# *Bluetooth*® SDK 3.1.0.0 GA

## Gecko SDK Suite 3.1

## December 9, 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 5.1-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.

These release notes cover SDK version(s):

3.1.0.0 released December 9, 2020

Bluetooth

**KEY FEATURES**

- Angle of Arrival General Availability
- LE Power Control
- Support for CMSIS-RTOS2
- New NCP tool - Bluetooth NCP Commander

## 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

## 1.1 New Features

### Added in release 3.1.0.0

**Angle of Arrival (AoA) and Angle of Departure (AoD)**

Beginning with this release, the Bluetooth stack supports Bluetooth 5.1 AoA and AoD features. The SDK provides soc_aoa_asset_tag and ncp_aoa_locator example applications for evaluating the AoA functionality. The AoD is supported only at the Bluetooth stack level and the SDK does not provide any sample application in this release. For more information see the examples, API reference, and *AN1296: Application Development with Silicon Labs' RTL Library* in the SDK.

**LE Power Control**

The Bluetooth stack now supports the Bluetooth 5.2 LE Power Control feature. Both procedures, Power Control Request and Power Change Indication, are supported. In order to use these procedures, the connection peer entity must support the same procedures.

At the receiver side, the user can specify so-called 'Golden RSSI range' for each PHY. Based on the configuration the receiver side requests the transmitter to adjust transmit power such that the received signal level would be within the configured range.

At the transmitter side, the user is allowed to configure the transmit power range. The transmitter shall not use a transmit power out of the configured range even if the connection peer entity sends such a request.

The power control feature can be used seamlessly with or without adaptive frequency hopping.

The Golden RSSI range and the transmit power range can be configured in the bluetooth_feature_power_control software component.

The global minimum and maximum TX power are also configurable in the Bluetooth stack component.

**FreeRTOS Support**

The Bluetooth stack can also run on FreeRTOS in this release. This is achieved by using the CMSIS-RTOS2 in the RTOS adaptation software component of the Bluetooth stack. User application can choose to run the stack either on Micrium OS or FreeRTOS. The Bluetooth SDK provides soc_thermometer_micrium and soc_thermometer_freertos examples.

Additionally, the Bluetooth stack RTOS adaptation is now in its own software component and it supports configuring the Bluetooth RTOS task priorities.

## 1.2 New APIs

For additional documentation and command descriptions please refer to the Bluetooth API reference in the SDK installation or the online API reference specific to the SDK version you are using. The most up-to-date version is at https://docs.silabs.com/bluetooth/latest/.

### Added in release 3.1.0.0

**Minimum and maximum TX power configuration:** In the Bluetooth stack component (sl_bluetooth_config.h), including items SL_BT_CONFIG_MIN_TX_POWER and SL_BT_CONFIG_MAX_TX_POWER.

**LE Power Control configuration:** In the bluetooth_feature_power_control component (sl_bt_power_control_config.h).

**Bluetooth RTOS task priority configuration:** In the Bluetooth_rtos_adaptation component (sl_bt_rtos_config.h).

**sl_bt_system_get_version command:** Returns the firmware version information, the same as the sl_bt_evt_system_boot event returns.

**sl_bt_system_set_tx_power command:**Sets the global minimum and maximum radiated TX power levels for Bluetooth.

**sl_bt_system_get_tx_power_setting command:** Returns the global minimum and maximum radiated TX power levels for Bluetooth.

**sl_bt_connection_set_power_reporting command:** Enables or disables the transmit power reporting for the local device on a Bluetooth connection.

**sl_bt_connection_set_remote_power_reporting command:** Enables or disables the transmit power reporting for the remote device on a Bluetooth connection.

**sl_bt_connection_get_tx_power command:** Returns the transmit power of the local device on a Bluetooth connection.

**sl_bt_connection_get_remote_tx_power command:** Reads the transmit power of the remote device on a Bluetooth connection.

**sl_bt_evt_connection_get_remote_tx_power_completed event:** Indicates that reading remote transmit power operation has completed.

