 and 2.1.7, respectively, are recommended to update their firmware (preferred) or manually add the `Platform|Security|Mbed TLS|Elliptic Curves|Curve25519` component to their project.
- Added components for use of PKCS #5 and #12 modules in Mbed TLS
- Added support for configuring dynamic memory allocation functions in Mbed TLS via SLC components.
- Added support for configuration of alternative threading support in Mbed TLS (aka MBEDTLS_THREADING_ALT) via SLC components.
- Changed the TrustZone root key automatic key renewal. The key is now only renewed on Device Erase.
- Changed TRNG handling logic to reduce idle current consumption for xG22 in EM0/1.
- Added PSA API documentation links to Mbed TLS SLC components.

8.3 Fixed Issues

Fixed in release 4.2.2.0

ID #	Description
821834	Fixed key material leakages into RAM from intermediate buffers and registers in PSA drivers on Series-1 devices (EFR32xG1 and EFM32xG1) for AEAD, CIPHER and MAC operations. The issue affects all series-1 devices in any of the following scenarios: • When using the PSA oneshot CCM encryption and decryption functions: - sli_crypto_transparent_aead_encrypt_tag (called by the psa_aead_encrypt API) with PSA_ALG_CCM. - sli_crypto_transparent_aead_decrypt_tag (called by the psa_aead_decrypt API) with PSA_ALG_CCM. • The user provides key buffers not aligned on multiples of 4 bytes (32-bit) to any of the following: - the PSA AEAD, CIPHER and MAC drivers. - the classic MBEDTLS_AES_ALT-plugin (crypto_aes.c). - the internal RADIO stack crypto functions (sli_protocol_crypto_crypto.c). Affected SDK versions are 4.2.1 and earlier.

ID #	Description
1107715	Fixed key material leakages into RAM from intermediate buffers in the PSA drivers on EFR32xG21 and EFR32xG22. The leaked key material is the private key in plaintext when the key was not wrapped during psa_import_key (transparent mode). When the key is an ECC SECP521R1 private key and it was wrapped during psa_import_key (opaque mode) the private key material in plaintext is leaked in the psa_import_key function. The issue affects EFR32xG21 devices when using the SECP521R1 ECC curve for the following functions: • the PSA SE opaque import key function on High Vault leaks the private key material in plaintext. • sli_se_opaque_import_key is called by the psa_import_key API. • the PSA SE transparent sign message and sign hash functions. • sli_se_sign_message is called by the psa_sign_message API. • sli_se_sign_hash is called by the psa_sign_hash API. • the PSA SE transparent key agreement function. • sli_se_driver_key_agreement is called by psa_raw_key_agreement. Other PSA key agreement functions in the PSA Crypto API are also affected. The issue affects EFR32xG21 devices when using HMAC keys that are not multiple of 4 bytes for the following functions: • the PSA SE MAC compute function when using PSA_ALG_HMAC. • sli_se_driver_mac_compute is called by the psa_mac_compute API. The issue affects EFR32xG22 devices when using the following functions: • the PSA CRYTPOACC transparent key agreement function. • sli_cryptoacc_transparent_key_agreement is called by psa_raw_key_agreement. Other PSA key agreement functions in the PSA Crypto API are also affected. Affected SDK versions are 4.2.1 and earlier.
1110338	Fixed: Added ITS upgrade script which was missing from SDK version 4.2.1. The script is used to upgrade from ITS version 1 or 2 to version 3.

Fixed in release 4.2.1.0

ID #	Description
421759	Fixed EMLIB functions CRYPTO_SHA_1 and CRYPTO_SHA_256 to support input buffer pointers that are not aligned on 32-bit words. Affected devices are EFR32xG1x and EFM32xG1x. Affected SDK versions are 2.6.0-2.6.5, 2.7.0-2.7.11, 3.0.0-3.0.2, 3.1.0 - 3.1.2, 3.2.0 - 3.2.3, 4.0.0 - 4.0.2, 4.1.0 - 4.1.3 and 4.2.0
691903	Removed redeclarations of variables with the same name in the function sl_se_gcm_multipart_starts in the SE Manager component. Affected devices are EFR32xG21. Affected SDK versions are 3.1.0 - 3.1.2, 3.2.0 - 3.2.3, 4.0.0 - 4.0.2, 4.1.0 - 4.1.3 and 4.2.0

