Gecko Platform 2.7.9.0 GA
Gecko SDK Suite 2.7
March 3, 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 and EMLIB are low-level core and peripheral support libraries. EMLIB provides a complete peripheral API for all Silicon Labs EFM32, EZR32 and EFR32 MCUs and SoCs.

EMDRV 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.

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.

NVM3 (Non-Volatile Memory Version 3) is a data storage driver that provides a means to read and write data objects (key/value pairs) stored in flash memory. NVM3 can be used with the Bluetooth, Zigbee, Thread, and Connect protocol stacks.

mbed TLS provides an SSL library that makes it easy to use cryptography and SSL/TLS in your applications. mbed TLS is open source software licensed by ARM Limited.

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.

This document aggregates information that in previous releases was spread across multiple documents. In earlier versions of the Gecko SDK, this content would have been found in: 32-bit MCU SDK Release Notes, RAIL Library Release Notes, and the Gecko Bootloader change log.

These release notes cover SDK version(s):

Gecko Platform 2.7.9.0 GA released March 3, 2021
Gecko Platform 2.7.8.0 GA released October 28, 2020
Gecko Platform 2.7.7.0 GA released August 26, 2020
Gecko Platform 2.7.6.0 GA released May 27, 2020
Gecko Platform 2.7.5.0 GA released May 1, 2020
Gecko Platform 2.7.4.0 GA released April 22, 2020
Gecko Platform 2.7.3.0 GA released March 20, 2020
Gecko Platform 2.7.2.0 GA released February 21, 2020
Gecko Platform 2.7.1.0 GA released January 24, 2020
Gecko Platform 2.7.0.0 GA released December 13, 2019

KEY FEATURES

CMSIS
- Bug fixes

EMLIB
- Bug fixes and deprecations

EMDRV
- Bug fixes and deprecations

RAIL Library
- Added support for a mode to select the best PA for a given power level
- Added support for new IEEE 802.15.4G-2012 features

NVM3
- Fixed data corruption bug
- Improved worst case repack execution time

mbed TLS
- Added AES-GCM support for Series-2
- Updated mbed TLS library to version 2.7.12

Gecko Bootloader
- Added certificate support
- Added EZSP GPIO plugin
- Added EUART driver

Other Gecko Platform Software

Contents

CMSIS

1.1 New Items

Added in release 2.7.2.0

- Added CMSIS device files for BGM22 and MGM22 families

1.2 Improvements

None

1.3 Fixed Issues

Fixed in release 2.7.4.0

ID #	Description
471098	Value of LFRCO_PRECISION_MODE changed from false/true to 0/1.

Fixed in release 2.7.0.0

ID #	Description
334234	SystemHFClockGet() in CMSIS system file now accounts for HFRCODIV2 (Series-1 only).
426081	Fixed handling of USHFRCO frequency in SystemHFClockGet(). CMSIS system now keeps track of the frequency and EMLIB CMU works in concert with CMSIS system. This applies to EFM32HG, EFM32GG11 and EFM32GG12 families.|

1.4 Known Issues in the Current Release

None

1.5 Deprecated Items

None

1.6 Removed Items

None

EMLIB

2.1 New Items

Added in release 2.7.4.0

- An errata EMU_E220 and subsequent product change notification (PCN) will be published by May 2020. This covers a problem where systems operating with core voltage scaling can infrequently experience a decouple voltage brown out (DECBOD) reset when waking from EM2 or EM3. A workaround is included in this release. However, the workaround increases the wakeup time by 2.7 μs only when waking up from EM2->EM0 or EM3->EM0 while using voltage scaling. If this is unacceptable, the customer can use the macro ERRATA_FIX_EMU_E220_DECBOD_IGNORE to remove the errata fix, and the code will behave like it used to.

Added in release 2.7.3.0

- Added MSC_WriteWordDma() function in em_msc. With this function the application is able to write to the flash memory using the DMA on series 1 and series 2 devices. This function should be used in order to achieve optimal flash write speeds.
- Added functions to read EMU internal temperature sensor on Series-1 and 2 products.
- Added support for new IADC result alignment options and IADC digital averaging on device families with hardware support for this.
- Added defines for more SE error codes to em_se.h.

