Bluetooth® mesh SDK 1.7.4.0 GA

Gecko SDK Suite 2.7

March 17, 2021

Bluetooth mesh is a new topology available for Bluetooth Low Energy (LE) devices that enables many-to-many (m:m) communication. It's optimized for creating large-scale device networks, and is ideally suited for building automation, sensor networks, and asset tracking. Our software and SDK for Bluetooth development supports Bluetooth Mesh and Bluetooth 5 functionality. Developers can add mesh networking communication to LE devices such as connected lights, home automation, and asset tracking systems. The software also supports Bluetooth beaconing, beacon scanning, and GATT connections so Bluetooth mesh can connect to smart phones, tablets, and other Bluetooth LE devices.

These release notes cover SDK versions:

1.7.4.0 released March 17, 2021
1.7.3.0 released December 22, 2020
1.7.2.1 released November 25, 2020
1.7.2.0 released November 4, 2020
1.7.1.0 released September 2, 2020
1.7.0.0 released June 17, 2020

Bluetooth

KEY FEATURES

- New part support – EFR32[B|M]G22 and Thunderboard EFR32BG22
- Time model API and documentation
- Scheduler model API and documentation
- HSL model API and documentation

Compatibility and Use Notices

If you are new to the Silicon Labs Bluetooth mesh 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.

- Link-time optimization feature of GCC has been disabled, resulting in slight increase of image size

Contents

1 New Items

1.1 New Features

Added in release 1.7.2.0

Support for HSL models (HSL client, HSL server, HSL setup server, HSL hue server, HSL saturation server) has been added as an alpha release.

Added in release 1.7.1.0

Model qualification listing updated to include Time and Scheduler models

Added in release 1.7.0.0

Added support for EFR32[B|M]G22 and Thunderboard EFR32BG22 hardware. Note that the limited amount of RAM on [B|M]G22 devices restricts their use in some roles that may need comparatively large amounts of memory, such as relays, GATT proxies, and provisioners. [B|M]G22 devices are, however, well suited for battery-powered operation. Of the SDK example applications, SoC empty, SoC switch, SoC sensor client and SoC sensor server are supported on [B|M]G22 devices.

Support for scheduler models (Scheduler client, Scheduler server, Scheduler setup server) has been added as an alpha release.

A new memory configuration value for setting the application message queue length length has been added. The application message queue is used for delayed application messages, such as status responses.

1.2 New APIs

For additional documentation please refer to the Bluetooth Mesh Software API Reference Manual installed with the Bluetooth Mesh SDK.

Added in release 1.7.0.0

BGAPI commands and events for Scheduler models have been added. Note that Scheduler model functionality has been released as an alpha release.

Scheduler client model commands and events added:

mesh_scheduler_client_init(),
mesh_scheduler_client_deinit(),
mesh_scheduler_client_get(),
mesh_scheduler_client_get_action(),
mesh_scheduler_client_set_action(),
mesh_scheduler_client_status(),
mesh_scheduler_client_action_status().

Scheduler server and setup server model commands and events added:

mesh_scheduler_server_init(),
mesh_scheduler_server_deinit(),
mesh_scheduler_server_get(),
mesh_scheduler_server_get_action(),
mesh_scheduler_server_set_action(),
mesh_scheduler_server_action_changed().

2 Improvements

2.1 Changed APIs

Changed in release 1.7.4.0

Added BGAPI commands and events for testing replay protection list functionality during development.

The following BGAPI command has been added to the test BGAPI:

mesh_test_set_replay_protection_list_diagnostics()

The following BGAPI events have been added to the test BGAPI:

mesh_test_replay_protection_list_entry_set(),
mesh_test_replay_protection_list_entry_cleared(),
mesh_test_replay_protection_list_saved(),
mesh_test_replay_protection_list_full()

Added a command for reducing relayed data when PDUs are destined to a unicast address across a GATT bearer.

The following BGAPI command has been added to the proxy BGAPI:

mesh_proxy_optimization_toggle()

Changed in release 1.7.2.0

The behavior of mesh_prov_provision_device_with_address() and mesh_prov_provision_gatt_device_with_address() has been extended so that the “elements” parameter indicates the maximum number of elements the device to be provisioned may have, instead of an exact number of elements.

A BGAPI command has been added to the node BGAPI:

mesh_node_set_model_option()

An option for altering the behavior of the generic level model has been implemented.

A BGAPI command has been added to the generic client model BGAPI:

mesh_generic_client_init_hsl()

A BGAPI command has been added to the generic server model BGAPI:

mesh_generic_server_init_hsl()

A BGAPI command has been added to the test BGAPI

mesh_test_clear_replay_protection_list_entry()

Changed in release 1.7.1.0

A BGAPI command has been added to the test BGAPI:

mesh_test_get_replay_protection_list_entry()

Changed in release 1.7.0.0

A BGAPI command has been added to the Scene server BGAPI:

mesh_scene_server_reset_register()

A BGAPI command has been added to mesh node BGAPI:

mesh_node_set_iv_update_age()

3 Fixed Issues

Fixed in release 1.7.4.0

ID #	Description
655402	Reset element sequence numbers after IV recovery has been completed
666981	Fixed an issue with macOS code signing with utilities that are used when building applications
666959	Fixed an issue with a backward-incompatible change in persistently stored data for an embedded provisioner
670760	Fixed an issue with publishing using TTL=1
671151	Fixed high idle current consumption with low power nodes
674055	Fixed issues with key deployment to a newly provisioned node during key refresh
676055	Fixed an issue with relaying messages encrypted with friend security credentials
679139	Fixed a potential one-byte buffer overrun if key count is an odd number
680002	Fixed an issue in Battery Server model state handling

Fixed in release 1.7.3.0

ID #	Description
640696	Fixed reporting of delayed requests in LC server model
646954	Artificial limit on IV update age removed, as age of over 192 hours is needed for IV recovery
646982	Fixed poor error handling when node or provisioner is initialized more than once
649840	Fixed a regression in GATT proxy connection closing, caused by code change done for 604997
655727	Fix hardcoded default TTL on embedded provisioner to read the value from configuration model state instead

Fixed in release 1.7.2.1

This release includes the 1.7.2.0-specific documentation, including release note content, omitted from the 1.7.2.0 release.

