# Gecko Platform 4.5.1.0 GA

## Gecko SDK Suite 4.5.1

## August 19, 2026

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
- Peripherals
- Drivers
- Services
- CPC (Co-Processor Communication)
- Common
- Middleware
- Security
- Operating System
- Gecko Bootloader
- MVP Math Library
- Examples
- Boards and External Devices
- Other Gecko Platform Components.
- RAIL (Radio Abstraction Interface Layer)

**Continued Software Support for Your Silicon Labs Devices**

At Silicon Labs, we're committed to delivering dependable, well-supported software for all actively supported hardware products.

The GSDK 4.5 LTS is a long-term support release tailored specifically for **Series 0 and Series 1 devices** that are **not compatible with the Simplicity SDK**. This release ensures ongoing maintenance and stability for devices that are still in active use (i.e., not marked as NRND or EOL).

For customers using **Series 2 or Series 3 devices**, all new features and updates will be provided through the **Simplicity SDK**. To take advantage of the latest innovations and continued support, we encourage you to adopt or migrate to the most recent Simplicity SDK release.

If you have questions about migration or need help choosing the right SDK for your product, our support team is here to help.

These release notes cover SDK version(s):

Gecko Platform 4.5.1.0 released August 19, 2026

Gecko Platform 4.4.7.0released October 8, 2025

**KEY FEATURES**

- The GSDK 4.5 LTS is a long-term support release tailored specifically for Series 0 and Series 1 devices that are not compatible with the Simplicity SDK. This release ensures ongoing maintenance and stability for devices that are still in active use (i.e., not marked as NRND or EOL).

**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)


**Copyright © 2026 by Silicon Laboratories**

**Contents**

1 New Items ..... 3
2 CMSIS Device ..... 4
3 Peripherals ..... 5
4 Drivers ..... 7
5 Services ..... 9
6 CPC ..... 11
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

The **GSDK 4.5 LTS** is a long-term support release tailored specifically for **Series 0 and Series 1 devices** that are **not compatible with the Simplicity SDK**. This release ensures ongoing maintenance and stability for devices that are still in active use (i.e., not marked as NRND or EOL).

For customers using **Series 2 or Series 3 devices**, all new features and updates will be provided through the **Simplicity SDK**. To take advantage of the latest innovations and continued support, we encourage you to adopt or migrate to the most recent Simplicity SDK release.

# 2 CMSIS Device

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

## 2.1 New Items

None.

## 2.2 Improvements

None.

## 2.3 Fixed Issues

**Fixed in release 4.5.1.0**

| ID # | Description |
|---|---|
| 1556184 | Updated sl_gsdk_version.h to reflect the correct version. |
| 1257561 | Fixed an IOStream issue on EFM32GG11, EFM32TG11, and EFM32GG12 where selecting UART could cause the application to hang. Updated initialization to correctly support both UART and USART modules. |

**Fixed in release 4.4.4.0**

| ID # | Description |
|---|---|
| 1260083 | PRS: Added missing PRS Asynch channels in header file for each channel in xG24 and xG26 family. |

**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

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

## 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 Fixed Issues

**Fixed in release 4.4.7.0**

| ID # | Description |
|---|---|
| 1415521 | Fixed Errata 220 in em_emu for Series 1 devices.<br>Possible lock-up issue when using voltage scaling when coming back from EM2 sleep. To fix the issue, a 14us delay is added coming back from deepsleep when voltage scaling is used. |
| 1516693 | Em_emu: Fixed a bug where a device can fail to enter EM4 as expected after the EM4 entry sequence has been written. The WFI instruction has been added to the EMU_EnterEM4 function after the EM4 entry sequence finishes. |

**Fixed in release 4.4.5.0**

| ID # | Description |
|---|---|
| 1286933 | em_ldma: Removed unnecessary assert in LDMA_Init and LDMA_StartTransfer. |
| 1337998 | em_ldma: On EFR32XG25 devices, the MATCHEN and MATCHVAL fields in the LDMA descriptors described in the em_ldma driver were inverted.<br>This issue was fixed so that the structure properly represents the EFR32XG25 hardware. |

**Fixed in release 4.4.4.0**

| ID # | Description |
|---|---|
| 1286382 | em_rmu, em_bus: fixed compiler warnings for analyzer-shift-count-overflow and analyzer-null-dereference. |

**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). |

## 3.3 Known Issues in the Current Release

None.

