



Bluetooth[®] LE SDK 3.3.1.0 GA

Gecko SDK Suite 4.0

January 26, 2022

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):

3.3.1.0 GA released January 26, 2022
3.3.0.0 GA released December 15, 2021



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

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](#).

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.

Contents

- 1 New Items2
 - 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 Issues6
- 4 Known Issues in the Current Release8
- 5 Deprecated Items9
- 6 Removed Items10
- 7 Using This Release11
 - 7.1 Installation and Use.....11
 - 7.2 Security Information11
 - 7.3 Support.....12

1 New Items

1.1 New Features

Added in release 3.3.0.0

bluetooth_feature_legacy_advertiser

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_legacy_advertiser`. When the application uses this component, some advertising commands stop working. See section [2.1 Changed Items](#) for more details.

bluetooth_feature_extended_advertiser

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_extended_advertiser`. When the application uses this component, some advertising commands stop working. See section [2.1 Changed Items](#) for more details.

bluetooth_feature_periodic_advertiser

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_periodic_advertiser`. 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_bt_advertiser_configure command: Configure the legacy and extended advertising of an advertising set. It consists of configurations that are common for legacy and extended advertising.

sl_bt_legacy_advertiser_set_data command: Set user-defined legacy advertising data or scan response packets.

sl_bt_legacy_advertiser_generate_data command: Ask the stack to generate legacy advertising data and scan response packets.

sl_bt_legacy_advertiser_start command: Start an undirected legacy advertising with specified connection mode.

sl_bt_extended_advertiser_set_phy command: Set advertising PHYs of an extended advertising.

sl_bt_extended_advertiser_set_data command: Set user-defined data for an extended advertising.

sl_bt_extended_advertiser_set_long_data command: Set long user-defined data for an extended advertising.

sl_bt_extended_advertiser_generate_data command: Ask the stack to generate the extended advertising data.

sl_bt_extended_advertiser_start command: Start an undirected extended advertising with specified connection mode and flags.

sl_bt_periodic_advertiser_set_data command: Set data for a periodic advertising.

sl_bt_periodic_advertiser_set_long_data command: Set long data for a periodic advertising.

sl_bt_periodic_advertiser_start command: Start a periodic advertising.

sl_bt_periodic_advertiser_stop command: Stop a periodic advertising.

sl_bt_dtm_tx_cw command: Start a transmitter test for a custom wave type.

New option in sm_configure 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 to 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 3.3.1.0

Parameter `min_ce_length`

The parameter `min_ce_length` of the commands `sl_bt_connection_set_default_parameters` and `sl_bt_connection_set_parameters` is taken into use for prioritizing the Link Layer task scheduling in a situation where other tasks need to run immediately after the connection event.

Function `sl_bt_external_signal()`

The API function `sl_bt_external_signal()` now returns an `sl_status_t` error code to indicate whether the external signal event was successfully queued. This allows the application to detect errors and retry later if the event generation fails due to temporary lack of resources.

Event `sl_bt_evt_system_boot`

The event `sl_bt_evt_system_boot` no longer reports the bootloader version. The bootloader information can be read from the function `bootloader_getInfo()` of the `bootloader_interface` component.

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 3.3.1.0

ID #	Description
737702	Fix app_log.h linkage errors in C++ projects.
751931	A new connection will not be established if the connection request comes from a connected device.
758288	The sl_bt_connection_get_remote_tx_power command now correctly returns the current TX power level.
759061	Clarify that the coding scheme S=8 is used when advertising on the LE Coded PHY in the API documentation.
760055	Improve the connection stability when there are concurrent scanning and advertising activities.
760581	Fix the NCP host buffer overflow issue.
761124	Fix the SL_BT_CONFIG_RF_PATH_GAIN_TX configuration. Previously the value was used in reversed manner.
761125	The API function sl_bt_external_signal() now returns an sl_status_t error code to indicate whether the external signal event was successfully queued. This allows the application to detect errors and retry later if the event generation fails due to temporary lack of resources.
762670	Fix an LLCP procedure triggering issue where a queued procedure is not always started after the previous procedure ended.
773172	Clarify the connection event length parameters of the sl_bt_connection_set_default_parameters and sl_bt_connection_set_parameters commands in the API documentation.
773395	Fix an issue that the scanner state is not properly reset when issuing the HCI reset command.
773448	Fix an LE power control issue that may cease sending power control PDUs for a few minutes.
775949	Fix the issue in the sl_bt_test_dtm_tx_cw() command that the TX power could not be set above 12.7 dBm.

