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 5.3 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):

3.3.0.0 GA released December 15, 2021

**Compatible Compilers:**

IAR Embedded Workbench for ARM (IAR-EWARM) version 8.50.9

- 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 10.2.1, provided with Simplicity Studio.

**Compatibility and Use Notices**

For information about security updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the Silicon Labs Release Notes page. Silicon Labs also strongly recommends that you subscribe to Security Advisories for up-to-date information. For instructions as well as notes on using Secure Vault features, or if you are new to the Silicon Labs Bluetooth SDK, see Using This Release.

**KEY FEATURES**

- Bluetooth v5.3 qualified
- New Co-Processor Communication (CPC) transport for RCP/HCI
- RTOS support in RCP mode
- Improved tools for Angle-of-Arrival evaluation and development
- Interoperability testing example added to the SDK
## Contents

1. New Items  .................................................................................................................................................................................. 2  
   1.1 New Features .................................................................................................................................................................. 2
   1.2 New APIs ........................................................................................................................................................................ 3

2. Improvements ............................................................................................................................................................................. 4  
   2.1 Changed Items ................................................................................................................................................................. 4
   2.2 Changed APIs ................................................................................................................................................................... 5

3. Fixed Issues ............................................................................................................................................................................... 6

4. Known Issues in the Current Release ...................................................................................................................................... 7

5. Deprecated Items ...................................................................................................................................................................... 8

6. Removed Items ........................................................................................................................................................................... 9

7. Using This Release .................................................................................................................................................................. 10  
   7.1 Installation and Use ........................................................................................................................................................... 10
   7.2 Security Information .......................................................................................................................................................... 10
   7.3 Support ............................................................................................................................................................................. 11
1 New Items

1.1 New Features

Added in release 3.3.0.0

bluetooth_feature_legacyAdvertiser

Provides advertisements using legacy advertising PDUs. This component is the result of separating functionality from the existing component bluetooth_feature_advertiser to support better application size optimization. Bluetooth APIs from this component start with the prefix sl_bt_legacyAdvertiser. When the application uses this component, some advertising commands stop working. See section 2.1 Changed Items for more details.

bluetooth_feature_extendedAdvertiser

Provides advertisements using extended advertising PDUs. This component is the result of separating functionality from the existing component bluetooth_feature_advertiser to support better application size optimization. Bluetooth APIs from this component start with the prefix sl_bt_extendedAdvertiser. When the application uses this component, some advertising commands stop working. See section 2.1 Changed Items for more details.

bluetooth_feature_periodicAdvertiser

Provides periodic advertisements. This component is the result of separating functionality from the existing component bluetooth_feature_advertiser to support better application size optimization. Bluetooth APIs from this component start with the prefix sl_bt_periodicAdvertiser. When the application uses this component, some commands stop working. See section 2.1 Changed Items for more details. It replaces existing component bluetooth_feature_periodic_adv.

More application support in GATT Configurator

The GATT Configurator also supports Bluetooth network coprocessor (NCP) host projects for EFR32 and desktop hosts.

Coprocessor Communication (CPC) transport in Radio Coprocessor (RCP) mode

In addition to the Bluetooth SIG’s standard UART (H4) transport protocol, HCI commands can now be sent over Silicon Labs’ proprietary CPC transport protocol.

RTOS in RCP

RTOS support is automatically enabled if a kernel-component is present in the project configuration. A new kernel thread is instantiated for the Link Layer, and it blocks on a semaphore waiting for event flags to be passed in. The BTLE_LL_EventRaise function will be called to pass flags to the Link Layer thread, which will modify the flags bitmap and raise the semaphore to allow the Link Layer thread to handle the events.

The HCI-UART component will also instantiate a kernel thread for reading HCI messages from UART. Current implementation is simplistic and blocks waiting for new messages to arrive from UART.

RCP Controller de-initialization

The controller can be de-initialized with a new vendor command.

RCP Dynamic memory allocations

Multiple new vendor-specific commands are added for dynamically allocating and deallocating memory objects in the controller.

New tools for Angle-of-Arrival (AoA) development

The new Positioning tool is able to track Asset Tags in 3D using the RTL library.