**sl_bt_evt_connection_tx_power event:**Reports a transmit power change of the local device on a Bluetooth connection.

**sl_bt_evt_connection_remote_tx_power event:** Reports a transmit power change of the remote device on a Bluetooth connection.

**sl_bt_gatt_server_send_notification command:** Sends a characteristic value notification to a remote GATT client.

**sl_bt_gatt_server_send_indication command:** Sends a characteristic value indication to a remote GATT client.

**sl_bt_gatt_server_notify_all command:** Sends a characteristic value notifications or indications to all connected remote GATT clients.

**sl_bt_gatt_server_read_client_configuration command:** Reads a remote GATT client's current value of the client characteristic configuration of a characteristic.

**sl_bt_evt_gatt_server_indication_timeout event:** Indicates the indication to a remote GATT client has timed out.

**sl_bt_test_dtm_tx_v4 command:** Supports setting the TX power level in DTM transmitter test v4 of the Bluetooth specification.

**sl_bt_user_manager_event_filter command:** Used for managing the API event filter over the NCP.

# 2 Improvements

## 2.1 Changed APIs

### Changed in release 3.1.0.0

**Behavior change in TX power setting command**

The new command sl_bt_system_set_tx_power replaces the sl_bt_system_set_max_tx_power command,with the change that the set TX powers are the radiated values from the antenna, i.e., without the TX RF path gain. Previously,, the TX powers in sl_bt_system_set_max_tx_power command had the TX RF path gain applied.

**RSSI value in sl_bt_connection_get_rssi event**

The RSSI value in this event is the median of the last seven measured RSSI values. Previously the last RSSI value was returned.

**PA auto-mode**

The Bluetooth stack now uses PA auto-mode for power level handling to optimizing the operation processing time.

**Simultaneous scanning on LE 1M and Coded PHYs**

The command sl_bt_scanner_start() now supports simultaneous scanning on LE 1M and Coded PHYs. The stack alternates the scanning on two PHYs by switching the PHY at every scan interval.

**Bluetooth component ID and name change**

The Bluetooth component 'bluetooth' is changed to 'bluetooth_feature_default'. The feature selection in this component remains unchanged.

**Maximum number of advertisers in ncp_empty application**

The maximum number of advertisers for user is increased to 4 in the ncp_empty application.

**Separate headers for common BGAPI protocol types and generated types**

Common types of the BGAPI protocol are separated into their own header file sl_bgapi.h, which is automatically included from sl_bt_api.h. No change is required in applications.

**Prefix of Bluetooth API types**

Bluetooth types and defines that do not have the "sl_bt_" prefix are deprecated and replaced by types and defines that include the "sl_bt_" prefix to the name. A compatibility header, sl_bt_api_compatibility.h, is automatically included from sl_bt_api.h to maintain source code compatibility to existing code until it has migrated to use the new names.

# 3 Fixed Issues

**Fixed in release 3.1.0.0**

| ID # | Description |
|---|---|
| 457227 | Optimize the use case where an advertisement task was delayed if the scanning task was running at the same time. |
| 485334 | Fix an issue that caused the Bluetooth connection to drop when setting a breakpoint |
| 465088 | BGTool now works on MacOS 10.13 (High Sierra) and older Mac versions. |
| 488915 | The Bluetooth stack performs a version exchange before doing feature exchange to improve interoperability with devices that do not properly handle the control packets from the device in slave role. |
| 489436 | Update the slave feature request handling to improve the interoperability.<br>Previously, the Bluetooth stack sent the slave feature request automatically when connected as the peripheral role. However, some existing master Bluetooth 4.0 implementations freeze. In order to be compliant with those master implementations, the Bluetooth stack executes the version information exchange procedure before executing the slave feature request. If the peer master is version 4.1 or newer, the slave executes the feature request procedure. For older peer implementations the slave feature request is not executed. Instead, the slave assumes the master will support the encryption feature until the master indicates otherwise. |
| 622436 | Fix the issue that timers started using Bluetooth APIs are not fired at the correct time. |
| 628916 | Fix duplicate whitelist address issue. |
| 631755 | Set Bluetooth stack component to always require HFXO. |
| 635058 | Fix an invalid packet transmission issue where the Bluetooth link layer may transmit invalid packet content if radio interrupts are delayed. |
| 635417 | Fix an issue that causes hanging during CTE reception if a corrupted packet header length was read. |
| 637232 | Fix a case where the connection might be dropped with "unspecified error" code. |
| 645453 | Fix an issue on EFR32BG22 where receiving after a transmit from EM2 sleep based on RAIL state transitions would result in dropped packets. |
| 648010 | Fix an issue in the Bluetooth stack that causes the TX RF path gain to be applied twice. |
| 649592 | Update the maximum connection to 32 in the Bluetooth configuration. |

