



Platform 5.0.1 GA

Simplicity SDK Suite 2024.6.1

July 24, 2024

The 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 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.
- **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 2 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):

Platform 5.0.1 released July 24, 2024
Platform 5.0.0 released June 5, 2024



KEY FEATURES

- Removed support for Series 0/1
- Support for EFR32xG26, EFR32xG22E and EFR32xG24 devices

Peripherals

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

Services

- Memory Manager, APIs to manage embedded dynamic memory allocations
- Clock Manager, for clock tree initialization
- Event System, for inter-process communication.
- Interrupt manager, APIs to manage embedded interrupts
- APIs for accessing value of tokens added to Token Manager

CPC

- Small improvement and bug fixes

RAIL

- Added support for collision detection feature on the EFR32xG25 parts
- Added support for additional Coex TX and RX metrics events to RAIL coexistence utility
- Updated some RAIL APIs to better prepare for future API changes while retaining backwards compile-time compatibility

Other Components

- Compilers upgraded (to GCC 12.2.1 and IAR 9.40.1)

Contents

- 1 New Items 3
- 2 CMSIS Device 4
- 3 Peripherals 6
- 4 Drivers 8
- 5 Services 9
- 6 CPC 12
- 7 Common 14
- 8 Middleware 15
- 9 Security 16
- 10 Operating System 18
- 11 Gecko Bootloader 19
- 12 MVP Math Library 20
- 13 Examples 21
- 14 Boards and External Devices 22
- 15 Other Gecko Platform Software Components 23
- 16 RAIL Library 24

1 New Items

Simplicity SDK is an embedded software development platform for building IoT products based on our Series 2 and Series 3 wireless and MCU devices. It integrates wireless protocol stacks, middleware, peripheral drivers, a bootloader, and application examples – a solid framework for building power-optimized and secure IoT devices.

The Simplicity SDK offers powerful features such as ultra-low power consumption, strong network reliability, support for a large number of nodes, and abstraction of complex requirements like multiprotocol and pre-certification. Additionally, Silicon Labs provides over-the-air (OTA) software and security updates to remotely update devices, minimize maintenance costs, and enhance the end-user product experience.

Simplicity SDK is a follow-on from our popular Gecko SDK, which will continue to be available providing long-term support for our Series 0 and Series 1 devices. For additional information on the Series 0 and Series 1 devices please reference: [Series 0 and Series 1 EFM32/EZR32/EFR32 device \(silabs.com\)](#).

2 CMSIS Device

2.1 New Items

Added in release 5.0.1:

- Added support for EFR32BG27C320F768I39

Added in release 5.0.0

- Added support for EFR32xG26 family.
- Added support for the following EFR32xG26 OPNs:
 - EFR32MG26B420F3200IM48-A, EFR32MG26B220F3200IM48-A, EFR32MG26B510F3200IM48-A, EFR32MG26B520F3200IM48-A, EFR32MG26B210F2048IM68-A, EFR32MG26B220F2048IM68-A, EFR32MG26B410F3200IM68-A, EFR32MG26B420F3200IM68-A, EFR32MG26B510F3200IM68-A, EFR32MG26B520F3200IM68-A, EFR32MG26B510F3200IL136-A, EFR32MG26B310F3200IL136-A, EFR32MG26B610F3200IM48-A, EFR32MG26B410F3200IM48-A, EFR32MG26B510F3200IL36-A
 - EFM32PG26B300F2048IM68-A, EFM32PG26B300F1024IM68-A, EFM32PG26B300F2048IL136-A, EFM32PG26B100F512IL136-A, EFM32PG26B300F1024IL136-A
 - EFR32BG26B310F2048IM48-A, EFR32BG26B320F2048IM48-A, EFR32BG26B310F2048IM68-A, EFR32BG26B320F2048IM68-A, EFR32BG26B310F2048IL136-A, EFR32BG26B310F1024IM68-A, EFR32BG26B320F1024IM68-A, EFR32BG26B310F1024IL136-A
- Added support for the following EFR32xG22 OPNs:
 - EFR32BG22E224F512IM32-C, EFR32BG22E224F512IM40-C
 - EFR32FG22E121F512IM32-C, EFR32FG22E121F512IM40-C
 - EFR32MG22E224F512IM32-C, EFR32MG22E224F512IM40-C

2.2 Improvements

None.