Added in release 2.7.2.0

- A version check has been added to guard against use of Secure Boot with Anti-rollback on products with (V)SE version lower than v1.2.1.
- Added function to check if any (W)TIMER instance supports Dead Time Insertion (DTI) and added configuration support for Dead Time Insertion (DTI) on other timers than TIMER0.

Added in release 2.7.0.0

- MSC_MassErase() function is added for Series-2 devices.
- Add remote frame support in EMLIB CAN.

2.2 Improvements

Fixed in release 2.7.2.0

Improved HFLE clock branch handling for Series-1.

2.3 Fixed Issues

Fixed in release 2.7.7.0

ID #	Description
500265	Added checks and recovery mechanisms when attempting to set the frequency of HFRCODPLL to an unsupported value.

Fixed in release 2.7.4.0

ID #	Description
471165	An errata EMU_E220 and subsequent product change notification (PCN) will be published by May 2020. This covers a problem where systems operating with core voltage scaling can infrequently experience a decouple voltage brown out (DECBOD) reset when waking from EM2 or EM3. A workaround is included in this release. However, the workaround increases the wakeup time by 2.7 μs only when waking up from EM2->EM0 or EM3->EM0 while using voltage scaling. If this is unacceptable, the customer can use the macro ERRATA_FIX_EMU_E220_DECBOD_IGNORE to remove the errata fix, and the code will behave like it used to.
479597	Added missing enable of re-calibration in errata fix EMU_E220
464458	CMU_LFRCOSetPrecision must check PLFRCO_DEFEAT before setting HIGHPRECEN. Precision mode is defeatured on some Lynx OPNs.

Fixed in release 2.7.3.0

ID #	Description
465887	Fixed issue where calling LETIMER_Reset() could cause BusFault when called on a disabled LETIMER peripheral.
467724	The MSC clock in CMU->CLKEN1 is now enabled by MSC write functions for devices where this clock can be enabled/disabled.

Fixed in release 2.7.2.0

ID #	Description
464465	EMLIB version defines _EMLIB_VERSION_x added back to em_version.h with deprecation notice. Note that the version represented by the defines is actually the 32-bit MCU SDK version. The version of EMLIB follows Gecko Platform. A new version for EMLIB and Gecko Platform will be introduced in a later release.
456751	Fixed issue where GPIO_EM4EnablePinWakeup() sometimes did not clear interrupts, causing an immediate wakeup from EM4.

Fixed in release 2.7.0.0

ID #	Description
415819	CHIP_Init() sets HFRCOEM23 clock as TRACECLK.
447797	EMLIB IADC: The definition of `iadcNegInputGnd` has been modified to set PINNEG to 1. This prevents a polarity error when performing IADC conversions between supply pins and ground.
370421	Fixed conversion of raw data in IADC_ConvertRawDataToResult().
428960	Fixed issue that could cause dcdcEm01LoadCurrent_mA, a parameter of EMU_DCDCOptimizeSlice(),to be used before value assignment.

2.4   Known Issues in the Current Release

None

2.5   Deprecated Items

Notified in version 2.7.0.0

Functions in em_msc are placed in flash for Series-0 and Series-1 devices, except for the EFM32G. MSC_WriteWordFast() function is deprecated. Calling the MSC_WriteWordFast() function will have the same effect as calling MSC_WriteWord().

2.6   Removed Items

Removed in version 2.7.4.0

The CMU_LFRCOSetPrecision() function is now only available on devices that actually have the precision LFRCO feature.

Platform Driver/EMDRV

3.1   New Items

Added in release 2.7.0.0

SPIDRV is now using Sleeptimer instead of RTCDRV for time keeping in slave mode.

3.2   Improvements

None

3.3   Fixed Issues

Fixed in release 2.7.3.0

ID #	Description
465334	Fixed an issue where TEMPDRV_GetTemp() could return incorrect temperature results if not called immediately following a temperature interrupt.

3.4   Known Issues in the Current Release

None

3.5   Deprecated Items

Notified in version 2.7.0.0

RTCDRV driver is marked as deprecated and will be removed in a later release.

3.6   Removed Items

None

RAIL Library

4.1 New Items

Added in release 2.7.7.0

- Added support for new BGM220 modules.

