Gecko Platform 3.2.3.0 GA
Gecko SDK Suite 3.2
October 13, 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 3.2.3.0 released October 13, 2021
Gecko Platform 3.2.2.0 released September 8, 2021
Gecko Platform 3.2.1.0 released July 21, 2021
Gecko Platform 3.2.0.0 released June 16, 2021

KEY FEATURES

CMSIS Device
- Added support for EFR32xG23 devices

Peripherals
- EFP Voltage scaling and Direct mode on EFR32xG22 and EFM32xG22.
- EFP Coulomb counter

Services
- Several fixes related to Power Manager

Middleware
- CSLIB can now be used with the Gecko SDK components

Security
- Mbed TLS library updated to v2.26.0

Operating System
- Added more NOR Flash options for Micrium OS FS

Examples
- Coulomb Counter
- CLI & I2C

Boards and External Devices
- Added support for several new boards

Other Platform Components
- Updated IAR compiler to version 8.50.9.
- Updated GCC compiler to version 10.2.
- Amazon FreeRTOS Libraries have been added to the Gecko SDK.

Contents

1 CMSIS Device

1.1 New Items

<u>Added in release 3.2.2.0</u>

- Added support for the following new OPNs:
  - EFR32xG23 Family

<u>Added in release 3.2.1.0</u>

- Added support for the following new OPNs:
  - EFM32PG23B Family

<u>Added in release 3.2.0.0</u>

- Added support for the following new OPNs:
  - EFR32ZG13P531F512GM48

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

Added in release 3.2.1.0

- Added the `EM_MSC_RUN_FROM_RAM` configuration. It can be used to enable running flash write functions from RAM.
- OUTPAD control is now supported by calling `IDAC_OutpadEnable()` on selected Series 1 chips.

2.2   Improvements

Changed in release 3.2.0.0

- Improved the disable and reset sequence for the VDAC peripheral, to make sure the disabling is properly reported in the peripheral's registers, under all clocking conditions.

2.3   Fixed Issues

Fixed in release 3.2.2.0

ID #	Description
721468	EUSART instances on PD1 power domain need special consideration when transitioning to/from EM23. This fix monitors Power Manager transition events and prepares the EUSART accordingly (this affects SPIDRV and memlcd drivers).
722634	Fixed a problem with `LESENSE_ScanFreqSet()` not calculating the right parameters for some Series 2 devices.
724839	Enabled FPU in Chip_Init by default on M33 devices to lower current consumption as prescribed by the hardware errata workaround.

Fixed in release 3.2.0.0

ID #	Description
698009	Fixed `CMU_DPLLLock()` hard faults by making sure the DPLL unit clock is enabled, under every condition.
580315	Fixed an issue when using OPA0 as VDAC channel 0 buffer on EFM32MG11.

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

Added in release 3.2.0.0

- Introduced a [Coulomb Counter driver](#) to measure coulombs flowing through EFP's outputs.
- Added support for [EFP Voltage scaling and Direct mode](#) on EFR32xG22 and EFM32xG22.

3.2 Improvements

Changed in release 3.2.2.0

- The CS pin is now optional for SPIDRV, correctly allowing the CS to be controlled by the application.

Changed in release 3.2.0.0

- On Series 2 chips (except EFR32xG21), voltage scaling in EM0 is now supported when EFP provides DECOUPLE.
- The initialization function of the gpiointerrupt driver can now be called multiple times without any problem.
- gpiointerrupt driver, which used to only support EXTI pins, now supports EM4WU pins.

3.3 Fixed Issues

Fixed in release 3.2.0.0

ID #	Description
688457	Fixed UARTDRV LEUART port location in autogenerated code.
661098	Removed requirement on energy level from UARTDRV when aborting transfers.
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.
674173	Fixed an issue in SPIDRV preventing the creation of a USART or EUSART SPIDRV instance in Simplicity Studio.
646417	Added EUSART signal support in emdrv's DMADRV.
693334	Disabled RGBW PWM LED component for Series 2 devices because it is not supported on these parts.

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

None

4 Services

4.1 New Items

Added in release 3.2.0.0

- In IO Stream, added compiler option SL_IOSTREAM_USART_FLUSH_TX_BUFFER to flush USART buffer before returning from write.