Fixed in release 4.2.0.0

ID #	Description
1074038	The AES-CTR one-shot encryption function of the PSA SE driver (sli_se_driver_cipher_encrypt) may fail when operating on input data that are not multiples of 16 bytes, and the alg argument is PSA_ALG_CTR or PSA_ALG_CCM_STAR_NO_TAG. The cipher oneshot encryption function of the PSA Crypto API psa_cipher_encrypt generates random IV values and subsequently calls sli_se_driver_cipher_encrypt. The failure depends on the combination of cipher key and random IV, where some random IV values may cause an incorrect ciphertext. The bug does not have exploitable properties and is not considered to be a security vulnerability. Affected devices are HSE-based devices (series-0/1 and VSE not affected). Affected SDK versions are 3.1.0 - 3.1.2, 3.2.0 - 3.2.3, 4.0.0 - 4.0.2 and 4.1.0 - 4.1.3
937813	Fixed an issue when the application requests random data and the radio is off, uninitialized or malfunctioning, then no entropy is available for the entropy accumulator of Mbed TLS and consequently the PRNGs (CTR-DRBG and HMAC-DRBG). The fix implements a fallback to request entropy from the ADC if the radio returns 0 bytes of entropy. The prerequisite is that the user must configure the ADC as entropy source. Affected devices are EFR32xG1x. Fixed in GSDK version 4.1.1.0
845005	Fixed issue on EFR32xG23 when DCI became unresponsive after running OTP initialization and Power-on-reset. In API function sl_se_init_otp reject setting reserved tamper sources to tamper levels other than "ignore" during OTP initialization by system manufacturer. Fixed in GSDK version 4.1.1.0

8.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
	The platform security software support has been moved from the folder util/third_party/crypto to platform/security. The submodules mbedtls and trusted-firmware-m are now located in util/third_party/.	This should not affect any projects that are using SLC components. However, if your project is not based on SLC components (which can be the case especially for projects that were created before Gecko SDK v3.0), you may need to update the source file locations in the project files.

.

8.5 Deprecated Items

None

8.6 Removed Items

Removed in release 4.2.0.0

- Remove deprecated multipart APIs with tag SL_DEPRECATED_API_SDK_3_3:
  - Deprecated CMAC multipart API in the SE Manager has been removed.
  - Deprecated GCM multipart API in the SE Manager has been removed.
  - Deprecated HASH multipart API in the SE Manager has been removed.

9 Operating System

No changes

10 Gecko Bootloader

10.1 New Items

None

10.2 Improvements

Changed in release 4.2.0.0

- Added automated support for executing post-build actions.
- Added support for upgrading from storage slots in a sequential order in a multi-slot configuration in an absence of a bootload list as opposed to an early exit. A configuration option is available in the Bootloader Storage Slot Setup component to enable/disable validation of the slot layout in case of a multi-slot configuration.

10.3 Fixed Issues

Fixed in release 4.2.2.0

ID #	Description
00297696	Added fix for display the SWO pin as None when debug prints are disabled in Debug component
00289764	Added changes for reserved space for reset reason at the start of RAM in linker file
00298080	Added fix for build error in BLE Apploader OTA DFU bootloader sample app for BG22

Fixed in release 4.2.1.0

ID #	Description
293766	Added fix for Bootloader custom projects app to show error when predefines pins are not defined

Fixed in release 4.2.0.0

ID #	Description
1043067	Added fix for XModem idle Timeout macro set to a non-default value in Bootloader - UART XMODEM sample app.
00289194	Added fix for Bootloader - SoC Bluetooth in-place OTA DFU application to display only for series 1 devices in Simplicity Studio.
00286863	Fixed an issue where the upgrade image size calculated was off by 4 bytes due to the CRC32 checksum that got appended to the image after the signature (fixed in version 4.1.1.0).

10.4 Known Issues in the Current Release

None

10.5 Deprecated Items

None

10.6 Removed Items

None

11 Machine Learning

11.1 New Items

None

11.2 Improvements

Changed in release 4.2.0.0

- Added support for more tflite micro operators in the slc generator script that autogenerates the OpResolver code.

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

Added in release 4.2.0.0

- Added ml_blink example showing a minimal tflite micro example blinking a led using a machine learning model. Meant to be used as a starting point for machine learning development.
- Added ml_magic_wand example. This example is based on a TensorFlow Lite micro example, but with an improved model to make gesture detection better.