2.3 Fixed Issues

Fixed in release 5.0.1

ID #	Description
1318450	Added missing xG26 devices to pintool.
1260083	PRS: Added missing PRS Asynch channels in header file for each channel in xG24 and xG26 family.

Fixed in release 5.0.0

ID #	Description
1260083	PRS: Update missing PRS channel in xG24 and xG26.

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 5.0.0

None.

3.2 Improvements

Changed in release 5.0.0

- em_cmu: Added an API to retrieve the current XOANA/delta between HFXO XI and XO.
- Peripheral drivers: Renamed peripheral components and files.
 - Peripheral components have been renamed. Component peripheral_dcdc_coulomb_counter has been renamed to hal_dcdc_coulomb_counter, peripheral_etampdet has been renamed to hal_etampdet, peripheral_keyscan has been renamed to hal_keyscan, and peripheral_sysrtc has been renamed to hal_sysrtc.
 - The peripheral header files, sl_hal_keyscan.h, sl_hal_dcdc_coulomb_counter.h, sl_hal_etampdet.h and sl_hal_sysrtc.h have been created. The new header files can be included to replace peripheral_keyscan.h, peripheral_dcdc_coulomb_counter.h, peripheral_etampdet.h, and peripheral_sysrtc.h.
 - The peripheral compatibility header files, sl_hal_keyscan_compat.h, sl_hal_dcdc_coulomb_counter_compat.h, sl_hal_etampdet_compat.h and sl_hal_sysrtc_compat.h have been created. The new header files can be included to replace peripheral_keyscan_compat.h, peripheral_dcdc_coulomb_counter_compat.h, peripheral_etampdet_compat.h, and peripheral_sysrtc_compat.h.

3.3 Fixed Issues

Fixed in release 5.0.1

ID #	Description
1312889	em_cmu: Fixed cmuClock_LCDCLK undeclared compilation error for EFR32SG23B020F512IM40 OPN.
1286382	em_rmu, em_bus: Fixed compiler warnings for analyzer-shift-count-overflow and analyzer-null-dereference.

Fixed in release 5.0.0

ID #	Description
1285105	em_cmu: Moved CMU access from SYSTEM_ChipRevisionGet() to SYSCFG_readChipRev().
1245232	em_acmp: Fixed missing ACMP input for EFR32xG24 devices. VDAC output have been added as valid ACMP positive or negative input selection sources.
1268728	em_burtec: 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.

3.4 Known Issues in the Current Release

None.

3.5 Deprecated Items

Deprecated in release 5.0.0

- EMLIB: the CORE_IrqIsBlocked() function is deprecated and replaced by the interrupt manager sl_interrupt_manager_is_irq()_blocked function.

- EMLIB: the `CORE_NvicIRQDisabled()` function is deprecated and replaced by the interrupt manager `sl_interrupt_manager_is_irq_disabled` function.
- EMLIB: the `CORE_SetNvicRamTableHandler()` function is deprecated and replaced by the interrupt manager `sl_interrupt_manager_set_irq_handler` function.
- EMLIB: the `CORE_NvicMaskSetIRQ()` function is deprecated and replaced by the interrupt manager `sl_interrupt_manager_enable` function.

3.6 Removed Items

None.

4 Drivers

4.1 New Items

None.

4.2 Improvements

Changed in release 5.0.0

ID #	Description
1228849	Added compile-time checks for "NVM3_MAX_OBJECT_SIZE_LOW_LIMIT" and "NVM3_MAX_OBJECT_SIZE_HIGH_LIMIT" defines.

4.3 Fixed Issues

Fixed in release 5.0.1

ID #	Description
1321680	Spidrv: Fixed issue of SPIDRV instantiation using USART on EFR32XG25 despite the absence of USART on these platforms.