4.2 Improvements

Changed in release 3.2.0.0

- The max interrupt disable counter now pauses when the device is in sleep, for projects that use the Power Manager.
- Improved accuracy of Sleep Timer where timers could expire one tick later than requested.
- The CLI now allows a command named 'group'.
- The CLI now allows the same subgroup name in multiple groups.
- Added GPIO_Interrupt initialization call to generated driver_init() function.
- Resolved a few MISRA rules violations in various Service components.
- Improved the documentation of the CLI by adding some missing information on argument separator and fixed a few typos.

4.3 Fixed Issues

Fixed in release 3.2.3.0

ID #	Description
687640	Fixed issue in Sleeptimer module where the timers' delta list could get desynchronized over time when timers are created/stopped during the Sleeptimer interrupt.

Fixed in release 3.2.2.0

ID #	Description
730915	IO Stream UART component don't show up for incompatible series 1 parts anymore.

Fixed in release 3.2.0.0

ID #	Description
656786	Fixed a bug in Sleep Timer where a variable was not declared using the volatile type qualifier and could lead to minor inaccuracies in timer timeouts when compiler optimization was enabled.
653374	The help is now displayed correctly for CLI commands added by application.
662722	EMU_EM23PresleepHook is now defined only once (in Power Manager), even when adding both EFP and Power Manager components to the project.
688784	Fixed an issue in the Power Manager related to creating a Sleep Timer requiring HF clock restored at timeout, but with a timeout smaller than the restore time. In some conditions, the HF clock restore was not done.
703231	Fixed an issue in Power Manager on Series 2 devices where we could get stuck waiting for HFXO to be ready when a hardware request on HFXO was removed in the middle of the restore process in Power Manager.
688482	Fixed a corner case issue in Power Manager that could lead to not restoring the HF clock when needed.
675253	Fixed an issue in Power Manager where, if the HFXO startup failed while actively waiting for it to be ready in a critical section, we could wait indefinitely.
677395	Fixed IO Stream-USART from signaling the semaphore before the kernel is started.
670648	Removed incorrect call to clear RXDATAV interrupt in IO Stream.
688275	Added option to configure LFXO timeout through Simplicity Studio.
695912	Fixed the symbols used to determine if the EM01GRPACLKC and EM01GRPBCLK are available, in device init.

ID #	Description
697483	Fixed bug affecting CTUNE for HFXO on Series 2 devices.
638429	Fixed a broken link to Power Manager component documentation.
627936	Fixed some formatting and typos in the CLI documentation.

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

None

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

Added in release 3.2.0.0

- CSLIB components are now released with the Gecko SDK.
- CSEN framework uses DMADRV rather than LDMA to avoid conflicts with other components.

6.2 Fixed Issues

Fixed in release 3.2.3.0

ID #	Description
734359	Fixed a potential issue in Micrium OS Net, where we now use a random value for the DNS Query ID number.
734360	Fixed issue in Micrium OS Net where the message length is used to validate when parsing a DNS response.

6.3 Known Issues in the Current Release

None

6.4 Deprecated Items

None

6.5 Removed Items

None

7   Security

7.1   New Items

Added in release 3.2.2.0

• Support x448 in PSA Crypto ECDH driver

Added in release 3.2.1.0

• Added support for hardware-accelerated Mbed TLS CCM-star

• Added CCM hardware acceleration for Mbed TLS on Series 1 devices

Added in release 3.2.0.0

• Updated the version of Mbed TLS to 2.26 with additional patches related to PSA Crypto support. The Mbed TLS base commit (https://github.com/ARMmbed/mbedtls) is 54650b389250728795005f928fab493ef011904d

• The TLS library is set up to use PSA Crypto APIs when the PSA Crypto UC component is included. The TLS library in this configuration is in Beta quality for this release.

• Added OS abstraction for the CMSIS RTOS2 API, which replaces the FreeRTOS and MicriumOS abstractions in SE Manager.

• 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.

• There is a new entropy source 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 PSA Crypto components for PSA cipher, mac, aead, hash APIs and PSA ECC curves.

• 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 more fine-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 in the near future releases.

• Added support for hardware-accelerated HMAC through PSA Crypto.