Added in release 2.7.6.0

- Some radio configurations on the EFR32XG22 are not usable with RAIL address filtering and RAIL 802.15.4 filtering. Add an assert to catch those cases.

Added in release 2.7.4.0

- Added new RAIL_BLE_ConfigAoxAntenna API to configure which GPIO pins are used for Bluetooth LE AoX.

Added in release 2.7.3.0

- Revamped Features and rail_features.h, providing runtime RAIL_SupportsSomeFeature() APIs for each of the features as some features may be restricted to certain chips within a family. Also added more consistent RAIL_SUPPORTS_ compile-time synonyms for the features while retaining the existing RAIL_FEAT_ defines for backwards compatibility. These defines can now be used in C code and not just preprocessor \#if statements.
- Added support for new BGM220 modules.

Added in release 2.7.1.0

- Added the new RAIL_STREAM_10_STREAM RAIL_StreamMode_t to allow you to send a 1010 stream for debugging.
- Added a new function, RAIL_StartTxStreamAlt, which allows the specific antenna to be specified for a stream transmit.
- Added new RAIL_RX_PACKET_HANDLE_OLDEST_COMPLETE packet handle to allow the user to get a reference to the oldest unreleased complete packet.
- Added a new External_Thermistor interface to RAIL. This allows access the user to connect and read the impedance of an external thermistor on supported chips.
- Added RAIL_IEEE802154_ConvertRssiToEd() and RAIL_IEEE802154_ConvertRssiToLqi() to assist Zigbee 802.15.4 certification testing.

Added in release 2.7.0.0

- Added a new PA mode which will attempt to automatically choose the PA which consumes the least amount of current to reliably produce the requested output power. See RAIL_EnablePaAutoMode() for details.
- On EFR32xG12 thru EFR32xG14, added support for 802.15.4G-2012 SUN PHY dynamic frame payload whitening on reception and transmit based on the PHY header's Data Whitening flag setting. This feature is automatically enabled when RAIL_IEEE802154_ConfigGOptions()' RAIL_IEEE802154_G_OPTION_GB868 is enabled, and assumes the radio configuration specifies the appropriate whitening algorithm and settings.
- On EFR32xG12 thru EFR32xG14, added support for 802.15.4G-2012 SUN PHY dynamic frame payload 2/4-byte FCS (CRC) on reception and transmit based on the PHY header's FCS Type flag setting. This feature is automatically enabled when RAIL_IEEE802154_ConfigGOptions()' RAIL_IEEE802154_G_OPTION_GB868 is enabled. The radio configuration's (single) CRC algorithm settings are ignored, overridden by RAIL.
- On EFR32xG12 thru EFR32xG14, 802.15.4 AutoACK behavior has also been updated so transmitted immediate ACKs reflect the Whitening and 2/4-byte FCS of the received frame being acknowledged.
- Added two new APIs, RAIL_GetSyncWords and RAIL_ConfigSyncWords(), to allow getting and setting the sync word configuration of your PHY at runtime.
- Added RAIL_TX_OPTION_CCA_ONLY to just perform CCA (CSMA/LBT), stopping short of automatically transmitting when the channel is clear.

- Added support for a new RAIL_EVENT_TX_STARTED, triggered when the first preamble bit is about to go on-air. Also included the ability to retrieve the equivalent RAIL_PACKET_TIME_AT_PREAMBLE_START timestamp of that event from the event's handler via RAIL_GetTxTimePreambleStart(). Note: This new event shifted the bit positions of some events in RAIL_Events_t.

- Added an API, RAIL_StopInfinitePreambleTx, that can stop an infinite preamble on PHYs configured to use infinite preambles.

- Added additional information to the packet trace stream for the Z-Wave protocol to indicate what region is currently active to help with decoding.

- Added support for RFSENSE Selective (OOK) mode for supported chips, which currently includes only EFR32xG22 devices. Please refer to RAIL internal chip specific documentation for more details.

4.2 Improvements

Changed in release 2.7.9.0

- Changed RAIL Timer Synchronization over sleep on the EFR32xG21 to use the RTCC instead of the PRORTC to reduce current consumption in EM2.