12.2 Improvements

Changed in release 4.2.0.0

- Fixed suppression issue in voice_control_light and audio_classifier example, improving the classification stability.

12.3 Fixed Issues

Fixed in release 4.2.0.0

ID #	Description
1075130	Fixed issue in the ml_magic_wand example where the tflite arena size was too small.

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

Removed in release 4.2.0.0

- Removed tflite_micro_hello_world example, this example was replaced by the ml_blink example.
- Removed tflite_micro_magic_wand example, this example was replaced by the ml_magic_wand example.

13 Boards and External Devices

13.1 New Items

Added in release 4.2.3.0

Added support for the following new OPNs:

- BRD4195B
- BRD4196B

Added in release 4.2.2.0

Added support for the following new OPNs:

- BRD4115A
- BRD4332A

Added in release 4.2.1.0

Added support for the following new OPNs:

- BRD2704A

Added in release 4.2.0.0

Added support for the following new OPNs:

- BRD4318A
- BRD4335A

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

No changes

15 RAIL Library

15.1 New Items

Added in release 4.2.2.0

- Added RAIL_PacketTimeStamp_t::packetDurationUs field, which is currently set only on EFR32xG25 for received OFDM packets.

Added in release 4.2.0.0

- Added HFXO temperature compensation in RAIL on platforms that support RAIL_SUPPORTS_HFXO_COMPENSATION. This feature can be configured with the new `RAIL_ConfigHFXOCompensation()` API. The user will also need to be sure to handle the new RAIL_EVENT_THERMISTOR_DONE event to trigger a call to RAIL_CalibrateHFXO to perform the compensation.
- Added options in "RAIL Utility, Protocol" component to control whether Z-Wave, 802.15.4 2.4 GHz and Sub-GHz, and Bluetooth LE are enabled so that the user can save space in their application by disabling unused protocols.
- Added a new API `RAIL_ZWAVE_PerformIrcal` to help perform an IR calibration across all the different PHYs used by a Z-Wave device.
- Added 40 MHz crystal support on EFR32xG24 devices to the "RAIL Utility, Built-in PHYs Across HFXO Frequencies" component.
- Added support for IEEE 802.15.4 fast RX channel switching with the new `RAIL_IEEE802154_ConfigRxChannelSwitching` API on supported platforms (see RAIL_IEEE802154_SupportsRxChannelSwitching). This feature allows us to simultaneously detect packets on any two 2.4 GHz 802.15.4 channels with a slight reduction in overall sensitivity of the PHY.
- Added a new Thermal Protection feature, on platforms that support RAIL_SUPPORTS_THERMAL_PROTECTION, to track temperature and prevent transmits when the chip is too hot.
- Added new table-based OFDM and FSK PAs for EFR32xG25 based devices. The output power of these can be modified through a new customer provided look-up table. Ask support or look for an updated app note on how to configure the values in this table for your board.
- Added support for the MGM240SA22VNA, BGM240SA22VNA, and BGM241SD22VNA modules and updated the configurations for the BGM240SB22VNA, MGM240SB22VNA, and the MGM240SD22VNA.

15.2 Improvements

Changed in release 4.2.2.0

- 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.

Changed in release 4.2.1.0

- 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).
- The RAIL_SetTune() and RAIL_GetTune() functions now use the CMU_HFXOCTuneSet() and CMU_HFXOCTuneGet() functions respectively on EFR32xG2x and newer devices.

Changed in release 4.2.0.0

- The RAIL_ConfigRfSenseSelectiveOokWakeupPhy() will now return an error when run on the EFR32xG21 platform because this device cannot support the wakeup PHY.
- Updated the pa_customer_curve_fits.py helper script to accept floating point value for the maximum power argument, similar to the increment argument.
- Added support in "RAIL Utility, Coexistence" component for configuring priority options when directional priority is enabled but no static priority GPIO is defined.
- Broke up some EFR32xG12 802.15.4 dynamic FEC code to save code size for Zigbee and Blluetooth LE, which never need this functionality.
- Remove "RAIL Utility, Coexistence" component dependency from the RAIL Utility, Coulomb Counter component.
- The `RAIL_PrepareChannel()` function has been made dynamic multiprotocol safe and will no longer return an error if called when your protocol is inactive.

