Gecko Platform 3.1.2.0 GA

Gecko SDK Suite 3.1
April 7, 2021

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.

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.

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 3.1.2.0 released April 7, 2021
Gecko Platform 3.1.1.0 released January 27, 2021
Gecko Platform 3.1.0.0 released December 9, 2020

KEY FEATURES

Peripherals

- Include handling pcntModeOvsQuadx modes in PCNT_Init

Services

- New NVM3 function that enables reading only parts of a data object

Common

- Platform converted to use BASEPRI interrupt levels, vs. PRIMASK interrupt enable/disable
- MCU SDK 6.0 integrated into Common platform.

Security

- Mbed TLS updated to version 2.24.0
- Added attestation support in SE Manager

Operating System

- Platform components are now compatible with any kernel providing a CMSIS-RTOS2 implementation.

Examples

- Added alpha quality Hello World and Micro Speech examples for TensorFlow Lite for Microcontrollers.
- Added several common sample apps for EFM and EFR devices.
- Removed several MCU sample apps.

Boards and External Devices

- Added better support for 60 Ohm ESR crystals.

Other Platform Components

- Added alpha support for TensorFlow Lite for Microcontrollers version 2.3.1.

Contents

1 CMSIS Device 2
2 Peripherals 3
3 Drivers....................4
4 Services 5
5 Common....................7
6 Middleware 8
7 Security 9
8 Operating System 11
9 Gecko Bootloader 12
10 Examples....................13
11 Boards and External Devices....................15
12 Other Gecko Platform Software Components 16
13 RAIL Library....................17
14 Legal 20

1 CMSIS Device

1.1 New Items

Added in release 3.1.1.0

- Added support for new OPN: EFR32ZG14P731F256GM32.

Added in release 3.1.0.0

- Added support for new OPN: EFM32PG22.

1.2 Improvements

None

1.3 Fixed Issues

None

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

None

2.2 Improvements

Changed in release 3.1.0.0

- CMU_OscillatorTuningGet() and CMU_OscillatorTuningSet() can now be used on Series 2 for HFXO Core Bias Current Optimization, in order to skip Optimization stage when needed.

2.3 Fixed Issues

Fixed in release 3.1.1.0

ID #	Description
653065	Fixed an issue where TIMER_MaxCount() returned the wrong value for 32-bit timers on Series 2 devices.

Fixed in release 3.1.0.0

ID #	Description
481036	Fixed issue with the SL_FLOOR macro.
482546	Include handling of the pcntModeOvsQuadx modes in PCNT_Init.
625661	Fixed EMU_DCDCModeSet() function on xG22, where it could hard-fault if trying to set the DCDC converter to bypass when it was already in bypass mode.
634635	Fixed an issue on BG22 where MSC_EccConfigSet() could lead to a hard fault. The initialization of the ECC is now done properly to avoid unwanted read or write operations while initializing.

2.4 Known Issues in the Current Release

None

2.5 Deprecated Items

None

2.6 Removed Items

None

3 Drivers

3.1 New Items

None

3.2 Improvements

Changed in release 3.1.2.0

- Clarified "SWO debug" component can be used to enable code correlation profiling.

3.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
677261	Fixed a bug which caused the TEMPDRV driver and example applications to be available for some boards or MCUs that do not support it.
674419	Fixed missing dependency to “status” component in UARTDRV.
674173	Fixed issue in SPIDRV preventing creation of a usart or eusart SPIDRV instance in Simplicity Studio.

Fixed in release 3.1.0.0

ID #	Description
399547	Fixed an issue with i2cspm with the lockup recovery happening as part of the initialization where the frequency of the pulse could be too fast and where the clock toggle ended with the clock signal being low instead of high. The lockup recovery function should now work as expected by standard I2C.

3.4 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
485403	UARTDRV operation in EM2 does not work when using EUART.
520149	When creating SPIDRV instances using the Simplicity Studio Project Configurator, code to initialize the instance is not automatically generated and must instead be added manually.

3.5 Deprecated Items

None

3.6 Removed Items

None

4 Services

4.1 New Items

Added in release 3.1.0.0

- Added new NVM3 function, nvm3_readPartialData, which enables reading only parts of a data object.

4.2 Improvements

Changed in release 3.1.2.0

- Added validation so that in Simplicity Studio an error message will appear if trying to configure the Sleep Timer component to use the SYSRTC peripheral when SYSRTC is not available on the selected device.

Changed in release 3.1.1.0

- Added better documentation to the new HFXO Manager module.