## 3.4 Deprecated Items

None.

## 3.5 Removed Items

None.

# 4 Drivers

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.

## 4.1 New Items

None.

## 4.2 Improvements

**Changed in release 4.4.6.0**

- uartdrv: Fixed several static analysis warnings.

**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 4.4.6.0**

| ID # | Description |
|---|---|
| 1371534 | uartdrv: Added an assert for baudrate before caclulating ticks.<br>sleeptimer: cast timeout_ms to uint64_t to avoid overflow and move the Null check of handle in sl_sleeptimer_stop_timer before derefrencing. |

**Fixed in release 4.4.5.0**

| ID # | Description |
|---|---|
| 1284781 | UARTDRV: Fixed issues in UARTDRV_Abort() leading to incorrect number of EM1 requirements. Fixed handling of enableRxWhensleeping. |
| 1361482 | UARTDRV: Fixed a bug in UARTDRV_Abort() where the callback function was called passing a NULL pointer instead of a pointer to the data buffer. |

**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 Improvements

**Changed in release 4.4.6.0**

- Emlib: Improved code size footprint by using SET, CLR and TGL register aliases instead of read-modify-write binary operations.

**Changed in release 4.4.5.0**

- em_qspi: Added QSPI_Reset() function to set QSPI's registers to original state.

**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.

## 4.5 Fixed Issues in the Current Release

None.

## 4.6 Deprecated Items

None.

## 4.7 Removed Items

None.

# 5 Services

Services includes common services such as NVM3 and Power Manager.

## 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**

- Sleeptimer: Fixed several static analysis warnings.

**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 4.4.6.0**

| ID # | Description |
|---|---|
| 1371534 | uartdrv: Added an assert for baudrate before caclulating ticks<br>sleeptimer: cast timeout_ms to uint64_t to avoid overflow and move the Null check of handle in sl_sleeptimer_stop_timer before derefrencing |

**Fixed in release 4.4.5.0**

| ID | Description |
|---|---|
| 1336478 | Sleeptimer: Fixed an issue where the callback would be incorrectly called when the callback execution time exceeded the period of a periodic timer. |
| 1251562 | HFXO Manager: Fixed a corner case where the HFXO startup time measurement could be started with a wrong start measurement, leading to a bad value. In a previous version of the SiSDK, the bad start-up values were discarded, but the assertion in sl_hfxo_manager.c – Line 199 would fail. This correction fixes the root cause of this issue. |

**Fixed in release 4.4.4.0**

| ID | Description |
|---|---|
| 1305241 | Fix in the FreeRTOS Power Manager/Sleeptimer integration code where the number of elapsed OS ticks during sleep was not evaluated properly, leading to the unnecessary creation of sleeptimers with very small timeout values. |

**Fixed in release 4.4.3.0**

| ID | Description |
|---|---|
| 1285915 | Fixed the Pin Assignment issue which prevents the configuration header files from being created/updated when using PinTool. |

**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.<br>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

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

## 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 4.4.6.0**

| ID # | Description |
|---|---|
| 1378738 | Enhanced reliability of CPC SPI driver at high-speed operation. |
| 1380066 | Fixed a bug where CPC daemon incorrectly believed the interrupt pin was not asserted in time. |

**Fixed in release 4.4.5.0**

| ID # | Description |
|---|---|
| 1355407 | Fixed an issue where some Linux SPI drivers would skip over a transfer of 0 bytes. This caused the primary and secondary to lose synchronization and cause the IRQ to be stuck abnormally long in the de-asserted state. |
| 1307636 | Fixed a rare race condition in the CPCd SPI driver on systems with high interrupt latency, which could lead to CPC system loosing communication with the host. |

**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

Common components are used throughout the SDKs,

## 7.1 New Items

None.

## 7.2 Improvements

**Changed in release 4.4.4.0**

- Readme.md: Updated the toolchain with correct compiler versions.