New Project Type to support Angle-of-Arrival (AoA) development

The Direction Finding Project is used to provide a working directory for all AoA-related configuration files. It also provides a dashboard to overview / edit / import / export configuration files.
New Example Applications

Bluetooth RCP – CPC: The Bluetooth RCP – CPC example uses the Silicon Labs proprietary CPC (Co-Processor Communication) protocol to transport HCI commands to the host, in contrast to Bluetooth RCP that uses the standard H4 transport protocol to transport the HCI commands.

1.2 New APIs

**Added in release 3.3.0.0**

- **sl_btAdvertiserConfigure command**: Configure the legacy and extended advertising of an advertising set. It consists of configurations that are common for legacy and extended advertising.
- **sl_btLegacyAdvertiserSetData command**: Set user-defined legacy advertising data or scan response packets.
- **sl_btLegacyAdvertiserGenerateData command**: Ask the stack to generate legacy advertising data and scan response packets.
- **sl_btLegacyAdvertiserStart command**: Start an undirected legacy advertising with specified connection mode.
- **sl_btExtendedAdvertiserSetPhy command**: Set advertising PHYs of an extended advertising.
- **sl_btExtendedAdvertiserSetData command**: Set user-defined data for an extended advertising.
- **sl_btExtendedAdvertiserSetLongData command**: Set long user-defined data for an extended advertising.
- **sl_btExtendedAdvertiserGenerateData command**: Ask the stack to generate the extended advertising data.
- **sl_btExtendedAdvertiserStart command**: Start an undirected extended advertising with specified connection mode and flags.
- **sl_btPeriodicAdvertiserSetData command**: Set data for a periodic advertising.
- **sl_btPeriodicAdvertiserSetLongData command**: Set long data for a periodic advertising.
- **sl_btPeriodicAdvertiserStart command**: Start a periodic advertising.
- **sl_btPeriodicAdvertiserStop command**: Stop a periodic advertising.
- **sl_btDtmTxCw command**: Start a transmitter test for a custom wave type.
- **New option in smConfigure command**: Bit 5 for configuring if a pairing should prefer just works or authenticated pairing when both options are possible based on the settings.
- **RCP HCI VS_SiliconLabs_Deinit 0xfc25**: De-initialize all memory allocated by the controller.
- **RCP HCI VS_SiliconLabs_Allocate_Connections 0xfc20**: Allocate memory objects for connections.
- **RCP HCI VS_SiliconLabs_Allocate_Advertisers 0xfc21**: Allocate memory objects for advertisers.
- **RCP HCI VS_SiliconLabs_Allocate_Addresses 0xfc22**: Allocate memory objects for addresses.
- **RCP HCI VS_SiliconLabs_Allocate_PeriodicAdv 0xfc23**: Allocate memory objects for periodic advertisers.
- **RCP HCI VS_SiliconLabs_Allocate_PeriodicScan 0xfc24**: Allocate memory objects for periodic scanners.
2 Improvements

2.1 Changed Items

Changed in release 3.3.0.0

Unsupported Bluetooth APIs in certain conditions

Some existing advertising commands will stop working (returning the SL_STATUS_NOT_SUPPORTED error) when one or more of the new bluetooth_feature_legacy_advertiser, bluetooth_feature_extended_advertiser, or bluetooth_feature_periodic_advertiser components are used by the application. This behavior change offers better flash consumption optimization by reducing the application binary size when extended advertising is not used by the application. These commands continue to work and are backwards compatible if none of the three components exists in the application in this release. The commands are as follows:

- `sl_bt_advertiser_set_phy` command: replaced by `sl_bt_extended_advertiser_set_phy`.
- `sl_bt_advertiser_set_configuration` command: replaced by `sl_bt_advertiser_configure`.
- `sl_bt_advertiser_clear_configuration` command: replaced by `sl_bt_advertiser_configure`.
- `sl_bt_advertiser_set_data` command: replaced by `sl_bt_legacy_advertiser_set_data` for legacy advertising PDUs, `sl_bt_extended_advertiser_set_data` for extended advertising PDUs, and `sl_bt_periodic_advertiser_set_data` for periodic advertising PDUs.
- `sl_bt_advertiser_set_long_data` command: replaced by `sl_bt_extended_advertiser_set_long_data` for extended advertising PDUs, and `sl_bt_periodic_advertiser_set_long_data` for periodic advertising PDUs.
- `sl_bt_advertiser_start` command: replaced by `sl_bt_legacy_advertiser_generate_data`, `sl_bt_legacy_advertiser_start`, `sl_bt_extended_advertiser_generate_data`, and `sl_bt_extended_advertiser_start`.
- `sl_bt_advertiser_start_periodic_advertising` command: replaced by `sl_bt_periodic_advertiser_start`.
- `sl_bt_advertiser_stop_periodic_advertising` command: replaced by `sl_bt_periodic_advertiser_stop`.