Changed in release 3.1.0.0

- Added the option to re-call sl_power_manager_init() if clock settings have changed, so that the Power Manager updates its internal settings and cached values.

4.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
672440	Fixed a notification issue in Power Manager where a notification from EM2 would wrongly not be emitted when an asynchronous event would wake-up the system in a certain window frame when a scheduled wake-up is also present. This issue led to RAIL module being lock up and radio activities not properly working after that.
661412	Added new “iostream_uart” component to fix IO Stream usage on series0 and series1 boards with UART and USART peripherals.

Fixed in release 3.1.1.0

ID #	Description
656678	Fixed an issue in the Power Manager so that it can always work properly in debug mode.
656291	Fixed an issue in the Power Manager when an asynchronous event would add an EM1 requirement from an ISR just after the Power Manager early wakeup for a scheduled event. In that case, the Power Manager would wrongly notify EM1 entry before the full HF clock was restored.

Fixed in release 3.1.0.0

ID #	Description
519083	Added initialization of the DPLL in device init for xG22 devices.
652396	Fixed an issue in the Power Manager where the HFXO startup delay was added to calculations of schedule wake-ups even when HFXO was not being used, which caused the system to wake up too early without needing to.

4.4 Known Issues in the Current Release

None

4.5 Deprecated Items

None

4.6 Removed Items

None

5 Common

5.1 New Items

None

5.2 Improvements

Changed in release 3.1.1.0

- Added a section in the documentation on docs.silabs.com to provide more details on the Atomic Sections and BASEPRI.

Changed in release 3.1.0.0

- The Gecko SDK has switched to use BASEPRI method for atomic section by default, when module_init_nvic component is present. This allows to have interrupt at a higher priority than the Gecko SDK's atomic sections, in order to meet strict latency requirements. Interrupts that are not masked by the Gecko SDK's atomic sections should not interact with anything in the Gecko SDK. For existing applications not relying on the module_init_nvic component, the behavior does not change.

5.3 Fixed Issues

None

5.4 Known Issues in the Current Release

None

5.5 Deprecated Items

None

5.6 Removed Items

None

6 Middleware

6.1 New Items

None

6.2 Improvements

None

6.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
658370	Fixed potential wrap-around issue in Micrium OS LIB Mem when creating dynamic memory pools.
651259	Fixed security vulnerability in Micrium OS HTTP Server related to HTTP Multipart Form Header processing.

Fixed in release 3.1.0.0

ID #	Description
644559	Added support for TRNG; TCP sequence is now generated with a true entropy source and it follows the specifications outlined in RFC 6528.

6.4 Known Issues in the Current Release

None

6.5 Deprecated Items

None

6.6 Removed Items

None

7 Security

7.1 New Items

Added in release 3.1.2.0

- Added support for the new PSA Crypto configuration options, MBEDTLS_PSA_CRYPTO_CONFIG, PSA_WANT_xxx and MBEDTLS_PSA_ACCEL_xxx, in the UC components for PSA Crypto. That is, the psa_crypto component is slimmed down, and a few new psa_crypto_xxx components have been added. This enables finer-grained configuration of PSA Crypto, e.g. for excluding unused code, fallback to mbed TLS library, and tune code size. Additional PSA Crypto configuration options for even better configuration granularity will be added to releases in the near future.
- In psa_crypto_ecdh and psa_crypto_ecdsa components, added auto-inclusion of fallback to standard mbed TLS libraries when one or more non-accelerated ECC curves are included.
- A new entropy source is implemented for this release. The new source can provide a device-unique non-volatile seed on devices which do not have other hardware entropy sources available. Due to the requirement of being able to store and update this seed, this implementation depends on the presence of NVM3. For more details, see the description of the component "mbed TLS Support for non-volatile entropy seed", which fulfils the requirement of having at least one entropy source available for CTR-DRBG and entropy collector capabilities.
- Added support for hardware-accelerated HMAC through PSA Crypto.

Added in release 3.1.0.0

- mbed TLS library updated to version 2.24.0 and moved to path util/third_party/crypto/mbedtls/.
- The new version of mbed TLS does not allow simultaneous use of both TRNG and RAIL entropy sources.
- Add API for retrieving attestation tokens through SE Manager

7.2 Improvements

Changed in release 3.1.2.0

- When using Simplicity Configurator to configure mbed TLS, the cipher modes CBC, CTR and OFB are no longer automatically included when adding AES support to the project. These modes can now be individually selected or deselected.
- Reduced entropy stack usage by providing MBEDTLS_CTR_DRBG_MAX_SEED_INPUT configuration.
- Multiple other minor code size and RAM usage improvements.