7.2   Improvements

Changed in release 3.2.0.0

• Changed the storage format for persistent keys stored through PSA Crypto, while retaining backwards compatibility for reading the old storage format. If you haven't stored any keys through PSA Crypto yet, you could save some code size by removing the storage format backwards compatibility support through setting the define SL_ITS_REMOVE_V1_HEADER_SUPPORT.

• Allow ITS files of size limited by NVM3_DEFAULT_MAX_OBJECT_SIZE in function psa_its_set(). In previous versions the file size was limited to 256 bytes.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 that can be accelerated by hardware. Previously, including MBEDTLS_ECP_NIST_OPTIM unnecessarily would cause an approximate 1.5 kB of code size overhead.

• 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.

• In PSA Crypto API function psa_key_derivation_setup(), return PSA_ERROR_NOT_SUPPORTED from when no KDF is enabled.

• Reduced entropy stack usage by providing MBEDTLS_CTR_DRBG_MAX_SEED_INPUT configuration.

• Other code size optimizations.

7.3   Fixed Issues

Fixed in release 3.2.1.0

ID #	Description
716363	Fixed issue with signing messages of non-word-multiple length with Ed25519 on Vault devices having hardware support for Ed25519.
714412	Under rare circumstances, the TRNG driver on certain devices could throw a spurious error. This would cause an error code to be bubbled up to the application, so there are no security implications. However, to avoid application misbehavior in case return codes are not properly checked, we have added some retry logic to mitigate the spurious errors as much as possible.

Fixed in release 3.2.0.0

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.
667205	In the psa_import_key() function, allow import of un-clamped Montgomery keys.
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).
479967	The APIs implemented by em_aes.c now correctly account for potentially unaligned buffers.

7.4 Known Issues in the Current Release

None

7.5 Deprecated Items

None

7.6 Removed Items

Removed in release 3.2.0.0

- Software fallback algorithms that were introduced as a workaround for a bug in EFR32xG21 SE Firmware versions before 1.2.2 are now removed by default. See the accompanying security advisory. This means ECDH operations and public key validity checks will by default return an error code on EFR32xG21 parts with SE firmware before 1.2.2. 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 3.2.0.0

- Added `configPRINT_STRING` to configuration file to enable Amazon FreeRTOS.
- Integrated File System driver for NOR Flash using SPIDRV.
    - Added macro to register NOR FS SPIDRV driver with Platform Manager.

8.2   Improvements

Changed in release 3.2.0.0

- Updated the version of FreeRTOS from 10.3.0 to 10.4.3 in order to fix a bug on Cortex-M33 device where the interrupts could be disabled when entering the first scheduled task.

8.3   Fixed Issues

Fixed in release 3.2.3.0

ID #	Description
730413	Fixed a wrap-around issue in Micrium OS Kernel's OSTimeGet().
738384	Fixed a bug in the Micrium OS Kernel where, in some rare instances, when doing floating point operation from the idle context, an ISR stack corruption could occur at a later time when entering an exception/interrupt.

Fixed in release 3.2.0.0

ID #	Description
660929	CMSIS-RTOS2: osMutexAcquire will no longer ignore the osMutexRecursive attribute.
661498	In Micrium OS, calling a blocking pend on an OS object with interrupts disabled now returns an error.
704971	Fixed an issue in FreeRTOS's Sleeptimer/Power Manager port where Interrupts would be wrongly re-enabled.
687142	Modified the tick_powermanager.c FreeRTOS port file to use a standard macro provided by portmacro.h to call a yield instead of using Cortex-M specific macros that may not always be available.
687939	Fixed missing component dependency for Micrium OS Shell.

8.4   Known Issues in the Current Release

None

8.5   Deprecated Items

None

8.6   Removed Items

Removed in release 3.2.0.0

- Removed older FS NOR SPI driver.

9   Gecko Bootloader

9.1   New Items

None

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 3.2.0.0

- Added an example application to demonstrate the Coulomb Counter. It uses Coulomb Counter driver API to read coulomb and NVM3 to store value in memory. The example also uses a CLI commands interface to read coulomb counter values.
- Added example applications for I2C with Micrium OS Kernel, FreeRTOS and no RTOS. These examples make use of the I2CSPM API's.
- Added example applications for the CLI with Micrium OS Kernel, FreeRTOS and no RTOS. These examples make use of the CLI API's.
- Added empty C++ project to platform examples.