Fixed in release 1.7.2.0

ID #	Description
462941	Removed internal limitations on key refresh state size
514579	Network Analyzer decoding issue when IV index is nonzero
621872	Fixed scene status remaining time representation in BGAPI
622673	Allow specifying a non-exact element count when provisioning a node with set address
624572	Improved GATT connection closing logic
625696	Fixed provisioning error codes to conform to specification when invalid data is detected
625784	Fixed heartbeat subscription state reporting to conform to specification when subscription is disabled
629268	Fixed IV index in GATT proxy messages during IV index update

Fixed in release 1.7.1.0

ID #	Description
487271	Network Analyzer segmented packet handling improved
494388	Generic model initialization now reports an error if no generic models are included in the project
495619	Added explicit RNG initialization in sample application startup
500250	Corrected large TAI-UTC delta value handling in Time models
522077	Fixed a link layer issue with events during disabled interrupts
604997	Fixed an issue with GATT connection cleanup if a timeout occurs

Fixed in release 1.7.0.0

ID #	Description
467080	BTMESH_HEAP_SIZE macro now takes model counts into account
470417	Fixed polymorphic GATT database capability settings clash with Mesh proxy and provisioning services
483973	PTI traces taken with xG21 hardware are fixed
485299	Fixed erroneous sending of periodic secure network beacons over GATT proxy connections
485543	Current scene number is no longer zeroed on scene recall
485545	Handle scene number zeroing properly when model state is updated
487324, 487339, 487626, 487632	Fixed errors in BTMESH_HEAP_SIZE calculations
489069	Fixed an error in releasing resources when stopping unprovisioned device beacons
490301	Fixed bad parameter validity check for heartbeat publishing request

4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
3878	Mesh GATT events visible to the application	Application can ignore BGAPI events related to GATT provisioning and proxying based on service and characteristic parameters
5662	Default device UUID does not conform to RFC4122	Customer needs to explicitly set UUID to a conformant one
339993	ISC file comments cause errors when generating code	Avoid using comments in ISC files
401550	No BGAPI event for segmented message handling failure	Application needs to deduce failure from timeout / lack of application layer response
418636	Issues with mesh_test local configuration state API (node identity, relay, network retransmission)	
454059	A large number of key refresh state change events are generated at the end of KR process, and that may flood NCP queue	Increase NCP queue length in the project
454061	Slight performance degradation compared to 1.5 in round-trip latency tests was observed	
454332	Missing Mesh-specific API for generating and receiving scan response data for GATT provisioning service advertisements	Use the LE GAP API
490276	mesh_node_get_uuid() does not work until mesh stack is initialized	Read the value after the stack is initialized
624514	Issue with re-establishing connectable advertising if all connections have been active and GATT proxy is in use	Allocate one more connection than is needed
650825	Issue with retransmissions when a model is publishing periodically	Set up retransmissions in the model state and trigger periodic publishing by an application timer
676798	**Automatic LPN polling may miss friendship termination deadline due to oscillator inaccuracy**	**Ensure application timer-based explicit polling takes place before friendship terminates, if the timeout for friendship termination is long**

5 Deprecated Items

Deprecated in release 1.7.2.0

Bluetooth Mesh SDK v1.x:

Starting with the next major release, the Bluetooth Mesh SDK version will be incremented to version 2.0.0, which is not backwards compatible with SDK version 1.x. This means that an application written with the SDK version 1.x will not work without modifications with the new SDK version v2.x.

The Bluetooth Mesh stack API will be modified as part of the upcoming Bluetooth Mesh SDK version 2.0.0. The changes will streamline functionality, further enhance usability, improve development experience, and simplify application development. These modifications also align with API conventions across other Silicon Labs software components, including not only wireless stacks but also lower level platform components such as RAIL, emlib, and emdrv.

While many of the new API commands/events have familiar naming to ensure a smooth migration, the structure and overall interface will have slight changes:

- commands/events will be prefixed with sl_btmesh instead of gecko_cmd_mesh.

  Example: `gecko_cmd_mesh_node_init` becomes `sl_btmesh_node_init`
- The new API uses unified SL_STATUS error codes that consists of common codes for platform and technology stacks as well as codes assigned in dedicated code spaces for a specific technology. Existing error codes are mapped to corresponding sl_status values, so no change is required if the existing application is using the error code enums currently defined in bg_errorcodes.h.
- Parameter naming and ordering is made consistent between commands where possible, to make development easier.

A full and comprehensive migration guide will be made available that explains all the details related to migration of applications into Silicon Labs Bluetooth Mesh SDK v2.0.

6 Removed Items

None.

7 Using This Release

This release contains the following

- Silicon Labs Bluetooth mesh stack library
- Bluetooth sample applications

If you are a first time user, see QSG148: Getting Started with Bluetooth® Mesh Software Development.

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.

Stack installation instruction are covered in QSG148: Getting Started with Bluetooth® Mesh Software Development.

Use the Bluetooth mesh 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. Use the Silicon Labs Bluetooth mesh web page to obtain information about all Silicon Labs Bluetooth products and services, and to sign up for product support.

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 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.

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®, 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.