Changed in release 2.7.4.0

- Relax constraints in RAIL to allow calling RAIL_SetRxTransitions, RAIL_SetTxTransitions, RAIL_ScheduleRx, and all of the RAIL_BLE_ConfigPhy before the radio is completely IDLE.

Changed in release 2.7.3.0

- Updated the pa_customer_curve_fits.py helper script to work with Python 3 as well as Python 2.

- Calling RAIL_ConfigSleep() with RAIL_SLEEP_CONFIG_TIMERSYNC_ENABLED on chips that use the PRORTC for synchronization (EFR32xG13 and newer) will now only configure the choose the LF clock source if the PRORTC IRQ is disabled. This allows for other code to safely configure the PRORTC like the Silicon Labs generic sleep timer.

Changed in release 2.7.1.0

- The RAIL_GetRadioEntropy() API will now ensure a valid radio configuration has been loaded using RAIL_ConfigChannels() since it can cause problems if the radio is used before this.

- Changed the value of RAIL_FREQUENCY_OFFSET_INVALID from -1 to -32768 since -1 is a reasonable frequency offset to pass to RAIL_SetFreqOffset(). Also added convenience definitions RAIL_FREQUENCY_OFFSET_MIN and RAIL_FREQUENCY_OFFSET_MAX to specify the valid range of offset values the radio supports.

Changed in release 2.7.0.0

- Changed RAIL_GetRxTimePreambleStartAlt(), RAIL_GetRxTimeSyncWordEndAlt(), and RAIL_GetRxTimeFrameEndAlt() to properly update its pPacketDetails' RAIL_PacketTimeStamp_t::timePosition to reflect the adjusted RAIL_PacketTimeStamp_t::packetTime rather than leaving it as RAIL_PACKET_TIME_DEFAULT.

- Enforced and clarified that RAIL_Init() should not be called more than once per protocol.

- Clarified documentation of the RAIL_EVENT_RX_ACK_TIMEOUT event and RAIL_AutoAckConfig_t::ackTimeout period which extends only to packet sync word detection of an expected ACK, not packet completion of that ACK.

- Documented RAIL's internal 16-packet metadata FIFO which exists on EFR32 platforms supplementing the receive FIFO of packet data. Refer to Data_Management and efr32_main for details. Included is support for a new RAIL_EVENT_RX_FIFO_FULL, triggered with any packet completion event in which the receive FIFO or packet metadata FIFO are full. This tells the application it must free up the oldest packets/data ASAP to reduce the chance of RAIL_EVENT_RX_FIFO_OVERFLOW (however, overflow may already have occurred). Note: This new event shifted the bit positions of some events in RAIL_Events_t.

4.3 Fixed Issues

Fixed in release 2.7.9.0

ID #	Description
645641	Fixed an issue on EFR32xG22 where a state transition to receive after a transmit from EM2 sleep would drop packets.
671168	Fixed an issue on EFR32xG22 parts where current consumption would be higher than expected after radio activity.

Fixed in release 2.7.8.0

ID #	Description
639833	Fixed a potential radio hang on a corrupted Bluetooth LE packet when doing Bluetooth LE AoX.

Fixed in release 2.7.7.0

ID #	Description
499216	Fixed an issue with the EFR32xG21 medium power PA that could cause the output power to be too low for certain power level and ramp time combinations.

Fixed in release 2.7.6.0

ID #	Description
484374	Fixed regression from 2.7.1 on EFR32xG21 where BLE did not include packet sync word on PTI.

Fixed in release 2.7.4.0

ID #	Description
465096	Fixed an issue where RAIL_Idle() was not properly terminating an ongoing RAIL_StartAverageRssi() process.
467589	Updated default dynamic multiprotocol (DMP) transition timings to make them work with Zigbee and Bluetooth LE DMP applications. The previously suggested workaround of adding 30 μs to the default transition time using RAIL_SetTransitionTime() is no longer required.
471373	Fixed an issue on the EFR32xG22 where loading IEEE 802.15.4 and BLE PHYs without a reset would cause an assert with error code RAIL_ASSERT_CACHE_CONFIG_FAILED.
471955	Fixed an issue with BGM220 modules that caused an assert, RAIL_ASSERT_INVALID_MODULE_ACTION, when using them in previous releases.