Fixed in release 5.0.0

ID #	Description
1257321	Spidrv: Fixed HFXO clock validation error in SPI driver components.
709964	<p>Added a function to register EM4WU Interrupts in emdrv which was used in the simple button driver to register a EM4WU Interrupt if the button pin supports EM4WU functionality.</p> <p>Since EM4WU interrupts are Level-triggered and not edge triggered, a rising edge interrupt is also registered along with the EM4WU to detect a button release along with the button press.</p>

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 5.0.0

- Added a new module called Clock Manager
 - It provides configuration of the different oscillators and the device clock tree through CMSIS Configuration Wizard Annotations C header files.
 - The module also provides an API to initialize the module and functions to perform a variety of oscillator and clock related operations. These include:
 - functions to fetch information about oscillators and clock branches frequency and precision.
 - functions to interface the CMU module functionality and features like enabling modules' bus clock, RCO calibration, retrieving or setting oscillator calibration values, output clocks on GPIO, etc.
 - The Clock Manager replaces the clock tree initialization that was done by the component `device_init_clocks`.
 - For more information about this new Clock Manager module and its capabilities, please go to <https://docs.silabs.com/gecko-platform/latest/platform-overview/>, section "Clock Manager".
- Added a new module called Memory Manager
 - It provides different APIs to manage embedded dynamic memory allocations.
 - Dynamic allocation/free: dynamically allocate and free memory blocks of various sizes.
 - Memory pools: reserve a chunk of memory large enough to hold a number of fixed-size blocks. This construct should only be used for models where a random dynamic memory allocation failure would be irrecoverable.
 - Dynamic reservation/free: similar to the dynamic allocation to reserve and free a block. A reserved block is meant to have a long-life duration during the application execution.
 - Memory regions: provides configuration and getter function of the C stack and heap for the supported toolchains.
 - It retargets the standard C library memory functions `malloc()/free()/calloc()/realloc()` to the Memory Manager ones.
 - It overloads the C++ standard `new/delete` operators to the Memory Manager `malloc()/free()`.
 - For C++ support with the GCC toolchain, the Memory Manager allows memory allocations to be done very early in static constructors. The same support is not yet present for C++ with the IAR toolchain.
 - The new Memory Manager replaces the component `silabs_core_sl_malloc` that was a simple wrapper on top of the standard C memory allocation functions.
 - For more information about this new Memory Manager module and its capabilities, please go to <https://docs.silabs.com/gecko-platform/latest/platform-overview/>, section "Memory Manager".
- Added a new module called Event System
 - It provides an event system for inter-process communication.
 - The event system is a software component enabling the development of event-driven software systems for Simplicity SDK-based firmware's by managing publisher creation, event dispatching/filtering, and event listening.
 - For more information about this new Event System module and its capabilities, please go to <https://docs.silabs.com/gecko-platform/latest/platform-overview/>, section "Event System".
- Added a new component called Code Classification
 - This component provides the `SL_CODE_CLASSIFY` macro to group functions into code classes. These code classes are the basis for code placement at link-time. Additionally, this component provides the `SL_CODE_RAM` macro to place application functions in RAM.
 - The current RAM functions support that allows to execute code from RAM via the macros `SL_RAMFUNC_XXXX()` has been ported to use the macros provided by the Code Classification component.
- Added a new module called Interrupt Manager
 - It provides different APIs to manage embedded interrupts. It allows devices to set the affinity of an interrupt, to enable/disable interrupts, set priority, and copy the vector table in RAM.
 - The Interrupt Manager replaces the component `device_init_nvic` that was handling the ARM Cortex-M Nested Vector Interrupt Controller (NVIC) and also replaces the component `ram_interrupt_vector_init` that was copying the interrupt vector table from flash to RAM and modifying the VTOR register to point to that location in RAM.
 - For more information about this new Interrupt Manager module and its capabilities, please go to <https://docs.silabs.com/gecko-platform/latest/platform-overview/>, section "Interrupt Manager".

5.2 Improvements

Changed in release 5.0.0

- Power Manager: Changed enter EM4 DVDD ramping algorithm for a dynamic algorithm with configurable tolerance on xG22 devices.
- Power Manager: Added optimization to peak current consumption when entering EM4 on xG22 devices.