Fixed in release 3.3.0.0

ID #	Description
668850	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.
703489	Return correct sl_status_t error codes in the sl_bt_dfu_flash_upload command and sl_bt_dfu_boot_failure event.
725498	The crash issue in the connection-based aoa_locator application was fixed by ID 725480 in the Bluetooth SDK v3.2.2 and onwards.
726925	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.
728217	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.
730008	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.
731458	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.
733994	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.
736359	Fix an issue that a EFR32[B M]G21 device transmits invalid coded S=2 packets sometimes in a connection creation or during a connection with longer data packets.
742842	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).
743872	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.
748062	Fix a COEX enabling issue in the RCP Bluetooth controller component when the COEX feature is used.

ID #	Description
748653	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.
751536	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.

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>.

ID #	Description	Workaround
337467	MGM12P has poor signal strength when doing OTA with Aploader.	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. If the OTA with Aploader 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().
682198	The Bluetooth stack has an interoperability issue on the 2M PHY with a Windows PC.	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().
695148	Bluetooth soft timer doesn't work when Bluetooth on-demand start feature is enabled.	Use the simple timer component in the Bluetooth SDK or the sleeptimer platform service.
730692	4-7% packet error rate is observed on EFR32[B 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.	None
753951	Opening a connection to an advertisement using extended advertising PDUs sometimes does not succeed.	Use a retry mechanism for opening the connection.
756253	The RSSI value on a Bluetooth connection returned by the Bluetooth API is incorrect on EFR32[B M]1, EFR32[B M]12, EFR32[B M]13, and EFR32[B M]21 devices. On EFR32[B M]21 devices. It is about 8~10 dBm higher than the actual value, according to a measurement.	Install the "RAIL Utility, RSSI" component in the application project. This component provides a default RSSI offset for the chip that is applied at the RAIL level and can help to achieve more accurate RSSI measurements.
756562	The Bluetooth scanning has an issue on the LE Coded PHY on EFR32[B 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.	Reset the device when the RAIL assertion happens. See section 2.1 Changed Items for the RAIL assertion callback overwriting.

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.

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

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.

SILICON LABS Search Within the Support Portal for Cases, etc... SEARCH CATHERIN...

HOME CASES SOFTWARE RELEASES

Update Preference

WHAT EMAILS WOULD YOU LIKE TO RECEIVE?

Newsletters

- Community Monthly Newsletter
- Sales Newsletter
- Micrium Newsletter

Product Specific Notifications

- Product Information and Newsletter
- Software/Security Advisory Notices & Product Change Notices (PCNs)
- Technical Document Updates (Release Notes, Data Sheets, etc.)

SELECT THE PRODUCTS TO RECEIVE UPDATES FOR

Select/Unselect All

<input type="checkbox"/> Audio and Radio	<input type="checkbox"/> Power over Ethernet
<input type="checkbox"/> Interface	<input type="checkbox"/> Sensors
<input type="checkbox"/> Isolation	<input type="checkbox"/> TV and Video
<input type="checkbox"/> Modems and DAAs	<input type="checkbox"/> Voice
<input type="checkbox"/> Microcontrollers	<input type="checkbox"/> Wireless
<input type="checkbox"/> 8-bit MCUs <input checked="" type="checkbox"/> 32-bit MCUs	<input type="checkbox"/> Bluetooth Classic <input type="checkbox"/> Bluetooth Low Energy <input checked="" type="checkbox"/> Proprietary
<input type="checkbox"/> Timing	<input type="checkbox"/> Wi-Fi
<input type="checkbox"/> Clocks	<input type="checkbox"/> ZigBee and Thread
<input type="checkbox"/> Buffers	<input type="checkbox"/> Z-Wave
<input type="checkbox"/> Oscillators	
<input type="checkbox"/> CDR and PHY	

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.



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

www.silabs.com