Changed in release 3.1.0.0

- Initialization of the TRNG peripheral has been improved.
- For the IAR toolchain, added forced optimization mode 'balanced' for certain functions in file bignum.c.

7.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
679916	In the psa_import_key() function, allow import of un-clamped Montgomery keys.
679435	In the PSA Crypto HKDF component, always include HKDF fallback to mbed TLS because the accelerated HKDF driver currently cannot operate standalone.
672219	When using mbed TLS with the default ECC acceleration settings, MBEDTLS_ECP_NIST_OPTIM will no longer be included when the only ECC curves requested are curves which can be accelerated by hardware. Previously, including MBEDTLS_ECP_NIST_OPTIM unnecessarily would cause an approximate 1.5 kB of code size overhead.
667205	In the psa_import_key() function, allow import of un-clamped Montgomery keys.
658868	In PSA Crypto API function psa_key_derivation_setup(), return PSA_ERROR_NOT_SUPPORTED from when no KDF is enabled.

ID #	Description
651954	Set MBEDTLS_MPI_MAX_SIZE correctly to support ECC curves secp384r1 and secp521r1 when selected with UC components mbedtls_ecc_secp384r1.slcc and mbedtls_ecc_secp521r1.slcc. In config-device-acceleration.h, add support for ECC curves secp384r1 and secp521r1 combined with ECDH and/or ECDSA for Series-2 'A' devices (treated like CURVE25519 and other non-accelerated curves).

Fixed in release 3.1.1.0

ID #	Description
651954	Set MBEDTLS_MPI_MAX_SIZE correctly to support ECC curves secp384r1 and secp521r1 when selected with UC components mbedtls_ecc_secp384r1.slcc and mbedtls_ecc_secp521r1.slcc. In config-device-acceleration.h, added support for ECC curves secp384r1 and secp521r1 combined with ECDH and/or ECDSA on standard SE (non-Vault) devices (treated like CURVE25519 and other non-accelerated curves).
654612	Added support for the new PSA Crypto configuration options, MBEDTLS_PSA_CRYPTO_CONFIG, PSA_WANT_xxx and MBEDTLS_PSA_ACCEL_xxx, in the UC components for PSA Crypto. That is, the psa_crypto component is slimmed down, and a few new psa_crypto_xxx components have been added. This enables finer-grained configuration of PSA Crypto, e.g. for excluding unused code, fallback to mbedtls library, and tune code size. Additional PSA Crypto configuration options for even better configuration granularity will be added in the near future releases.

7.4 Known Issues in the Current Release

None

7.5 Deprecated Items

None

7.6 Removed Items

Removed in release 3.1.2.0

Software fallback algorithms that were introduced as a workaround for a bug in EFR32xG21 SE Firmware versions before v1.2.2 are now removed by default. See security advisory A-00000201, available through your customer account. This means ECDH operations and public key validity checks will by default return an error code on EFR32xG21 parts with SE firmware before v1.2.2. It is recommended to upgrade to the latest SE firmware instead of re-enabling the software workaround. However, in case these fallback methods are still needed, they can be turned back on by defining the symbol 'SL_SE_SUPPORT_FW_PRIOR_TO_1_2_2'.

8 Operating System

8.1 New Items

Added in release 3.1.0.0

- Added CMSIS-RTOS2 support on Micrium OS Kernel. Converted Platform Services to rely on CMSIS-RTOS2 APIs, allowing them to run on either Micrium OS Kernel, FreeRTOS or other kernels providing a CMSIS-RTOS2 implementation.
- Added the option to select between Micrium OS Kernel or FreeRTOS in the Project Configurator Software Components tab in Simplicity Studio. Wireless stack developers should consult with the wireless stack documentation or release notes to confirm support for the chosen kernel.

8.2 Improvements

Changed in release 3.1.2.0

- Updated the version of FreeRTOS from 10.3.0 to 10.4.3 to fix a bug on Cortex-M33 device where the interrupts could be disabled when entering the first scheduled task.
- User is now warned of potential misconfiguration when using FreeRTOS without device_init_nvic component.

Changed in release 3.1.0.0

- In the version of FreeRTOS included with the Gecko SDK, we have added support for staying in sleep on ISR exit if no preemption occurs.