5.3 Fixed Issues

Fixed in release 5.0.1

ID #	Description
1323169	Memory Manager: added a missing closure of critical section within the function <code>sl_memory_free()</code> in an early exit path. An issue can arise if <code>sl_memory_free()</code> is called twice with the same block leading to an infinite atomic section.
1289318	Sleep Timer: Fixed issue of Sleptimer not logging its EM2 requirement for debug by adding module name for Power Manager debug feature in the different Sleptimer HALs.
1290210	Power manager: Fixed issue of GPIO bus clock not being enabled, causing a bus fault, if the project doesn't have component initializing it. The GPIO bus clock is now enabled in <code>sl_power_manager_init()</code> .
1305241	Fix in the FreeRTOS Power Manager/Sleptimer integration code where the number of elapsed OS ticks during sleep was not evaluated properly, leading to the unnecessary creation of sleptimers with very small timeout values.

Fixed in release 5.0.0

ID #	Description
1275412	Fixed the Lynx FG22 OPNs to be defined as not supporting the precision mode in the LFRCO Oscillator by removing its selection in the configuration file of Simplicity Studio.
1213717	Sleptimer: Fixed <code>sl_sleptimer_set_time()</code> calculation. The function now calculates time using 64 bits variable to avoid loss of accuracy.
1175143	Power Manager: Added volatile keyword to variables that are modified in and outside interrupt context.
1175157	Sleptimer: Added volatile keyword to variables that are modified within an interrupt context.
1175146	HFXO Manager: Added volatile keyword to variables that are modified within an interrupt context.
1231208	Device init: Corrected Auto band RFFPLL settings for 928 MHz band on EFR32G25.
1175116	HFXO manager: Fixed HFXO_IRQn priority to be defined depending on <code>CORE_ATOMIC_BASE_PRIORITY_LEVEL</code> . HFXO's Interrupt priority was hardcoded to 2. This could possibly prevent the HFXO_IRQn from being serviced on time when <code>CORE_ATOMIC_BASE_PRIORITY_LEVEL</code> was changed in an application.
1242293	When <code>SL_CLI_ADVANCED_INPUT_HANDLING</code> is enabled then only the cli is extended with arrow handling, autocompletion of commands and delete backspace in the middle of words, if disabled it won't halt when arrow key is pressed.
1286438	CLI: Added customizable end-of-line configuration to the CLI instance. The CLI tasks can be disabled for seamless integration with external components.

5.4 Known Issues in the Current Release

None.

5.5 Deprecated Items

Deprecated in release 5.0.0

- The `ram_interrupt_vector_init` software component will be replaced by the new `interrupt_manager` component with `SL_INTERRUPT_MANAGER_S2_INTERRUPTS_IN_RAM` option to 1.
- Power Manager:

- Adding or removing Energy Mode requirement on EM2 using argument SL_POWER_MANAGER_EM2 with API functions `sl_power_manager_add_em_requirement()` and `sl_power_manager_remove_em_requirement()` is now deprecated. The calls to `sl_power_manager_add/remove_em_requirement(SL_POWER_MANAGER_EM2)` can simply be removed since the system will automatically go to deepsleep (EM2/EM3) in the absence of EM1 requirements.
- Defines `SL_POWER_MANAGER_EVENT_TRANSITION_ENTERING_EM3` and `SL_POWER_MANAGER_EVENT_TRANSITION_LEAVING_EM3` are now deprecated and should not be used in the `event_info` argument of function `sl_power_manager_subscribe_em_transition_event()`.

5.6 Removed Items

None.

6 CPC

6.1 New Items

Added in release 4.5.0

- Added an option to accelerate CRC calculation using a hash table, at the expense of increased code space.
- Blocking calls to `sl_cpc_read()` will now return `SL_STATUS_INVALID_STATE` if the read was aborted due to a state change disallowing reads on the endpoint, and `SL_STATUS_ABORT` if terminated by a user call to `sl_cpc_read_abort()`. Note that `SL_STATUS_EMPTY` may no longer be returned.

