



# Bluetooth® SDK 2.13.4.0 GA

## Gecko SDK Suite 2.7

### April 22, 2020

Silicon Labs is a leading vendor in Bluetooth hardware and software technologies, used in products such as sports and fitness, consumer electronics, beacons, and smart home applications. The core SDK is an advanced Bluetooth 5-compliant stack that provides all of the core functionality along with multiple API to simplify development. The core functionality offers both standalone mode allowing a developer to create and run their application directly on the SoC, or in NCP mode allowing for the use of an external host MCU.

Extensions to the SDK include Bluetooth Mesh and Apple® HomeKit® for customers seeking the additional capabilities.

These release notes cover SDK version(s):

- 2.13.4.0 released on April 22, 2020
- 2.13.3.0 released on March 20, 2020
- 2.13.2.0 released on February 21, 2020
- 2.13.1.0 released on January 24, 2020
- 2.13.0.0 released on December 13, 2019



#### KEY FEATURES

- Adds directional priority PTA support for Wi-Fi Coex
- Updates to the Bluetooth Mobile app
- NVM3 support on EFR32[B|M]G1x devices
- New part support - EFR32[B|M]G22 and Thunderboard EFR32BG22
- Bluetooth 5.2 compliance

## Compatibility and Use Notices

If you are new to the Silicon Labs Bluetooth SDK, see [Using This Release](#).

### Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 8.30.1

- Using wine to build with the `IarBuild.exe` command line utility or IAR Embedded Workbench GUI on macOS or Linux could result in incorrect files being used due to collisions in wine's hashing algorithm for generating short file names.
- Customers on macOS or Linux are advised not to build with IAR outside of Simplicity Studio. Customers who do should carefully verify that the correct files are being used.

GCC (The GNU Compiler Collection) version 7.2.1, provided with Simplicity Studio.

**Contents**

- 1 New Items ..... 3
  - 1.1 New Features ..... 3
  - 1.2 New APIs ..... 3
- 2 Improvements ..... 5
  - 2.1 Changed APIs ..... 5
- 3 Fixed Issues ..... 6
- 4 Known Issues in the Current Release ..... 8
- 5 Deprecated Items ..... 9
- 6 Removed Items ..... 10
- 7 Using This Release ..... 11
  - 7.1 Installation and Use ..... 11
  - 7.2 Support ..... 11
- 8 Legal ..... 12
  - 8.1 Disclaimer ..... 12
  - 8.2 Trademark Information ..... 12

# 1 New Items

## 1.1 New Features

### Added in release 2.13.4.0

#### Advertising channel index randomization

Randomly select advertising channel usage order.

### Added in release 2.13.3.0

#### New Part Support

This release supports the new EFR32[B|M]G22 device family and Thunderboard EFR32BG22. Additionally, support for BRD4180B and BRD4181B radio boards is added.

### Added in release 2.13.2.0

#### Advertiser Random Address

Applications can set an advertiser's random device address used in advertising packets.

### Added in release 2.13.0.0

#### NVM3 support on EFR32[B|M]G1x devices

NVM3 is now supported on EFR32[B|M]G1x devices in addition to PS Store. These two data stores cannot be used simultaneously. The sample applications in the Bluetooth SDK continue to use PS Store by default. To replace PS Store with NVM3 please refer to [AN1135: Using Third Generation Non-Volatile Memory \(NVM3\) Data Storage](#).

Usage of NVM3 on EFR32[B|M]G1x devices is recommended only for new designs as there is no migration mechanism from PS to NVM3, when a new firmware is flashed via OTA or UART DFU. Doing DFU from PS to NVM3 will cause all PS data to be lost, which includes e.g., bonding information.

#### Sleep Timer

The sleep timer (`sl_sleeptimer.h`) is a platform component providing single-shot and periodic timer services. It also provides APIs for reading current tick count and conversions between ticks and milliseconds.

A Bluetooth application must include sleep timer component because the Bluetooth stack initializes and uses it for deep sleep. For more details, please refer to a Bluetooth SDK example such as the `soc-empty`.

## 1.2 New APIs

For additional documentation and command descriptions please refer to the [Bluetooth Software API Reference Manual](#).

### Added in release 2.13.4.0

`cmd_gatt_server_enable_capabilities`

`cmd_gatt_server_disable_capabilities`

`cmd_gatt_server_get_enabled_capabilities`

### Added in release 2.13.2.0

`cmd_le_gap_set_advertise_random_address`

`cmd_le_gap_clear_advertise_random_address`

### Added in release 2.13.1.0

`cmd_coex_set_directional_priority_pulse`

### Added in release 2.13.0.0

cmd\_gatt\_server\_get\_mtu

cmd\_le\_connection\_read\_channel\_map

cmd\_coex\_set\_parameters

## 2 Improvements

### 2.1 Changed APIs

#### Changed in release 2.13.2.0

##### **Bluetooth Stack Initialization**

A hardware error is returned if HFRCODPLL is not configured to 80 MHz and set as the system clock source on EFR32[B|M]G21x.