Fixed in release 2.7.3.0

ID #	Description
464735	Closed tiny timing window on EFR32xG13 that might corrupt PTI appended info when idling the radio.
469015	Fixed an issue on the EFR32xG21 that could cause the RAIL_GetRadioEntropy() function to return the same first 4 bytes when called with the radio off after a reset.

Fixed in release 2.7.2.0

ID #	Description
456338	Fixed an issue with RAIL state transitions where an internal timer wrapping could cause incorrect transition times. This error would previously affect a maximum of one packet every 15 minutes.
460062	Fixed a RAIL_ScheduleRx() issue where RAIL_EVENT_RX_SCHEDULED_RX_END might not be posted when the Rx RAIL_StateTransitions_t::error transition is to RAIL_RF_STATE_IDLE and the Rx window ended during receipt of an erroneous packet.

Fixed in release 2.7.1.0

ID #	Description
444205	Fixed a transmit-from-idle issue with RAIL_StartCcaCsmaTx() or RAIL_StartCcaLbtTx(), which would always fail when the RAIL_StateTiming_t::idleToRx is configured below the minimum the radio is capable of achieving (typically 65-100 microseconds depending on platform).
452628	Fixed an issue where idling the radio from an Rx antenna diversity mode would consume extra power.
452690	Fixed an issue where Rx antenna diversity could be left enabled after switching to a radio configuration that lacks diversity support.

Fixed in release 2.7.0.0

ID #	Description
197573	Suppressed extraneous RAIL_EVENT_TX_START_CCA events that might occur during long CCA durations. Now only one such event should occur per CCA try.
411498	RAIL_StartAverageRssi() now returns RAIL_STATUS_INVALID_STATE if called when the radio is not idle, enforcing its documented behavior.
417340	Fixed an issue where RAIL_RxPacketDetails_t::isAck would incorrectly be set true for non-ACK or unexpected ACK packets received successfully (e.g. when RAIL_IEEE802154_ACCEPT_ACK_FRAMES is enabled) or aborted while waiting for the expected ACK. Note that when RAIL_RX_OPTION_IGNORE_CRC_ERRORS is in effect, an expected ACK includes one that fails CRC, and will have isAck set true.
418493	RAIL_ConfigRadio will now return RAIL_STATUS_INVALID_STATE if called from the inactive config in dynamic multiprotocol instead of returning success but not applying the change.
427934	Fixed a race condition that could cause a device to not re-enable frame detection after an Rx overflow event if the overflow was processed and cleared very quickly.
430081	Fixed an issue where the first Clear Channel Assessment (CCA) of a CSMA/LBT transmit from radio idle state would consistently fail when the RAIL_CsmaConfig_t::ccaBackoff or RAIL_LbtConfig_t::lbtBackoff time is smaller than the RAIL_StateTiming_t::idleToRx time.
436163	Fixed a post-receive transition timing issue for received packets that were on the air longer than 32 milliseconds. AutoACK turnaround timing should now behave properly at low data rates.
437054	Fixed an issue with the pa_customer_curve_fits.py that caused values below -12 to not be considered when computing the fit. Re-generated default, Silicon Labs-provided curves to consume this fix, resulting in minor changes to the lowest-power segment in curve-fit based PA's. If using a custom power curve created using the documentation in AN1127 customers should re-run the script on the already collected output data to get slightly more accurate curves.
441635	Return the correct RAIL_TxPowerMode_t value of RAIL_TX_POWER_MODE_NONE from RAIL_GetTxPowerConfig if called before RAIL_ConfigTxPower.
446289	Fixed RAIL_IDLE_ABORT to idle the radio sooner when in RAIL_RF_STATE_RX, especially now that RAIL_RxChannelHoppingConfigEntry_t::delay can extend the time in that state.
447578	Fixed an issue where setting a transmit power over the maximum allowed for a given channel would result in no change in the output power instead of using the maximum allowed value.
450187	Fixed an issue where calling RAIL_Idle() with RAIL_IDLE_FORCE_SHUTDOWN while in receive with channel hopping enabled could corrupt some internal channel hopping state and trigger a bus fault or other radio problems.