**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

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.

## 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

Security includes mbed TLS and other security services.

## 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

**Added in release 4.5.0.0**

- Added separate components for the legacy Mbed TLS SHA224 and SHA384 algorithms in order to save code space when only a subset of SHA224, SHA256, SHA384 and SHA512 is needed by the application. In previous releases, SHA224 and SHA256 were combined in the same component, called mbedtls_sha256.slcc, which represented the legacy sha256.h interface, which implements both SHA224 and SHA256. Likewise, SHA384 and SHA512 were combined in the same component, called mbedtls_sha512.slcc, which represented the legacy sha512.h interface, which implements both SHA384 and SHA512.

## 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.<br>Affected secure services are PSA Attestation service (Vault High only), SE Manager and NVM3.<br>Affected devices are xG21, xG22, xG23, xG24, xG25, xG27, xG28.<br>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.<br>Affected devices are xG1, xG11, xG12, xG13, xG14, xG21, xG22, xG23, xG24, xG25, xG27, xG28.<br>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.<br>Affected device is xG27.<br>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.<br>The issue is not known to cause any different behavior.<br>Affected are series-1 based devices, xG1, xG11, xG12, xG13, xG14.<br>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.<br>The issue is not known to cause any different behavior.<br>Affected are series-1 based devices, xG1, xG11, xG12, xG13, xG14.<br>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.<br>The issue is not known to cause any different behavior.<br>Affected are HSE based devices, xG21, xG23, xG24, xG25, xG28.<br>Affected SDK versions are 4.3.2 and earlier. |

## 9.4 Known Issues in the Current Release

- Several use cases of Mbed TLS and PSA Crypto fail to build on Series 0 devices.

## 9.5 Deprecated Items

None.

## 9.6 Removed Items

None.

# 10 Operating System

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

## 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

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.

## 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.<br>The potential consequences of this improvement are as follows:<br>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

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.

## 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

Examples are example applications illustrating platform functionality.

## 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

Boards and External Devices cover supported hardware.

## 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

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

## 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

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.

## 16.1 New Items

**Added in release 4.5.1.0**

- Added new modes to the RAIL_BLE_ConfigSignalIdentifier() and RAIL_IEEE802154_ConfigSignalIdentifier() APIs to configure the signal identifiers for BLE and IEEE 802.15.4 protocols, respectively.

**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 4.4.4.0**

- Reduced the delay between TX completion and the start of PA ramp down on the EFR32xG24.

**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 super-set 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 4.5.1.0**

| ID # | Description |
|---|---|
| 1618746 | Fixed an issue on the EFR32xG24 where the IEEE 802.15.4 2.4 GHz antenna diversity PHY could lose approximately 6 dB of RX sensitivity when configured for a fixed antenna (ANTENNA0 or ANTENNA1) instead of AUTO. Also updated AGC settings on the IEEE 802.15.4 OQPSK production and diversity production PHYs to reduce gain recovery time and improve Wi-Fi coexistence. |

**Fixed in release 4.4.7.0**

| ID # | Description |
|---|---|
| 1439301 | Fixed an issue in dynamic multiprotocol applications where a radio does not enter into EM2 sleep mode when no active tasks are being performed. |
| 1450400 | Fixed an issue on EFR32xG23, EFR32xG25, EFR32xG26 and EFR32xG28 where RAIL was blocked in RAIL_RF_STATE_RX_ACTIVE. |
| 1499282 | Fixed an issue where RAIL could hang in its idle code path when switching from the IEEE 802.15.4 High Data Rate (HDR) PHY to the normal 802.15.4 PHY. |

**Fixed in release 4.4.6.0**

| ID # | Description |
|---|---|
| 1315679 | Fixed a multiprotocol issue where a custom sync word set for one channel could override another channel's sync word after a DMP switch back to that protocol. This change reverts earlier fix attempts by documenting this limitation: if multiple protocols share the same radio configuration. The setting of custom sync words in any of these protocols should be avoided, as they can again leak into another protocol sharing the same radio configuration. |
| 1377955 | Fixed a DMP issue where the device would incorrectly switch to a background receive task with insufficient time to switch back to an upcoming high priority scheduled task. |
| 1380153 | Fixed an issue where calling \ref RAIL_Idle() could prevent any future reception on the EFR32xG25 for OFDM-FSK concurrent PHYs. |