6.2 Improvements

Changed in release 4.5.0

- The CPC Secondary SPI driver has been updated to use COPI/CIPO naming for its bus signals. To ensure compatibility with existing project configurations, the previous TX/RX naming is still supported.
- CPCd now prints the `errno` when encountering an issue during the setup of its exit signals. Removed mentions of Re-sync from the CPCd logs.
- CPCd will now display a help message and terminate if given an invalid input argument.
- Improved CPC secondary performance by promptly releasing the `buffer_handle` upon becoming free, eliminating the need for the user to call `buffer_free()`.
- CPC Endpoints on the secondary now utilize dynamic allocation instead of relying on a static memory pool.
- The CPC Secondary SPI driver now leverages the IRQ line for flow control when it runs out of RX buffers. This enhancement prevents dropping CPC acknowledgments and reduces the occurrence of unnecessary re-transmissions.

6.3 Fixed Issues

Fixed in release 4.5.0

ID #	Description
1284893	Fixed a rare ASSERT issue in the CPC secondary driver that occurred when a frame was re-transmitted while being processed.
1276557	Fixed a rare memory leak in the UART CPC driver when using hardware flow control.
1237395	Fixed a segfault when CPCd encountered a fatal error on Alpine Linux.
1234611	Resolved minor bugs in CPCd related to <code>slist</code> removal, ensuring they do not affect the teardown path or normal operation logic.
1228447	Fixed an issue where the firmware upgrade process would hang if the host processed it too quickly.
1225330	Addressed a security vulnerability related to replay attacks.

6.4 Known Issues in the Current Release

None.

6.5 Deprecated Items

Deprecated in release 4.5.0

None.

6.6 Removed Items

None.

7 Common

7.1 New Items

Added in release 5.0.0

- Common: Added the `sl_slist_is_empty()` function to check if a list is empty.
- Common: Set the quality of the component `toolchain_gcc_lto` to evaluation as some of the components of the Simplicity SDK haven't been fully tested with linker LTO option.
- Common: Added a port implementation of `sl_core` for the ARM Cortex-M0+.

7.2 Improvements

Changed in release 5.0.0

- Readme.md: Updated the toolchain information with officially supported compiler versions.

Changed in release 5.0.0

- Common: The `.ram` section in the GCC linker script template, used for functions copied to and executed from RAM, was renamed to `.text_application_ram`.

7.3 Fixed Issues

Fixed in release 5.0.0

ID #	Description
1266276	GCC linker file: Fixed wrong assert statement in <code>linkerfile_base.ld</code> resulting in the error message "internal storage is exceeding 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.
1217337	Common: set next node to NULL in <code>sl_slist_remove()</code> function of singly-linked list. It is safer for users.
1260520	IAR linker file: Fixed issue where stack was not placed at the beginning of the RAM memory when compiling with IAR.
1260216	GCC linker file: Fixed issue where stack was not placed at the beginning of the RAM memory when compiling with gcc.
1234615	slist: Fixed erroneous function header note of the function <code>sl_slist_remove()</code> about throwing an assert

7.4 Known Issues in the Current Release

None.

7.5 Deprecated Items

None.

7.6 Removed Items

Removed in release 5.0.0

None.

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

Removed in release 5.0.0

- Removed MicriumOS components incompatible with Serie 2 & 3: File System, TCP-IP, CAN, USB, IO.
- Removed old Gecko USB and USBXpress stacks.

9 Security

9.1 New Items

Added in release 5.0.0

- Added a configuration option (called `SL_VSE_BUFFER_TRNG_DATA_DURING_SLEEP`, ref `psa_crypto_config.h`) 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, thus saving on power consumption.

9.2 Improvements

None.