4.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
475184	On the EFR32xG22 the receiver is not automatically re-calibrated if temperature changes significantly while sitting in receive. This could cause the radio to go deaf if temperature changes significantly while in receive. The calibration will run on every entry to receive so protocols that do not sit in receive will not be largely impacted by this.	Avoid long running receives or call RAIL_Calibrate(railHandle, NULL, RAIL_CAL_TEMP) periodically to force a re-calibration while in a long running receive until this is fixed.

4.5 Deprecated Items

None

4.6 Removed Items

None

NVM3 (Non-Volatile Memory Version 3)

5.1 New Items

None

5.2 Improvements

Changed in release 2.7.8.0

The nvm3_repack function has been updated to use the cache to reduce the computational overhead to find objects that must be copied before a page erase. When the cache is valid (large enough to cache all the objects in the store), this improvement will reduce the overhead in the data copy part of nvm3_repack and reduce the maximum time for the function call.

Changed in release 2.7.7.0

During execution, the nvm3_repack function will do either nothing, copy data or erase a page. To limit the execution time, the copy part is now split into several calls where each call will never copies more than max-object-size number of bytes. This is done to limit the time when interrupts are disabled if the default locking functions are used.

5.3 Fixed Issues

Fixed in release 2.7.1.0

ID #	Description
453206	Fixed an issue in NVM3 that could cause an error in the content of an existing data object after firmware upgrade from GSDK 2.4.0 to GSDK 2.5.0 or higher. The problem is related to the fact that NVM3 released in GSDK 2.4.0 and earlier could write data objects in a format that was off spec. Although the format was incorrect, all functions handled the format correctly. From GSDK 2.5.0 this format issue was fixed, but as a side effect, repacking data written with GSDK 2.4.0 or earlier could unintentionally cause data error. The fix ensures that both the pre- and the post-GSDK 2.5.0 formats are handled correctly.

5.4 Known Issues in the Current Release

None

5.5 Deprecated Items

None

5.6 Removed Items

None

mbed TLS

6.1 New Items

Added in release 2.7.3.0

Added a routine that validates if public-key is on the curve in mbedtls_ecdh_compute_shared for Series 2 products.

Added in release 2.7.0.0

Added hardware-acceleration plugin for AES-GCM on EFR32xG21 products.

6.2 Improvements

Changed in release 2.7.1.0

Updated mbed TLS library to version 2.7.12

6.3 Fixed Issues

Fixed in release 2.7.0.0

ID	Description
445608	Hardware-accelerated AES-GCM on EFR32xG22 parts was not adhering to mbedTLS API contract.

6.4 Known Issues in the Current Release

ID #	Description	Workaround
429985	ECDSA curve P224 is not supported for xG21 and xGM21 products.	None

6.5 Deprecated Items

None

6.6 Removed Items

None

Gecko Bootloader

7.1 New Items

Added in release 2.7.4.0

- bootloader_secureBootEnforced() is added to check if signature verification on the application is enforced before every boot.

Added in release 2.7.3.0

- Added a new internal storage bootloader sample application bootloader-storage-internal-single-352k for devices with 352kB internal flash.
- Added certificate support for secure boot and GBL image parser for EFR32xG22.
- The size of ParserContext_t for Series-2 devices has been increased to 524 bytes to support certificate-based authentication of GBL images.

Added in release 2.7.2.0

- Added a function bootloader_getCertificateVersion for reading certificate version of the bootloader for Series 2 products.
- BOOTLOADER_STORAGE_VERIFICATION_CONTEXT_SIZE is added to btl_interface_storage.h, which indicates the size required to store verification context.

Added in release 2.7.0.0

- Added an AppBuilder plugin option APPLICATION_VERIFICATION_SKIP_EM4_RST to skip verification of the application only if the device has been to EM4.
- Added certificate support for secure boot and GBL image parser for EFR32xG21.
- Added first stage binaries for xG13 and xG14 devices that works with the second stage bootloader placed in the main flash.
- Added EZSP GPIO plugin.
- Added EUART driver.

7.2 Improvements

Changed in release 2.7.1.0

- The RMU reset level for soft resets is configured to be EXTENDED on Series-1 devices

Changed in release 2.7.0.0

- The size of ParserContext_t for EFR32xG21 has been increased to 524 bytes to support certificate boot chain.