10.2 Improvements

None

10.3 Fixed Issues

Fixed in release 3.2.0.0

ID #	Description
663929	Fix a bug in PSA Crypto asymmetric key example.
682804	UARTDRV example is now running correctly on BRD4206A. Adding support for LFRCO oscillator was required, because BRD4206A does not have an LFXO oscillator.
711560	Fixed an issue with the metadata of some NVM3 examples.
703945	BRD4180 (A and B) have been removed from the list of the compatible boards for the SPIDRV sample apps because of a pin allocation conflict.
692903	Fixed the list of boards that are compatible with the dmadrv_baremetal example.

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.si-labs.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 3.2.2.0

- Added support for several new boards with an EFR32xG23 (including BRD2600A, BRD4204D, BRD4210A).

Added in release 3.2.1.0

- Added support for the following new OPNs:
  - BRD2504A

Added in release 3.2.0.0

- Added support for the following new OPNs:
  - BG22-EK4108A/BRD4108A
  - BRD4183C
  - BRD4321A
  - WGM160P
- Added new display driver for the LPM013M126A in memlcd driver.
- Added support for CoreHW 12x12 PCB8 for our Bluetooth AoX Solutions.

11.2 Improvements

Changed in release 3.2.0.0

- Clarified that calling `sl_board_enable_sensor()` could have some side-effects on some boards (BRD4166A, BRD4184A, and BRD4184B).

11.3 Fixed Issues

Fixed in release 3.2.2.0

ID #	Description
706851	Fixed flow control configuration for VCOM on BRD4166A.
727759	Fixed wrong board override configuration for the iot_spi component (corrected pin definition for SPI signals).

Fixed in release 3.2.0.0

ID #	Description
685556	Metadata about radio bands have been correctly added to BRD4206A board.
695875	Fixed PA value for MGM12P32F1024GA.
660844	Fixed CS and CLK route settings for I2S microphone for Series 2 devices boards.

Fixed in release 3.2.1.0

ID #	Description
720809	Fixed BRD4321_A04 definition in the GSDK to avoid conflicting IC selection with metadata already defined in Studio

11.4 Known Issues in the Current Release

None

11.5 Deprecated Items

Deprecated in release 3.2.0.0

- Functions to configure EFP regulators peak current have been deprecated: `sl_efp_set_voa_em01_peak_current`, `sl_efp_set_voa_em23_peak_current`, `sl_efp_set_vob_em01_peak_current`, `sl_efp_set_vob_em23_peak_current`.