**Fixed in release 4.4.4.0**

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

| | |
|---|---|
| 1306597 | Fixed a CCA issue on EFR32xG28 where, in some cases, the channel could be reported as busy even though it was free. |

**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. |
| 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. |

| ID # | Description |
|---|---|
| 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 |
|---|---|---|
| 1335868 | On EFR32xG28, when large data is written and read back using rxFifoManualRead API, it fails to read even though the data is ready to be read. | On EFR32xG28:<br>If customers want to use the packet length up to 2058 bytes, the value of BUFFER_POOL_ALLOCATOR_BUFFER_SIZE_MAX needs to be set to 2102 for device_sdid_235 in RAIL - SoC RAILtest application. |
| | 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:<br>Wi-SUN FSK mode 1a exhibits a PER floor with frequency offsets around ± 8 to 10 KHz<br>Wi-SUN FSK mode 1b exhibits a PER floor with frequency offsets around ± 18 to 20 KHz | |

## 16.5 Deprecated Items

- None.

## 16.6 Removed Items

None.

SILICON LABS
CONNECTED INTELLIGENCE

# The Leading Innovator in Low-Power Wireless.

**IoT Portfolio**
silabs.com/products

**Simplicity Studio**
silabs.com/simplicity

**Quality**
silabs.com/quality

**Support & Community**
silabs.com/community

---


**Disclaimer**

Silicon Labs provides customers with data sheets, application notes and product briefs ("Product Documentation") that are intended to present current, accurate and in-depth information about Silicon Labs' integrated circuits, peripherals, modules and other products ("Products"). Silicon Labs may also provide reference designs, schematics, design files, software, tools, documentation, consultation and other customer support materials ("Support Materials") that are intended solely to assist customers in developing their own modules, boards or other applications or products that include one or more Silicon Labs Products and may not be used for any other purposes.

Characterization data, device specifications, memory sizes, memory addresses and typical parameters can and do vary in different applications and are provided for illustrative purposes only, are subject to change without notice, and should not be relied upon without independent verification. Product firmware may be updated during the manufacturing process for security or reliability purposes.

Product Documentation does not imply or expressly grant any license to design or fabricate integrated circuits or modules. Customers are solely responsible for the design and distribution of their products including the acquisition of licenses to any necessary intellectual property rights.

Product Documentation and Support Materials are not authorized for use in or in connection with FDA Class III devices or military, automotive or aviation applications and shall in no circumstances be used in weapons of mass destruction or systems capable of delivering such weapons.

SILICON LABS DOES NOT WARRANT THAT PRODUCT DOCUMENTATION OR SUPPORT MATERIALS WILL MEET YOUR REQUIREMENT OR THAT THEIR OPERATION WILL BE UNINTERRUPTED OR THAT ANY SOFTWARE WILL BE ERROR-FREE. PRODUCT DOCUMENTATION AND SUPPORT MATERIALS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." SILICON LABS MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, REGARDING THE ACCURACY, COMPLETENESS, RELIABILITY, OR SUITABILITY OF PRODUCT DOCUMENTATION OR SUPPORT MATERIALS, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SILICON LABS DISCLAIMS ALL IMPLIED WARRANTIES.

**Trademark Notice**

Silicon Laboratories®, Silicon Labs®, Silabs®, and the Silicon Labs S-logo as well as other product or service names used herein are trademarks or registered trademarks of Silicon Laboratories Inc. For more information, please visit: Silicon Labs Trademark Use Guidelines - Silicon Labs. Arm®, Cortex® and Keil® are trademarks or registered trademarks of Arm Holdings or Arm Limited. All other third-party products or brand names mentioned herein are trademarks of their respective holders.


---


SILICON LABS CONNECTED INTELLIGENCE

400 West Cesar Chavez
Austin, TX 78701 | USA

www.silabs.com