15.3 Fixed Issues

Fixed in release 4.2.3.0

ID #	Description
1058480	Fixed an RX FIFO corruption on EFR32xG25 that occurred when receiving/sending certain OFDM packets using FIFO mode.
1109993	Fixed an issue in the "RAIL Utility, Coexistence" component so that it simultaneously asserts request and priority if request and priority share the same GPIO port and polarity.
1118063	Fixed issue with recent RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE on EFR32xG13 and xG14 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.

Fixed in release 4.2.2.0

ID #	Description
747041	Fixed an issue on the EFR32xG23 and EFR32xG25 that could cause certain radio actions to delay for extended periods of time when the main core enters EM2 while the radio is still running.
1077623	Fixed an issue on EFR32ZG23 where multiple beam frames were lumped together on PTI as one large beam chain.
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. Previously this resulted in undefined behavior 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 PH,Y 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.
1103966	Fixed an unexpected Rx packet abort on the EFR32xG25 when using the Wi-SUN OFDM option4 MCS0 PHY.
1105134	Fixed an issue when switching 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.
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().

Fixed in release 4.2.1.0

ID #	Description
1077611	Fixed an issue on the EFR32xG25 that would cause a 40 µs porch before an OFDM TX.
1082274	Fixed an issue on the EFR32xG22, EFR32xG23, EFR32xG24, and EFR32xG25 chips that could cause the chip to lock up if the application attempted to re-enter EM2 within ~10 µs after wake-up and hit a <0.5 µs timing window. If hit, this lockup required a power on reset to restore normal operation to the chip.

Fixed in release 4.2.0.0

ID #	Description
843708	Moved function declarations from rail_features.h to rail.h to avoid a convoluted include dependency order.
844325	Fixed `RAIL_SetTxFifo()` to properly return 0 (error) rather than 4096 for an undersized FIFO.
845608	Fixed an issue with the `RAIL_ConfigSyncWords` API when using certain underlying demodulator hardware on EFR32xG2x parts.
851150	Fixed an issue on EFR32xG2 series devices where the radio would trigger RAIL_ASSERT_SEQUENCER_FAULT when PTI is used and GPIO configuration is locked. GPIO configuration can only be locked when PTI is disabled. See `RAIL_EnablePti()` for further information.

ID #	Description
857267	Fixed an issue when using the "RAIL Utility, Coexistence" component with TX abort, the signal identifier feature and DMP.
1015152	Fixed an issue on EFR32xG2x devices where RAIL_EVENT_RX_FIFO_ALMOST_FULL or RAIL_EVENT_TX_FIFO_ALMOST_EMPTY could trigger improperly when the event is enabled or the FIFO is reset.
1017609	Fixed an issue where PTI appended information could be corrupted when RAIL_RX_OPTION_TRACK_ABORTED_FRAMES is in effect when RAIL_IDLE_FORCE_SHUTDOWN or RAIL_IDLE_FORCE_SHUTDOWN_CLEAR_FLAGS is used. Also clarified that RAIL_RX_OPTION_TRACK_ABORTED_FRAMES is not useful with coded PHYs.
1019590	Fixed an issue when using the "RAIL Utility, Coexistence" component with BLE where the `sl_bt_system_get_counters()` function would always return 0 for GRANT denied counts.
1019794	Eliminated compiler warning in "RAIL Utility, Initialization" component when few of its features are enabled.
1023016	Fixed an issue on EFR32xG22 and newer chips where waits in between radio activity would consume slightly more power than necessary after the first 13 ms. This was especially noticeable when using `RAIL_ConfigRxDutyCycle` with large off time values.
1029740	Fixed issue where `RAIL_GetRssi()/RAIL_GetRssiAlt()` could return a "stale" RSSI value (the value was from previous RX state instead of the current one) if called quickly upon entering receive.
1040814	Added support to the "RAIL Utility, Coexistence" component for configuring the coexistence request priority on sync detect when using BLE.
1056207	Fixed an issue with IQ sampling when using the "RAIL Utility, AoX" component with only 0 or 1 antennas selected.
1062712	Fixed an issue where the "RAIL Utility, Coexistence" component would not always update request states correctly, which could lead to missed events triggered by new requests.
1062940	Prevented the "RAIL Utility, Coexistence" component from aborting BLE transmits when SL_RAIL_UTIL_COEX_BLE_TX_ABORT is disabled.
1063152	Fixed an issue where radio reception would not be fully cleaned up when a receive error occured with receive state transitions set to idle on error but transmit on success, a configuration mostly associated with BLE. On the EFR32xG24 this could cause a SYNTH calibration to not be properly restored and eventually cause the radio to stop working.