11.6 Removed Items

None

12 Other Gecko Platform Software Components

12.1   New Items

Added in release 3.2.0.0

- Updated IAR compiler to version 8.50.9.
- Updated gcc compiler to version 10.2.
- Amazon FreeRTOS Libraries have been added to the platform SDK. They are provided with the Gecko SDK for running aws_test and validating the low-level implementation of IoT API. This module is not used internally by any Silicon Labs component. Be aware of the following security vulnerabilities:
    - CVE-2019-13120 : "Amazon FreeRTOS up to and including v1.4.8 lacks length checking in prvProcessReceivedPublish, resulting in untargetable leakage of arbitrary memory contents on a device to an attacker. If an attacker has the authorization to send a malformed MQTT publish packet to an Amazon IoT Thing, which interacts with an associated vulnerable MQTT message in the application, specific circumstances could trigger this vulnerability."
    - CVE-2018-16524 : "Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component allow information disclosure during parsing of TCP options in prvCheckOptions."
    - CVE-2018-16525 : "Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component allow remote attackers to execute arbitrary code or leak information because of a Buffer Overflow during parsing of DNS\LLMNR packets in prvParseDNSReply."
    - CVE-2018-16526 : "Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component allow remote attackers to leak information or execute arbitrary code because of a Buffer Overflow during generation of a protocol checksum in usGenerateProtocolChecksum and prvProcessIPPacket."
    - CVE-2018-16527 : "Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component allow information disclosure during parsing of ICMP packets in prvProcessICMPPacket."
    - CVE-2018-16598 - "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. In xProcessReceivedUDPPacket and prvParseDNSReply, any received DNS response is accepted, without confirming it matches a sent DNS request."
    - CVE-2018-16599 : "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. Out of bounds memory access during parsing of NBNS packets in prvTreatNBNS can be used for information disclosure."
    - CVE-2018-16600 : "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. Out of bounds memory access during parsing of ARP packets in eARPProcessPacket can be used for information disclosure."
    - CVE-2018-16601 : "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. A crafted IP header triggers a full memory space copy in prvProcessIPPacket, leading to denial of service and possibly remote code execution."
    - CVE-2018-16602 : "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. Out of bounds memory access during parsing of DHCP responses in prvProcessDHCPReplies can be used for information disclosure."
    - CVE-2018-16603 : "An issue was discovered in Amazon Web Services (AWS) FreeRTOS through 1.3.1, FreeRTOS up to V10.0.1 (with FreeRTOS+TCP), and WITTENSTEIN WHIS Connect middleware TCP/IP component. Out of bounds access to TCP source and destination port fields in xProcessReceivedTCPPacket can leak data back to an attacker."
    - CVE-2019-13120 : "Amazon FreeRTOS up to and including v1.4.8 lacks length checking in prvProcessReceivedPublish, resulting in untargetable leakage of arbitrary memory contents on a device to an attacker. If an attacker has the authorization to send a malformed MQTT publish packet to an Amazon IoT Thing, which interacts with an associated vulnerable MQTT message in the application, specific circumstances could trigger this vulnerability."
- Added Unity Test framework to Gecko SDK delivery to enable Amazon FreeRTOS.
- Added bare-metal (no RTOS) support to SystemView.
- Added FreeRTOS support to SystemView.

12.2   Improvements

None

12.3   Fixed Issues

Fixed in release 3.2.0.0

ID #	Description
690494	Added Linker section in GCC linker script for Segger RTT in order to support the block/buffer auto-detection.
653931	Fixed tiny-printf library compilation issue with GCC9.
668857	Fixed issue when creating a project for Series 0/1 device on IAR 8.5 and a corrupt file error would happen.

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 3.2.2.0

- Added support for the EFR32xG23 series of chips.
- Added a new RAIL_ConfigDirectMode() API for configuring direct mode settings on all chips. This can set whether the data stream is synchronous or asynchronous and which GPIOs are used for the feature.
- Added PA curves for EFR32XG23 for HP, MP, LP and LLP modes for both 14dBm and 20dBm variants.
- Added a new RAIL_RxDataSource_t to capture direct mode data on supported devices.

Added in release 3.2.1.0

- Added the new RAIL_STREAM_CARRIER_WAVE_PHASENOISE RAIL_StreamMode_t for phase noise measurement.
- Added RAIL_PA_BAND_COUNT to count RAIL_PaBand_t.

Added in release 3.2.0.0

- Added RAIL_StartScheduledCcaCsmaTx and RAIL_StartScheduledCcaLbtTx APIs to allow applications to schedule a CSMA/LBT transmit.
- Added support for a new RAIL_EVENT_ZWAVE_LR_ACK_REQUEST_COMMAND, triggered on the reception of a Z-Wave Long range packet with acknowledgement request bit set, following which the application must call RAIL_ZWAVE_SetLrAckData to populate the fields of the Z-Wave Long range acknowledgement packet.
- Added RAIL_TX_OPTION_RESEND to allow re-transmitting the packet most recently loaded into the Transmit FIFO. This can be used in combination with RAIL_SetNextTxRepeat() on supported platforms to repeatedly transmit the same packet.
- Added new RAIL_SetNextTxRepeat() API and RAIL_TxRepeatConfig_t to allow configuration of repeated transmits triggered by an initial transmit.
- Added ability to configure TxToTx state transition time via RAIL_StateTiming_t::txToTx and RAIL_SetStateTiming(). This time is generally used between an autoACK transmit and a user transmit that was pending. It is also used during repeated transmits by default, but can be overridden by the RAIL_TxRepeatConfig_t::delayOrHop or RAIL_BLE_TxRepeatConfig_t::delayOrHop configuration.
- Added ability to hop channels during repeated transmits in RAIL_TxRepeatConfig_t using RAIL_TX_REPEAT_OPTION_HOP.
- Added BLE-specific hooks for repeated transmits with channel hopping in RAIL_BLE_SetNextTxRepeat() and RAIL_BLE_TxRepeatConfig_t.
- The "RAIL Utility, Callbacks" component can now be configured to not provide the RAILCb_AssertFailed() function for situations where the application wants to provide its own implementation.