8.3 Fixed Issues

Fixed in release 3.1.1.0

ID #	Description
657560	Fixed an issue in Micrium OS Kernel where a null pointer could be de-referenced when entering idle and Round-Robin scheduling is enabled and used. Fixed an issue in Micrium OS Kernel where a null pointer could be de-referenced when OS_CFG_TASK_PROFILE_EN is set to DEF_ENABLED and a scheduling is performed while being idle and returning to idle.
651155	Fixed an issue in Micrium OS OSTimeDly() where interrupts would stay disabled if ‘delay_ticks’ was greater than ‘OSDelayMaxTick’.

8.4 Known Issues in the Current Release

None

8.5 Deprecated Items

None

8.6 Removed Items

None

9 Gecko Bootloader

9.1 New Items

Added in release 3.1.2.0

- Added support for EFM32PG22 products.

Added in release 3.1.0.0

- Storage bootloaders will park in a busy wait loop if both the installed application image and the images in the storage slots are corrupt.

9.2 Improvements

None

9.3 Fixed Issues

None

9.4 Known Issues in the Current Release

The Bootloader SPI-EZSP example only builds successfully for parts whose design supports a Serial Peripheral Interface (SPI). See the datasheet for your part to verify if it supports SPI.

9.5 Deprecated Items

None

9.6 Removed Items

None

10 Examples

10.1 New Items

Added in release 3.1.0.0

- Added common applications that can be used on EFM and EFR devices:
  - Blink (with Micrium OS Kernel, with FreeRTOS, with PWM on one LED).
  - DMADRV
  - IO Stream (EUSART with and without Micrium OS Kernel, USART with Micrium OS Kernel)
  - mbedTLS (AES Crypt, ECDH, ECDSA)
  - MEMLCD (with and without Micrium OS Kernel) (based on previous LCDDRV example)
  - Simple MPU
  - NVM3
  - Power Manager (both with and without Micrium OS Kernel)
  - SE Manager (Asymmetric Key Handling, Block Cipher, ECDH, EC J-PAKE, Hash, Host Firmware Upgrade, KDF, Key Provisioning, SE Firmware Upgrade, Secure Debug, Secure Identity, Signature, Stream Cipher, Symmetric Key Handling, Tamper, User Data)
  - Simple Button (with and without Micrium OS Kernel)
  - Sleep Timer (Timer, Wall-clock)
  - SPIDRV (master with and without Micrium OS Kernel, slave with and without Micrium OS Kernel)
  - TEMPDRV (with Micrium OS Kernel)
  - UARTDRV (with and without Micrium OS Kernel)
  - WiFi Commissioning using a WF200 (with Micrium OS Kernel)
- Added Alpha quality Hello World and Micro Speech examples for TensorFlow Lite for Microcontrollers.

10.2 Improvements

None

10.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
663929	Fix a bug in PSA Crypto asymmetric key example.
682502	Relevant sample applications for EFR32FG22 now appear properly when selecting this device.
674528	Filtered out some sample applications that use too much RAM to be run on some kits.
674528	Filtered out MPU sample app on kits that do not have MPU.

Fixed in release 3.1.1.0

ID #	Description
664297	Fixed minor miscellaneous UI issues related to the version and component descriptions.

10.4 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.**

10.5 Deprecated Items

None

10.6 Removed Items

None

11 Boards and External Devices

11.1 New Items

Added in release 3.1.0.0

- Added support for new Thunderboard BG22 board (PG22-DK2503A).
- Added support for MGM220PC22HNA module as well as board SLWRB4311B.
- Added better support for 60 Ohm ESR crystals, as well as “sleepy” crystals, allowing to more reliably start crystal oscillation.
- Added support for WiFi boards BRD8022 and BRD8023 to be used as components.

11.2 Improvements

None

11.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
685556	Updated metadata of various boards, including frequency bands for BRD4206A.

Fixed in release 3.1.1.0

ID #	Description
661241	Fixed a bug that would prevent flash shutdown to trigger on Gecko SDK 3.1.0.

11.4 Known Issues in the Current Release

None

11.5 Deprecated Items

Deprecated in release 3.1.2.0

- Removed support for Thunderboard Sense SLTB001A, see End of Life notice #190827613 from 2019-08-27.

11.6 Removed Items

None

12 Other Gecko Platform Software Components

12.1 New Items

Added in release 3.1.0.0

- Added Alpha support for TensorFlow Lite for Microcontrollers version 2.3.1.