15.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

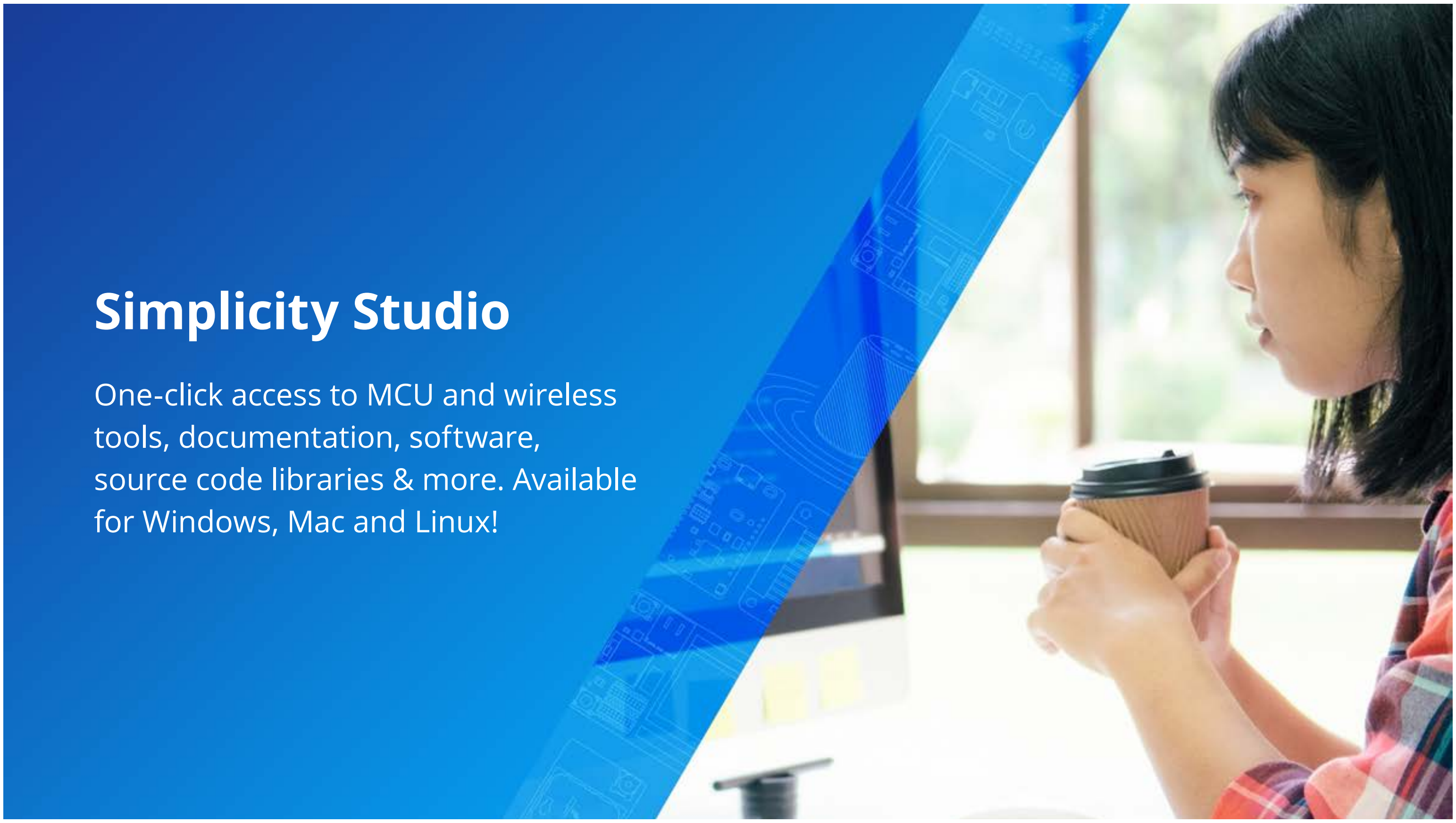
ID #	Description	Workaround
	Using direct mode (or IQ) functionality on EFR32xG23 requires a specifically set radio configuration that is not yet supported by the radio configurator. For these requirements, reach out to technical support who could provide that configuration based on your specification	
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

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