9.3 Fixed Issues

Fixed in release 5.0.0

ID #	Description
1244117	<p>Bugfix for VSE based devices in which the TRNG may not have been initialized before generating an ECDSA nonce through the PSA Crypto API.</p> <p>Affected devices are xG22 and xG27.</p> <p>Affected SDK versions are all versions between 3.1.0.0 and 4.4.0.0.</p>
1248879 and 1295390	<p>Bugfix in the PSA ITS driver when</p> <ul style="list-style-type: none"> - running RTOS and - <code>SL_PSA_ITS_SUPPORT_V3_DRIVER=1</code> and - there exists a PSA key created with <code>SL_PSA_ITS_SUPPORT_V1_DRIVER=1</code> and/or <code>SL_PSA_ITS_SUPPORT_V2_DRIVER=1</code> <p>The PSA ITS driver will automatically attempt to upgrade the key to V3 but the calling thread will try to acquire the ITS mutex two times (without releasing it in between). However, the calling thread was blocked infinitely since the ITS mutex was not setup to be recursive/nested. The bugfix implements support for configuring any mutex (of type <code>mbedtls_threading_mutex_t</code>) in the SL Mbed TLS support components to be recursive/nested, and specifically configures the ITS mutex to be recursive/nested.</p> <p>Affected devices are xG21, xG22, xG23, xG24, xG25, xG27, xG28.</p> <p>Affected SDK versions are 4.3.x and 4.4.x (since the PSA ITS driver was made MT-safe and RTOS aware).</p>
1253097	<p>The VSE firmware upgrade application has been changed to not do any flash writes, which could prevent the VSE firmware from being upgraded when secure boot is enabled.</p> <p>Affected devices are xG22 and xG27.</p> <p>Affected SDK versions are 4.4.3 and earlier.</p>

9.4 Known Issues in the Current Release

SE Manager no longer uses `em_se.h` internally, as `em_se.h` will be deprecated in a future release. This implies that applications currently using APIs from both `sl_se_manager.h` and `em_se.h` may see an increase in code size. The recommended solution is to replace the usage of `em_se.h` with SE Manager. The APIs from `em_se.h` have been moved to `sli_se_manager_mailbox.h`.

9.5 Deprecated Items

None.

9.6 Removed Items

Removed the following EMLIB API functions from `em_se.h`. The functions have been deprecated since SDK v3.0 and in version 5.0.0 they are End-Of-Life.

End-Of-Life API Name	Signature	Replaced by (API in SE Manager)
SE_initOTP	SE_Response_t SE_initOTP(SE_OTPInit_t *otp_init);	sl_se_init_otp
SE_initPubkey	SE_Response_t SE_initPubkey(uint32_t key_type, void* pubkey, uint32_t numBytes, bool signature);	sl_se_init_otp_key
SE_writeUserData	SE_Response_t SE_writeUserData(uint32_t offset, void *data, uint32_t numBytes);	sl_se_write_user_data
SE_eraseUserData	SE_Response_t SE_eraseUserData(void);	sl_se_erase_user_data
SE_debugLockStatus	SE_Response_t SE_debugLockStatus(SE_DebugStatus_t *status);	sl_se_get_debug_lock_status
SE_debugLockApply	SE_Response_t SE_debugLockApply(void);	sl_se_apply_debug_lock
SE_debugSecureEnable	SE_Response_t SE_debugSecureEnable(void);	sl_se_enable_secure_debug
SE_debugSecureDisable	SE_Response_t SE_debugSecureDisable(void);	sl_se_disable_secure_debug
SE_deviceEraseDisable	SE_Response_t SE_deviceEraseDisable(void);	sl_se_disable_device_erase
SE_deviceErase	SE_Response_t SE_deviceErase(void);	sl_se_erase_device
SE_getStatus	SE_Response_t SE_getStatus(SE_Status_t *output);	sl_se_get_status
SE_serialNumber	SE_Response_t SE_serialNumber(void *serial);	sl_se_get_serialnumber

10 Operating System

10.1 New Items

None.

10.2 Improvements

None.

10.3 Fixed Issues

Fixed in release 5.0.0

ID #	Description
1213824	Sleeptimer: Fixed Micrium OS OSTimeDly() overflow error (RTOS_ERR_WOULD_OVF) which happens after the device has been operating for a long period of time with OS_OPT_TIME_PERIODIC option.
1108940	Fixed a bug in the Micrium CMSIS RTOS 2 layer where a memory leak would happen when a task would call osThreadTerminate on self. This is fixed when the micriumos_enable_task_exit_garbage_collect component is included in the project. The CMSIS RTOS 2 layer is to be used by Silicon Labs components only. For customer applications, it is recommended to call the Micrium OS or Free RTOS directly.

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

Added in release 5.0.1

- Added support for delta DFU upgrades. The Gecko Bootloader now accepts GBLs with Delta Patches for applications. These updates can be enabled by adding the, 'GBL Delta DFU' component in the bootloader.