These commands will be deprecated in the next major SDK release.

Removal of RAIL assertion errors from sl_bt_evt_system_error event

In earlier versions of Bluetooth SDKs, the Bluetooth stack had a callback implementation overwriting the RAIL assertion RAILCb_AssertFailed that by default enters into an eternal loop. When the callback is called after an assertion has happened, the stack will generate an sl_bt_evt_system_error event with the SL_STATUS_BT_RADIO (value 0x040C on v3.0.0 – v3.2.3 SDK versions, and 0x0199 on 2.x SDK versions) as the reason and the RAIL error code in data field.

Beginning with this release, the Bluetooth stack no longer has the callback implementation. Applications should overwrite the RAILCb_AssertFailed callback directly, if RAIL assertions need to be handled. This change makes the assertion approach visible to user applications and applications have more flexibility in handling assertions.

Bluetooth SoC Thermometer

Bluetooth SoC Thermometer examples no longer uses the LED.

Improved PHY update in the Controller

The PHY update request has been improved for the case where another LLCP procedure is active at the same time. Previously the procedure would fail. Now it will wait and start after the other procedure is complete. This improvement is in the Bluetooth SDK v3.2.3 and onwards.

Memory optimization in CTE component

The memory usage of the CTE component has been optimized so that the IQ sample buffer is allocated only if the CTE receiver is initialized. The buffer is no longer allocated for the CTE transmitter. This improvement is in the Bluetooth SDK v3.2.2 and onwards.

Export feature in Bluetooth host examples

The export feature has been generalized for Bluetooth host example projects. The `make export` target collects every resource into the export folder that is needed by the example project, while keeping the GSDK installation folder structure unchanged. It is advised to export the examples before starting to work on them because of the following benefits:

- Changes in the (config) files during development will not affect the GSDK content.
- Makes transfer to a Raspberry Pi easy.
- Multiple instances can coexist, e.g., for testing different variants.
• It is clear at a glance which files belong to the example project.

GATT database support in host examples

It is now possible to store GATT database information in host applications the same way as it is stored in an SoC application. The editable .btconf file is stored under the /config folder, and it can be turned into gatt.db.c and gatt.db.h using the 'make gattdb' command in the command line. gatt.db.c and gatt.db.h are automatically parsed by the application, and the corresponding Dynamic GATT Bluetooth API commands are automatically called to build the GATT database on the target on every reset.

AoA Bluetooth host examples

Statically-allocated lists are replaced with dynamic lists in AoA Bluetooth host examples. This eliminates the need for manually adjusting the maximum number of list entries. Configuration parameters like AOA_MAX_TAGS, MAX_NUM_TAGS, MAX_NUM_LOCATORS have been removed.

aoa_multilocator example is renamed to positioning.

RCP Controller Reset

The HCI reset command does a soft reset by default, which only resets the controller state. Users can optionally do a full hard reset.

Bluetooth SoC Interoperability Test

This demo is now opened up as an application example to support customization of the code.

Improved Tools for Angle-of-Arrival (AoA) Development

AoA Analyzer has new functions.
• One asset tag can be associated with multiple estimators to compare and evaluate different settings of the RTL library.
• Multiple charts can be added to analyze the incoming data from different aspects, such as phase distribution over the antennas or antenna SNR over time.
• Incoming data can be recorded and played back for future analysis.
• The pseudo spectrum (i.e., the probability density function of the Angle of Arrival) can be calculated for recorded data, which helps identifying reflections.