#### Changed in release 2.13.0.0

##### **cmd\_system\_linklayer\_configure**

New configuration key system\_linklayer\_config\_key\_set\_priority\_table is added for changing task priorities at run time.

##### **cmd\_sm\_store\_bonding\_configuration**

The default maximum allowed bonding count is changed to 13.

### 3 Fixed Issues

#### Fixed in release 2.13.4.0

ID #	Description
467342	Fix OTA DFU update on BGM210P032.
469935	<p>Fix the issue that the thermometer RTOS example does not enter to deep sleep. The root cause is that the application's idle hook uses OSIdleEnterHook(void), which is not called by the kernel if HW interrupt wakes up the device and no tasks are triggered by the interrupt. The fix is to use OSIdleContextHook(void). In this function, use a loop to set the stack to sleep, e.g.,</p> <pre> void SleepAndSyncProtimer(); void OSIdleContextHook(void) {     while (1) {         /* Put MCU in the lowest sleep mode available, usually EM2 */         SleepAndSyncProtimer();     } } </pre> <p>See thermometer RTOS example from the Bluetooth SDK.</p>
472920	Improve extended advertising packet reception during connections with very short intervals.
475785	Fix an issue in Bluetooth RTOS adaptation that may cause incorrect conversion from sleeptimer frequency to OS frequency.
477676	Improve the performance of Bluetooth connections using very short intervals on EFR32[B M]G22.

#### Fixed in release 2.13.3.0

ID #	Description
457174	Bluetooth applications are able to use RTCC on EFR32[B M]G13 and EFR32[B M]G21 again. To use RTCC, configure sleep timer to use PRORTC (#define SL_SLEEPTIMER_PERIPHERAL SL_SLEEPTIMER_PERIPHERAL_PRORTC).
466477	Improve scanning robustness.
467371	Fix an issue that may cause the stack to use an invalid Bluetooth device address stored in PS key after the application is migrated from an old version of the SDK. This fix checks if an address is stored in PS, and removes the PS key if the address is invalid.
467479	Prevent from erasing the next flash page after storage slot in OTA if the upgrade firmware is too big to fit storage slot.
468416	Function gecko_sleep_for_ms() now returns the sleep time in milliseconds. Previously a tick count value is returned.

#### Fixed in release 2.13.2.0

ID #	Description
431452	TX power is now set to correct values correlating to the requests. Previously, when the requested TX Power was above 0 dBm, the actual transmit power may have deviated randomly.
444469	Fix an issue that may cause the application to be unable to disconnect a Bluetooth connection when writing large amounts of data to characteristic values using the write-without-response command.
446074	Fixed an issue that causes the stack to be unable to establish new connections after multiple connection drops.
453828	Fix an issue that causes the application to be unable to confirm a characteristic indication in a low memory situation when writing large amounts of data to characteristic values using the write-without-response command.
454960	Fix the tick rate RTOS_TICK_HZ in rtos_bluetooth.c to the correct value (1000).
458216	Fix an issue in gecko_sleep_for_ms() which limits the sleep time to no more than 131 seconds.
461654	BGAPI soft timers now work correctly regardless of the value SL_SLEEPTIMER_FREQ_DIVIDER is configured to in sleep timer.

**Fixed in release 2.13.1.0**

ID #	Description
448020	In <code>system_set_tx_power</code> command, the stack now sets and returns the nearest possible value if the requested value is out of range.
450903	Fix the issue when application compiles mbedTLS into source that uses AES with padding, mbedTLS context objects in Bluetooth stack may be corrupted and cause hard faults.
450922	Use of GPCRC in Bluetooth stack is now thread safe.
452831	Fix an issue in Apploader that causes application OTA update to fail if the data length in the first data packet is not multiple of four.
455666	Fix a compatibility issue in Apploader with bootloader v1.10 on EFR32[B M]G21. This issue causes OTA update to fail and the device to stay in OTA mode indefinitely.