13.2 Improvements

Changed in release 3.2.2.0

- Now when using a RAIL_RxDataSource_t other than RX_PACKET_DATA the receiver will be disabled and must be manually restarted any time a RAIL_EVENT_RX_FIFO_OVERFLOW occurs. If the RAIL_EVENT_RX_FIFO_OVERFLOW event is not enabled, then the receiver will continue to run and lose some amount of data until the buffer is processed that matches the old behavior.

Changed in release 3.2.0.0

- Added support for the RAIL_TX_POWER_MODE_*_HIGHEST options in the RAIL_GetTxPowerCurve() function.
- Changed when RAIL_EVENT_IEEE802154_DATA_REQUEST_COMMAND event is issued to better facilitate support for 802.15.4E-2012 Enhanced ACKing and reduce time spent in the event handler. The event is now issued *after* receiving the Auxiliary Security Header (if present) in the MAC Header of the incoming frame, and for MAC Command frames, after receiving the MAC Command byte (which may be encrypted). This change is *not* backwards-compatible with prior releases for Enhanced ACK requesting frames, but is for Immediate ACK requesting frames. Use of RAIL_IEEE802154_EnableEarlyFramePending() is no longer required to support Enhanced ACKing; the notion of early frame pending notification has also shifted to after the Auxiliary Security Header for Enhanced ACK requesting frames.
- Changed RAIL Timer Synchronization over sleep on the EFR32xG21 to use the RTCC instead of the PRORTC to reduce current consumption in EM2.

- The posting of RAIL_EVENT_TX_ABORTED now occurs before the PA ramps down, at the same time other transmit completion events get posted.
- Allow RAIL_ZWAVE_ReceiveBeam to be run on US LR regions instead of always returning an error.

13.3 Fixed Issues

Fixed in release 3.2.3.0

ID #	Description
737292	Fixed an issue on the EFR32xG21 that was causing some Bluetooth LE coded PHY packets to be sent with a corrupted sync word and payload, breaking the connection.
738158	Fixed the RX_DIRECT_MODE_DATA RAIL_RxDataSource_t data source on the EFR32xG23.
741120	Fixed an issue where the IQ data captured during Bluetooth LE's CTE could be all zeros on the 2Mbps PHY.
743258	Fixed an issue that could cause the antenna switching pattern in Bluetooth LE AoX applications to sometimes repeat a previous sequence instead of following the expected antenna sequence.

Fixed in release 3.2.2.0

ID #	Description
671651	Fixed timing problems with certain State_Transitions or Rx_Channel_Hopping delay values on the EFR32xG22 and newer parts.
714271	Fixed an issue where RAIL_IEEE802154_Config2p4GHzRadio*() and RAIL_IEEE802154_ConfigGB*Radio() functions were improperly clearing or were setting certain RAIL_IEEE802154_EOptions_t. Also documented that these functions still implicitly clear or set certain RAIL_IEEE802154_GOptions_t suitable for that configuration.
716369	Fixed an issue where incorrect radio transition times were being applied at higher temperatures when using the high power PA on EFR32xG22 parts.