2.2 Changed APIs

Changed in release 3.3.0.0

Output parameter NULL pointer safe

The Bluetooth API now accepts NULL pointers to output parameters. If an application does not need an output value that is merely informational, the application can pass a NULL pointer to that output parameter and avoid allocating storage for the output.
### 3 Fixed Issues

#### Fixed in release 3.3.0.0

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>668850</td>
<td>Fix a memory allocation issue in the CTE component that may cause memory overflow if more connections are allocated after the stack has been started.</td>
</tr>
<tr>
<td>703489</td>
<td>Return correct sl_status_t error codes in the sl_bt_dfu_flash_upload command and sl_bt_dfu_boot_failure event.</td>
</tr>
<tr>
<td>725498</td>
<td>The crash issue in the connection-based aoa_locator application was fixed by ID 725480 in the Bluetooth SDK v3.2.2 and onwards.</td>
</tr>
<tr>
<td>726925</td>
<td>Fix Amazon FreeRTOS BLE HAL issue where the pairing state change callback was missing in a situation where the remote device has deleted the bonding and is bonding again but the EFR device still has a bonding entry for the remote device. This fix has been released in the Bluetooth SDK v3.2.3 and onwards.</td>
</tr>
<tr>
<td>728217</td>
<td>Fix Amazon FreeRTOS BLE HAL issue that prevents a remote client from reading or writing GATT characteristic descriptors that were created with encrypted read/write permissions.</td>
</tr>
<tr>
<td>730008</td>
<td>The LE Read Number of Supported Advertising Sets HCI command now returns the correct response. The issue has been fixed in the Bluetooth SDK v3.2.2 and onwards.</td>
</tr>
<tr>
<td>731458</td>
<td>Fix an issue that causes Bluetooth connection disconnections when the scanning is enabled. This issue has been fixed in the Bluetooth SDK v3.2.2 and onwards.</td>
</tr>
<tr>
<td>733994</td>
<td>Fix a GATT procedure failure on the GATT server role when the MicriumOS is used. The root cause of the failure is a bug in the MicriumOS's CMSIS RTOS2 port which has been fixed in this release.</td>
</tr>
<tr>
<td>736359</td>
<td>Fix an issue that a EFR32[B</td>
</tr>
<tr>
<td>742842</td>
<td>Fixed Bluetooth HCI reset reason retrieval process. This fix involves reading the RMU_RSTCAUSE register and checking the appropriate bit that tells us if a system reset has occurred, implying that a reset command was issued by the host. This replaces the prior method of storing the reset reason in memory (and eliminating the possibility of it being overwritten).</td>
</tr>
<tr>
<td>743872</td>
<td>There is an issue in the sl_bt_test_dtm_tx_v4 command such that it does not support a TX power higher than 14 dBm for unmodulated carrier testing. This is solved by a new command sl_bt_test_dtm_tx_cw() for unmodulated carrier testing specifically.</td>
</tr>
<tr>
<td>748062</td>
<td>Fix a COEX enabling issue in the RCP Bluetooth controller component when the COEX feature is used.</td>
</tr>
<tr>
<td>748653</td>
<td>Fix a static GATT attribute table generation issue when a descriptor value is empty in the GATT XML file. This issue caused an incorrect response to the ATT_READ_BY_TYPE_RSP PDU procedure.</td>
</tr>
<tr>
<td>751536</td>
<td>The HCI Read Local Supported Features Command returns a correct response now. The bit 3 of octet 25 in the bitmap is reserved by the Bluetooth specification and it is cleared in the response.</td>
</tr>
</tbody>
</table>
## 4 Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on [https://www.silabs.com/products/software](https://www.silabs.com/products/software).