**Fixed in release 2.13.0.0**

ID #	Description
234520	In <code>cmd_gatt_read_characteristic_value_by_uuid</code> command, if multiple characteristic values are received, an <code>evt_gatt_characteristic_value</code> event is generated for each value. Previously only one event is generated for this case.
281984	First advertisement packet is now sent with the right timing in dynamic multi-protocol use cases.
335894	Documentation fix to the <code>lolen</code> field in a BGAPI command header binary. The field specifies the minimal number of bytes the parameters take. It does not include the array data length if the command has an array type parameter.
347844	Event <code>evt_user_message_to_host</code> can now be sent during user command handling in NCP target application.
358171	Fix a random advertising hang issue which was observed when the application optimization build option was disabled or the device was exposed to low temperature.
408097	Extended advertising now works in dynamic multi-protocol use cases.
415583	The stack now returns out of memory error if the application sends a characteristic notification to all connections in low memory situation. Previously wrong state error was returned.
420866	Improves robustness when streaming large data packets to and from a device which does not support data length extension.
421731	When sending a characteristic value notification to all connections using <code>cmd_gatt_server_send_characteristic_notification</code> command, the stack now returns success if devices are connected but none has enabled the notification. Previously wrong state error was returned.
425187	<code>SL_BT_prefix</code> is added into the include guard in <code>bg_version.h</code> .
425532	Fix whitelisting when an extended advertising packet does not contain an auxiliary pointer.
430225	Fix CTUNE control using PS key on EFR32[B M]G2x devices.
430752	Fix an issue that the stack may lose a periodic advertising synchronization because the scanner uses highest priority for secondary tasks. This was solved by lowering the priority.
434406	Remove the use of variable length array in the stack.
436243	Fix a HW initialization issue in the soc-dtm example that causes incorrect timeout for incoming data.
441445	Fix an advertisement data update race condition which causes no advertisement is sent out.
445627	Repeated bonding attempts handling is now connection specific. After first failed attempt there is cool down period of 1 s. After second attempt this is doubled to 2 s then to 4 s etc. The maximum wait time is 64 s.
450515	Fix an issue that causes BGTool does not run on Linux.

## 4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
243009	With certain events, GCC breakpoints cannot be set.	Change optimization level to none in project settings
337467	MGM12P has poor signal strength when doing OTA with Aploader.	None
360313	Default RAIL assertion (RAILCb_AssertFailed) enters an infinite loop causing the stack not responsive.	Override RAIL assertion function in application space following <a href="#">these instructions</a> .
361592	The sync_data event does not report TX power.	None
368798	Example soc-thunderboard_sense_2 does not print float values correctly in GCC project.	In project C/C++ settings, enable the printf float option.
396308	In NCP, BGAPI may be out of sync between the host and target when the target is reset in the middle of sending a BGAPI message to host. The out of sync issue could cause the host unable to communicate with the target.	In NCP target, add a delay to flush NCP Tx queue after a system reset command is received.
456701	Before gecko_init() is called, enabling busfaults generation on access to peripherals with clocks disabled causes a crash.	Enable busfaults generation after gecko_init().
463724	The Network Analyzer fails to decode a connection request packet when the connection is opened on LE Coded PHY.	None
464918	The stack has a performance issue that may cause advertising to fail if the application is compiled without optimization (-O0).	None
470424	Connection drop may happen in rare cases if multiple simultaneous connections use the same connection interval.	Try to use different connection intervals in different connections to improve the robustness.



## 5 Deprecated Items

### Deprecated in release 2.13.4.0

#### Thunderboard Sense (SLTB001) and Thunderboard React (RD-0057-0201)

Support for Thunderboard Sense (SLTB001) and Thunderboard React (RD-0057-0201) is now deprecated and will be removed in the next major SDK release.

## 6 Removed Items

None

## 7 Using This Release

This release contains the following

- Silicon Labs Bluetooth stack library
- Bluetooth sample applications

This SDK depends on Gecko Platform. The Gecko Platform code provides functionality that supports protocol plugins and APIs in the form of drivers and other lower layer features that interact directly with Silicon Labs chips and modules. Gecko Platform components include EMLIB, EMDRV, RAIL Library, NVM3, and mbedTLS. Gecko Platform release notes are available through Simplicity Studio's Launcher Perspective, under this SDK's **Release Notes** doc header.

For more information about the Bluetooth SDK see [QSG139: Getting Started with Bluetooth® Software Development](#). If you are new to Bluetooth see [UG103.14: Bluetooth LE Fundamentals](#).

### 7.1 Installation and Use

A registered account at Silicon Labs is required in order to download the Silicon Labs Bluetooth SDK. You can register at [https://siliconlabs.force.com/apex/SL\\_CommunitiesSelfReg?form=short](https://siliconlabs.force.com/apex/SL_CommunitiesSelfReg?form=short).

Stack installation instruction are covered in [QSG139: Getting Started with Bluetooth® Software Development](#).

Use the Bluetooth SDK with the Silicon Labs Simplicity Studio V4 development platform. Simplicity Studio ensures that most software and tool compatibilities are managed correctly. Install software and board firmware updates promptly when you are notified.

Documentation specific to the SDK version is installed with the SDK. Additional information can often be found in the [knowledge base articles \(KBAs\)](#). API references and other information about this and earlier releases is available on <https://docs.silabs.com/>.

### 7.2 Support

Development Kit customers are eligible for training and technical support. You can use the [Silicon Labs Bluetooth LE web page](#) to obtain information about all Silicon Labs Bluetooth products and services, and to sign up for product support.

You can contact Silicon Laboratories support at <http://www.silabs.com/support>.

## 8 Legal

### 8.1 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.

### 8.2 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.

Apple and HomeKit are registered trademarks of Apple Inc.

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