# 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 |
|---|---|---|
| 337467 | MGM12P has poor signal strength when doing OTA with Apploader. | None |
| 361592 | The sync_data event does not report TX power. | None |
| 368403 | If setting CTE interval to 1, a CTE request should be sent in every connection interval. But it is sent only in every second connection interval. | None |
| **641122** | **The Bluetooth stack component does not provide a configuration for RF antenna path.** | **This is an issue specifically for BGM210P. One workaround is to manually update the configuration in sl_bluetooth_config.h in text edit mode.**<br>**If the OTA with Apploader is used, include the bluetooth_feature_ota_config component in application project. Call command sl_bt_ota_set_rf_path() to set the RF path for OTA mode.** |
| **650079** | **LE 2M PHY on EFR32[B\|M]G12 and EFR32[B\|M]G13 doesn't work with smartphones using the Mediatek Helio chip due to an interoperability issue.** | **No workaround exists. For application development and testing, the disconnection can be avoided by disabling 2M PHY with sl_bt_connection_set_preferred_phy() or sl_bt_connection_set_default_preferred_phy().** |

# 5 Deprecated Items

## Deprecated in release 3.1.0.0

BGTool

BGTool has been deprecated and it will be removed in the next major release. A new tool, Bluetooth NCP Commander, has been introduced as the replacement. It includes a modern, intuitive, web-based user interface as well as a smart console with IntelliSense and built-in API documentation.

**API command sl_bt_system_set_max_tx_power**

This command is replaced by the command sl_bt_system_set_tx_power, which supports setting the minimum and maximum TX powers.

**API command sl_bt_gatt_server_ send_characteristic_notification**

This command is replaced by sl_bt_gatt_server_send_notification, sl_bt_gatt_server_send_indication, and sl_bt_gatt_server_notify_all commands.

**API command sl_bt_test_dtm_tx**

This command is replaced by sl_bt_test_dtm_tx_v4, which supports setting the TX power level.

# 6 Removed Items

## Removed in release 3.1.0.0

Function sl_bt_wait_event() has been removed from this release. This function is the blocking mode of receiving Bluetooth events and it becomes incompatible with the application software architecture beginning with the v3.0 SDK. Applications can still receive events in blocking mode by calling function sl_bt_pop_event() in a loop until a valid event is returned by the function.

# 7 Special Notices

**Silicon Labs Apple® HomeKit®**

Silicon Labs implementation of Apple HomeKit is not currently available in GSDK 3. Silicon Labs is committed to providing an Apple HomeKit solution and is investigating ways to add support for it in a future release. Customers developing products targeting HomeKit applications may continue development using GSDK 2.7.x releases and upgrade to GSDK 3 when HomeKit is available.

# 8 Using This Release

This release contains the following

- Silicon Labs Bluetooth stack library
- Bluetooth sample applications

For more information about the Bluetooth SDK see QSG169: Bluetooth® SDK v3.x Quick Start Guide. If you are new to Bluetooth see UG103.14: Bluetooth LE Fundamentals.

## 8.1 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.

Stack installation instruction are covered in the Simplicity Studio 5 online User's Guide.

Use the Bluetooth SDK v3.x with the Silicon Labs Simplicity Studio 5 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. Only use Simplicity Studio 4 with Bluetooth SDK v2.13.x and lower.

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/.

## 8.2 Support

Development Kit customers are eligible for training and technical support. 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.

# 9 Legal

## 9.1 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.

## 9.2 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.