Fixed in release 3.2.1.0

ID #	Description
646980	An attempt to use an unsupported built-in radio channel configuration, e.g. on a module that does not support that protocol or configuration, will now trip RAIL_ASSERT_FAILED_INVALID_CHANNEL_CONFIG rather than returning success and ignoring the configuration.
675252	Fixed an antenna diversity regression where a transmitted auto-ACK would incorrectly go out the currently configured TX antenna rather than the antenna used to receive the packet being acknowledged.
676896	Fixed an issue in OOK PHYs where dynamic adjustments made to receive a packet with a high RSSI could persist after the packet and decrease the ability to receive packets with a lower RSSI.
696198	Fixed an issue on PHYs that do not support dual sync words. If RAIL_RX_OPTION_ENABLE_DUALSYNC is called when one of these PHYs is loaded the RAIL_ConfigRxOptions function will now return RAIL_STATUS_INVALID_PARAMETER.
697097	Fixed a rare situation where a premature RAIL_EVENT_TX_BLOCKED event might occur when auto-ACK is enabled and a scheduled transmit using RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX is pending when an erroneous packet is received.
700439	Fixed an issue where configuring a Selective RF Sense Wakeup packet by calling RAIL_ConfigRfSenseSelectiveOokWakeupPhy followed by RAIL_SetRfSenseSelectiveOokWakeupPayload would put RAIL in fixed length mode and leave it there even after a new PHY was loaded. We will now revert the fixed length settings to their default when loading a new PHY after the Selective RF Sense Wakeup PHY was loaded so that the incoming PHY's settings are used.
710273	Fixed an issue using RAIL_TX_REPEAT_OPTION_HOP with RAIL_SetNextTxRepeat() or RAIL_BLE_SetNextTxRepeat() when they return an error found in the channel-hopping configuration, yet would still attempt to repeat the next transmit.

Fixed in release 3.2.0.0

ID #	Description
456701	Fixed an issue on EFR32xG1x parts where calling RAIL_Init() with the MSC->CTRL.CLKDISFAULTEN bit set would cause a bus fault.
654600	On EFR32XG21, a watchdog has been added to terminate an RSSI averaging operation in case the RAIL_EVENT_RSSI_AVERAGE_DONE event does not occur.
655541	Fixed an issue on EFR32XG22 and later where packet filtering might be incomplete in FEC-enabled radio configurations causing good packets to be improperly dropped. Note that if packet filtering fails close to the end of an otherwise successfully received packet, the packet may be flagged RAIL_RX_PACKET_READY_CRC_ERROR rather than RAIL_RX_PACKET_ABORT_FILTERED. Note this issue is still present on earlier chips.
665705	Fixed an issue where a transmit with RAIL_TX_OPTION_SYNC_WORD_ID 1 to use SYNC2 would improperly indicate SYNC1 was used in the packet trace appended information.
666275	Fixed potential delays when using RAIL's channel hopping or duty cycling features in EM1P mode on the EFR32xG22 and newer parts.
667103	Fixed RAIL_ReadRxFifo() to behave as documented when passed NULL dataPtr: the data is thrown away rather than copied out.
671817	Fixed an issue when switching between certain radio configurations (for example, Z-Wave) where use of RAIL_TX_OPTION_REMOVE_CRC can become permanently stuck.
673333	Fixed an issue with RAIL_TX_OPTION_WAIT_FOR_ACK transmits where an RX overflow during the ACK wait period would silently abort the ACK timer resulting in no RAIL_EVENT_RX_ACK_TIMEOUT being generated.
682032	Fixed an issue where setting RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX for a scheduled transmit could cause subsequent non-scheduled transmits to be blocked.
696665	Fixed an issue in RAIL_ConvertRawToDbm for PAs that use piecewise-linear line segment fit, where the minimum raw power level was incorrectly compared to the minimum deci-dBm value.
697097	Fixed a rare situation where a premature RAIL_EVENT_TX_BLOCKED event might occur when auto-ACK is enabled and a scheduled transmit using RAIL_SCHEDULED_TX_DURING_RX_ABORT_TX is pending when an erroneous packet is received.
699890	Fixed missing C++ compatibility in the "RAIL Utility, Init" component's generated header file.
701604	Fixed an issue where using Rx_Channel_Hopping with channels in different frequency bands would cause RAIL to assert.
703788	Fixed an issue on EFR32xG2x devices where RAIL would not allow you to initialize the radio with voltage scaling enabled even though this is supported on these devices.
705595	Fixed an issue where custom PA curves provided through the PA Module were not respected when building Silicon Labs Zigbee applications.
708511	Fixed possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_TX_FIFO when RAIL_SetTxFifo() is called after the transmit FIFO had been filled enough to cause its write offset to wrap.

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.si-labs.com/products/software.

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.	

13.5 Deprecated Items

None

13.6 Removed Items

None

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

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

www.silabs.com