12.2 Improvements

None

12.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
673360	Fixed an issue allowing selection of pins that were not able to wake the CPU up from EM2 for the SL_WAKE_LOCK_INPUT in the wake lock component for Series 2 MCU. For Series 2 MCU, only pins from GPIO Port A and B, and pins labelled as EM4WU can wake the CPU up from EM2.

12.4 Known Issues in the Current Release

None

12.5 Deprecated Items

None

12.6 Removed Items

None

13 RAIL Library

13.1 New Items

Added in release 3.1.1.0

- Added support on the EFR32xG22 parts for a new BLE PHY that can receive 1Mbps and LR Coded packets simultaneously. See RAIL_BLE_ConfigPhySimulscan().

Added in release 3.1.0.0

- Added new API RAIL_GetRadioStateDetail() that provides more detailed radio state information than RAIL_GetRadioState.
- Added RAIL_RxPacketInfo_t::filterMask field of type RAIL_AddrFilterMask_t, which is a bitmask representing which address filter(s) the packet has passed.
- Added the ability for RAIL_GetRssi() to wait for a valid RSSI in radio states that are transitioning into RX. Additionally, a maximum wait timeout for a valid RSSI can be configured using the new API RAIL_GetRssiAlt().
- Added a new RAIL_ZWAVE_ConfigRxChannelHopping() API to configure Z-Wave Rx channel hopping using the recommended hopping parameters.
- Added a new RAIL_ZWAVE_GetRegion() API to determine the currently selected Z-Wave region.
- Added a new RAIL_SupportsTxPowerModeAlt() API to get the minimum and maximum power levels for a specific power mode if the power mode is supported by the chip.
- Added a new API RAIL_SetAddressFilterAddressMask() that allows for setting a bit mask pattern for packet data in the address filters.
- Added support for MGM210PB22JIA, MGM210PB32JIA, BGM210PB22JIA and BGM210PA32JIA modules.
- Added an event RAIL_EVENT_PA_PROTECTION to indicate the power protection circuit has kicked in.
- Created a “RAIL Utility, Callbacks” component for application-level callbacks.

13.2 Improvements

Changed in release 3.1.2.0

- If the selected RAIL_TxPower_t is not supported by the chip, then by default RAIL_TX_POWER_MODE_2P4_HIGHEST is chosen as the power mode.
- A multiprotocol scheduled receive with a relative start time and an absolute end time now respects that end time regardless of when the receive actually starts. Before, the end time was made relative and pushed out based on the actual start time to keep the receive window the same width in duration. This is a backwards-incompatible change, but should align better with what users expected when specifying an absolute end time.

Changed in release 3.1.0.0

- Added support for reporting more detailed transmit errors on the Packet Trace Interface (PTI).
- Updated RAIL_ZWAVE_ReceiveBeam() to automatically idle the radio when RAIL_ZWAVE_ReceiveBeam() finishes even when no beam is detected.
- Added the ability to use the “RAIL Utility, Initialization” component multiple times when creating a multiprotocol application.
- Changed RAIL_PacketTimeStamp_t::totalPacketBytes from uint32_t to uint16_t to reduce RAM usage.
- The “RAIL Utility, Initialization” component now defaults most options to a disabled state, instead of enabled. Now you have to opt-in, instead of opt-out, of RAIL init functionality.
- The “RAIL Utility, PA” component now enables PA calibration by default to ensure that PA power remains consistent chip-to-chip.
- Add new RAIL_EVENT_RF_SENSED as an alternative to the current RAIL_StartRfSense() callback parameter.
- Added a new API RAIL_ConfigSleepAlt() to allow configuring the PRS channel, RTCC channel, and whether sleep is enabled in one call.
- Created a new “RAIL Utility, Protocol” component for setting up RAIL to use one of the standards based PHYs by default.
- In multiprotocol RAIL, when the supplied handle is not the active handle, RAIL_GetRadioState now returns RAIL_RF_STATE_RX rather than RAIL_RF_STATE_IDLE if a background receive is currently scheduled.
- Antenna diversity settings for xGM210 modules are now split in a new config file: sl_rail_util_ant_div_config.h.