11.2 Improvements

None.

11.3 Fixed Issues

Fixed in release 5.0.0

ID #	Description
1276505	Fixed the issue of accessing Global Variable when "Bootloader Spi Controller EUSART Driver" component is enabled.

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 5.0.0

None.

13.3 Fixed Issues

Fixed in release 5.0.0

None.

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

Removed in release 5.0.0

- MCU examples: Removed MCU SDK sample applications for EFM32 and EZR32 as it only supported Series 0 & 1 parts. Series 0 & 1 parts have been removed from the Simplicity SDK 2024.6.

14 Boards and External Devices

14.1 New Items

Added in release 5.0.0

Added support for the following new OPN's boards:

- BRD2606A
- BRD2608A
- BRD2710A
- BRD4415A
- BRD4116A
- BRD4117A
- BRD4118A

14.2 Improvements

None.

14.3 Fixed Issues

Fixed in release 5.0.0

None.

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 5.0.0

None.

15.2 Improvements

None.

15.3 Fixed Issues

ID #	Description
1151558	Updated timeout_in_ticks value to UINT32_MAX before starting the timers.

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

Removed in release 5.0.0

- Removed series 0 and 1 parts support from the Simplicity SDK 2024.6.
 - All CMSIS device OPN, peripheral, startup code files under `gsdk/platform/Device/SiliconLabs` have been removed.
 - All pin tool files under `/gsdk/platform/hwconf_data/pin_tool/` have been removed.
 - All series 0/1 metadata have been removed.

16 RAIL Library

16.1 New Items

Added in release 5.0.1

- Added support of automatic LNA bypass on the EFR32xG25 parts.

Added in release 5.0.0

- Added support of collision detection for concurrent PHYs on the EFR32xG25 parts.
- Added support for additional Coex TX and RX metrics events to the "RAIL Utility, Coexistence" component.

16.2 Improvements

Changed in release 5.0.1

None.

Changed in release 5.0.0

- Updated a few APIs to return RAIL_Status_t which were previously returning void.
- Updated RAIL_IEEE802154_WriteEnhAck and RAIL_WriteAutoAckFifo() which formerly took uint8_t ackDataLen parameter now take uint16_t ackDataLen parameter.
- Added missing runtime equivalent API RAIL_SupportsPathDiversity() of RAIL_SUPPORTS_PATH_DIVERSITY.

16.3 Fixed Issues

Fixed in release 5.0.1

ID #	Description
1271542	Fixed an issue on EFR32xG21 where an inopportune-timed RAIL_StopTx() using RAIL_STOP_MODE_ACTIVE can hang a subsequent CSMA/LBT transmit during its clear-channel assessment (CCA).

Fixed in release 5.0.0

ID #	Description
1242662	Fixed incorrect default values for SL_RAIL_UTIL_PA_CALIBRATION_ENABLE on the EFR32xG24 and EFR32xG25 parts, along with an issue where Power Amplifier calibrations were not properly disabled when SL_RAIL_UTIL_PA_CALIBRATION_ENABLE was disabled.
1255524	Fixed a dynamic multiprotocol issue where RAIL_BLE_Init() was called (which implicitly establishes the BLE 1Mbps PHY), but after a protocol switch returns to BLE, its RAIL_BLE_ConfigChannelRadioParams() settings were not being properly reapplied.
1256045	Fixed an issue where Signal Identifier options were being errantly displayed in Simplicity Studio for the "RAIL Utility, Coexistence" component on platforms that don't support the Signal Identifier feature.
1265376	Fixed an issue in the UC components where the RAIL power manager support could be initialized after the radio was started in certain example applications. This would lead to radio state and power mode requirements being improperly tracked, as you must initialize power manager support before any radio operations.
1266003	Fixed an issue where RAIL_BLE_ConfigChannelRadioParams() was improperly overriding the BLE Quoppa PHY whitening configuration.

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

- 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 get rid of some unused features, add better support for concurrent listening use cases, and to simplify channel and PA configurations.
 - The migration is intended to be straight forward and simple for the majority of customers. However, in some cases manual help might be required to ease this transition.

16.6 Removed Items

None.