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>337467</td>
<td>MGM12P has poor signal strength when doing OTA with Apploader.</td>
<td>None</td>
</tr>
<tr>
<td>361592</td>
<td>The sync_data event does not report TX power.</td>
<td>None</td>
</tr>
<tr>
<td>368403</td>
<td>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.</td>
<td>This is an issue specifically for BGM210P. One workaround is to manually update the configuration in <code>sl_bluetooth_config.h</code> in text edit mode. If the OTA with Apploader is used, include the <code>bluetooth_feature_ota_config</code> component in application project. Call command <code>sl_bt_ota_set_rf_path()</code> to set the RF path for OTA mode.</td>
</tr>
<tr>
<td>641122</td>
<td>The Bluetooth stack component does not provide a configuration for RF antenna path.</td>
<td>None</td>
</tr>
<tr>
<td>650079</td>
<td>LE 2M PHY on EFR32[B</td>
<td>M]G12 and EFR32[B</td>
</tr>
<tr>
<td>682198</td>
<td>The Bluetooth stack has an interoperability issue on the 2M PHY with a Windows PC.</td>
<td>No workaround exists. For application development and testing, the disconnection can be avoided by disabling 2M PHY with <code>sl_bt_connection_set_preferred_phy()</code> or <code>sl_bt_connection_set_default_preferred_phy()</code>.</td>
</tr>
<tr>
<td>695148</td>
<td>Bluetooth soft timer doesn’t work when Bluetooth on-demand start feature is enabled.</td>
<td>Use the simple timer component in the Bluetooth SDK or the sleeptimer platform service.</td>
</tr>
<tr>
<td>730692</td>
<td>4-7% packet error rate is observed on EFR32[B</td>
<td>M]G13 devices when RSSI is between -25 and -10 dBm. The PER is nominal (as per the datasheet) both above and below this range.</td>
</tr>
<tr>
<td>753951</td>
<td>Opening a connection to an advertisement using extended advertising PDUs does not succeed sometimes.</td>
<td>Use a retry mechanism for opening the connection.</td>
</tr>
<tr>
<td>756253</td>
<td>The RSSI value on a Bluetooth connection returned by the Bluetooth API is incorrect on EFR32[B</td>
<td>M]1, EFR32[B</td>
</tr>
<tr>
<td>756562</td>
<td>The Bluetooth scanning has an issue on the LE Coded PHY on EFR32[B</td>
<td>M]G21 devices. After some time (minutes or hours) of scanning, a RAIL assertion (RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO) may happen and the scanning stops.</td>
</tr>
</tbody>
</table>
5 Deprecated Items

Deprecated in release 3.3.0.0

**bluetooth_feature_periodic_adv component**: deprecated and replaced by the Bluetooth_feature_periodic_advertiser component.

**sl_bt_dfu_reset command**: Deprecated. It is no longer supported by the latest bootloader.
6 Removed Items

**Removed from release 3.3.0.0**

- `sl_bt_system_set_max_tx_power` command
- `sl_bt_gatt_server_send_characteristic_notification` command
- `sl_bt_test_dtm_tx` command

aoa_compass demo application. It is replaced with the AoA Analyzer tool.
7 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.

7.1 Installation and Use

The Bluetooth SDK is provided as part of the Gecko SDK (GSDK), the suite of Silicon Labs SDKs. To quickly get started with the GSDK, install Simplicity Studio 5, which will set up your development environment and walk you through GSDK installation. Simplicity Studio 5 includes everything needed for IoT product development with Silicon Labs devices, including a resource and project launcher, software configuration tools, full IDE with GNU toolchain, and analysis tools. Installation instructions are provided in the online Simplicity Studio 5 User's Guide.

Alternatively, Gecko SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.

Simplicity Studio installs the GSDK by default in:

- (Windows): C:\Users\<NAME>\SimplicityStudio\SDKs\gecko_sdk
- (MacOS): /Users/<NAME>/SimplicityStudio/SDKs/gecko_sdk

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 Security Information

Secure Vault Integration

When deployed to Secure Vault High devices, sensitive keys such as the Long Term Key (LTK) are protected using the Secure Vault Key Management functionality. The table below shows the protected keys and their storage protection characteristics.

<table>
<thead>
<tr>
<th>Wrapped Key</th>
<th>Exportable / Non-Exportable</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Long Term Key (LTK)</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Local Long Term Key (legacy only)</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Remote Identity Resolving Key (IRK)</td>
<td>Exportable</td>
<td>Must be Exportable for future compatibility reasons</td>
</tr>
<tr>
<td>Local Identity Resolving Key</td>
<td>Exportable</td>
<td>Must be Exportable because the key is shared with other devices.</td>
</tr>
</tbody>
</table>

Wrapped keys that are marked as “Non-Exportable” can be used but cannot be viewed or shared at runtime.

Wrapped keys that are marked as “Exportable” can be used or shared at runtime but remain encrypted while stored in flash.

For more information on Secure Vault Key Management functionality, see AN1271: Secure Key Storage.
Security Advisories

To subscribe to Security Advisories, log in to the Silicon Labs customer portal, then select Account Home. Click HOME to go to the portal home page and then click the Manage Notifications tile. Make sure that ‘Software/Security Advisory Notices & Product Change Notices (PCNs)’ is checked, and that you are subscribed at minimum for your platform and protocol. Click Save to save any changes.

7.3 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.
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

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and “Typical” parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A “Life Support System” is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications. Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information
Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, “the world’s most energy friendly microcontrollers®, Redpine Signals®, WiSeConnect®, n-Link, ThreadArch®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, 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.