13.3 Fixed Issues

Fixed in release 3.1.2.0

ID #	Description
666275	Fixed potential delays when using RAIL's channel hopping or duty cycling features in EM1P mode on the EFR32xG22 and newer parts.
666917	Previously runtime-configuring the Zigbee coexistence PHY select timeout to 255ms would only guarantee 255ms on the coexistence-optimized PHY. Now the coexistence-optimized PHY will be selected indefinitely after runtime-configuring the coexistence PHY select timeout to 255ms.
667555	Fixed an issue with the “Radio Utility, FEM” component where it would generate an error if the Rx pin was not required and turned off by the user.
669697	Updated the configuration libraries on BGM210PB22JIA and BGM210PB32JIA modules to fix issues using these parts in the field.
671817	Fixed an issue when switching between certain radio configurations (e.g. Z-Wave) where use of RAIL_TX_OPTION_REMOVE_CRC can become permanently stuck.
672904	The minimum and maximum power levels for BGM220P and BGM220SC modules are updated to be 0 and 15 respectively.
672909	Fixed an issue with PA auto mode where it might try to use a RAIL_TxPowerMode_t that is not valid for the current part.
684407	Fixed an issue where calling RAIL_ConfigAntenna() on EFR32xG22 and newer devices could fault if the GPIO block clock was not explicitly enabled beforehand.
687455	Fixed the conversion of power level to dBm to fetch correct values for BGM220P and BGM220S modules when using the RAIL_TX_POWER_MODE_2P4GIG_HP power mode and power levels above the maximum.

Fixed in release 3.1.1.0

ID #	Description
652969	Restored automatic IR calibration on EFR32xG22 at RAIL_Init() time.
653955	Fixed an issue when using the “RAIL Utility, Coexistence” component where PWM was enabled if SL_RAIL_UTIL_COEX_PWM_DEFAULT_ENABLED was set, even if SL_RAIL_UTIL_COEX_PWM_REQ_ENABLED was disabled. Now SL_RAIL_UTIL_COEX_PWM_DEFAULT_ENABLED is ignored when SL_RAIL_UTIL_COEX_PWM_REQ_ENABLED is disabled.
654726	Fixed an issue where antenna diversity was not being enabled on EFR32xG1x devices when configured through the “RAIL Utility, Antenna Diversity Configuration” component.
656175	Fixed an issue where any RAIL_ChannelConfigEntry_t in a RAIL_ChannelConfig_t with a maximum power less than 0 dBm would be stuck at the maximum power no matter what power was requested.
663815	Fixed an issue when using the “RAIL Utility, Antenna Diversity Configuration” component where moving from TX antenna mode HAL_ANTENNA_MODE_ENABLE2 to HAL_ANTENNA_MODE_DIVERSITY would cause the antenna selection to get stuck on antenna 2.
665161	Fixed an issue on EFR32xG22 and later PTI where the network analyzer could misrepresent the RSSI of incoming packets.

Fixed in release 3.1.0.0

ID #	Description
362133	The default RSSI offset on the EFR32xG1, EFR32xG12, EFR32xG13, EFR32xG14, and EFR32xG21 chips does not compensate for a known internal hardware offset. This offset is chip specific and can be found using the new “RAIL Utility, RSSI” component which will load the correct value for your chip by default when the plugin is enabled. Since the hardware and antenna design can also impact this offset it is recommended that you measure this value for your particular hardware for the best accuracy. This correction is not enabled by default on the chips listed above

ID #	Description
	to prevent changing radio behavior significantly without the user opting into this change. For the EFR32xG22 and future chips the hardware offset is measured and included by default.
471715	Fixed an issue when using RAIL_ConfigAntenna() on the EFR32xG22 with an RF path other than 0 since these parts do not have multiple RF paths.
519195	The EFR32xG21 will now use RTCC channel 0, as opposed to the PRORTC, to perform sleep timer synchronization. This will help lower the EM2 current consumption for this chip.
630457	On custom boards, the “RAIL Utility, PTI“ component no longer reserves pins for use without being configured.
632723	The EFR32xG22 will limit going to EM1P sleep mode when an 80MHz HRFRCO PLL system clock is selected. Going to EM1P sleep is not supported when using the DPLL on this hardware as it can cause clock drift which would impact radio timing and tuning.
638067	Fixed a DMP issue that poached transmit power when switching between protocols using the same channel configuration and channel.
639833	Fixed a potential radio hang on a corrupted BLE packet when doing BLE AoX.
642893	Reduced RAIL library flash data alignment needs on the EFR32xG22.
645641	Fixed an EFR32xG22 issue where a state transition to receive after a transmit from EM2 sleep would drop packets.

13.4 Known Issues in the Current Release

None

13.5 Deprecated Items

None

13.6 Removed Items

None

14 Legal

14.1 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**

14.2 Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, ClockBuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, “the world’s most energy friendly microcontrollers”, Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.