7.3 Fixed Issues

None

7.4 Known Issues in the Current Release

None

7.5 Deprecated Items

None

7.6 Removed Items

None

Other Gecko Platform Software Components

8.1 New Items

None

8.2 Improvements

Changed in release 2.7.4.0

- Added caching mechanism in Sleep Timer to avoid re-querying the frequency at which the hardware timer runs once the initialization is done.

Changed in release 2.7.3.0

- Created a new port for Sleep Timer that allows to use the Radio internal real-time clock as the hardware timer and hence free up RTCC for application's usage. In order to select this port, the configuration `SL_SLEEPTIMER_PERIPHERAL` in `sl_sleep-timer_config.h` must be set to `SL_SLEEPTIMER_PERIPHERAL_PRORTC`.
- Improvement of the round-robin scheduling mechanism. We now only restart a round-robin timer in the switching context hook. Timing should now be more accurate by being reset at the end of a context switch. Stopping the round-robin only happens if a context switch is needed.
- Reduced the number of conversions between the Sleep Timer ticks and the Micrium OS Kernel ticks, in the Kernel.
- Modified internal Sleep Timer tick count data types so tick count cannot overflow before 272 years.

Changed in release 2.7.0.0

- Updated LwIP package to version 2.1.2.

8.3 Fixed Issues

Fixed in release 2.7.7.0

ID #	Description
497581	Fixed compilation issue in Sleep Timer's PRORTC HAL in some conditions.

Fixed in release 2.7.6.0

ID	Description
486079	Fixed a potential issue in Sleep Timer where, in very specific conditions, there could have been undesirable behavior when creating a timer and going to sleep.
482920	Fixed a bug where there was a chance a null pointer could be de-referenced is some rare occasions.

Fixed in release 2.7.5.0

ID	Description
483020	Fixed a bug where the value returned by sl_sleeptimer_get_tick_count64() could be incorrect.

Fixed in release 2.7.4.0

ID	Description
477471	In the MPU, fixed a bug where, in some cases, we would not be covering the entirety of the main RAM.
472312	In Micrium OS Kernel, fixed a bug if OS_CFG_MAX_PRIO was smaller or equal to 8.
475327	Add fix for potential bug with some compiler optimization in slists.

Fixed in release 2.7.3.0

ID	Description
465205	In the Sleep Timer, fixed a bug where, under some circumstances, a timer could expire too early.
467213	Fixed a bug where an ISR stack underflow could occur when FPU was used.
467599	Fixed some static analysis issue in the Micrium OS Kernel and in Micrium OS LIB String module.
462961	Added checks in `OSTimeDlyHMSM()` to make sure there could be no overflow occurring.
461249	Fixed issues with round-robin and Time Quantats in Micrium OS Kernel.

Fixed in release 2.7.2.0

ID	Description
459690	Fixed bug in EFP driver sl_se_init() function. This function could return SL_STATUS_OK even though an error was detected.
452516	Fixed problem when using the MPU that prevented LDREX and STREX instructions from working properly.

Fixed in release 2.7.0.0

ID	Description
340730	GLIB now offers a GLIB_invertBitmap() function, which inverts every bit in the bitmap.
340726	GLIB: Bugfix in optimized drawing of bitmaps for inverse monochrome displays.
340053	GLIB now supports characters wider than 16 pixels.
451076	DMD_writeData now consistently treats a bit value of 1 as white and 0 as black for monochrome displays.

8.4 Known Issues in the Current Release

None

8.5 Deprecated Items

None

8.6 Removed Items

None

Hardware Support

<u>Added in version 2.7.4.0</u>

- Added board support for BRD4183B.

<u>Added in version 2.7.3.0</u>

- Added board support for BRD4184 and BRD4184A.

Legal

10.1 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 and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System. 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.

10.2 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®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave and others are trademarks or registered trademarks of Silicon Labs.

ARM, CORTEX, Cortex-M0+, Cortex-M3, Cortex-M33, Cortex-M4, TrustZone, Keil and Thumb are trademarks or registered trademarks of ARM Holdings.

Zigbee® and the Zigbee logo® are registered trademarks of the Zigbee Alliance.

Bluetooth® and the Bluetooth logo® are registered trademarks of Bluetooth SIG Inc.

All other products or brand names mentioned herein are trademarks of their respective holders.