Bluetooth® mesh SDK 1.6.3.0 GA

Gecko SDK Suite 2.7
May 15, 2020

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:

1.6.3.0 released May 15, 2020
1.6.2.0 released March 20, 2020
1.6.1.0 released February 12, 2020
1.6.0.0 released December 20, 2019

Bluetooth

KEY FEATURES

- Light Controller model API and documentation
- Scene model AP and documentation
- NVM3 is supported for Series 1 SoC projects
- Qualification updated for all models released as GA

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

- Usingwine 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 1.6.3.0

Added support for a Provisioner node to relinquish its Provisioner role by rebooting as a node, and to re-acquire its Provisioner role by rebooting as a Provisioner.

Added in release 1.6.2.0

Profile qualification listing updated.

Model qualification listing updated for all models released as GA.

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

Model-specific initialization of generic models has been added.

Additional Provisioner commands for manipulating the device database and key refresh procedure have been added.

Added in release 1.6.1.0

Models: qualification readiness for all implemented models.

Added in release 1.6.0.0

Models: support for light control models (LC client, LC server, LC setup server) has been added.

Models: support for scene models (scene client, scene server, scene setup server) has been added. Note that due to a Mesh Model specification limitation only a single instance of a Scene server model is supported.

New model qualification effort is ongoing, but not completed at the time of release.

Persistent storage: NVM3 is supported for series 1 hardware SoC projects

SDK light sample application has been updated to support light controller functionality and scene functionality

SDK switch sample application has been updated to support scene recall functionality

Mesh node functionality has been updated to support more than 240 replay protection list entries (amount is limited by the size of persistent storage and RAM available for Mesh)

Mesh provisioner functionality has been updated to support up to 512 device database entries (amount is limited by the size of persistent storage and RAM available for Mesh)

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 1.6.3.0

Enumerations for LC Server state machine state and LC Server debug event types have been added.

BGAPI commands and events for debugging LC Server state have been added.

LC Server commands added:

mesh_lc_server_get_lc_state(),
mesh_lc_server_set_event_mask()

LC Server events added:

mesh_lc_server_regulator_debug_info(),
mesh_lc_server_state_updated()

BGAPI command for querying the application keys bound to a model has been added.

Test command added:

mesh_test_get_local_model_app_bindings()

Added in release 1.6.2.0

Data types for signed and unsigned 64-bit numbers have been added.

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

Time client model commands and events added:

mesh_time_client_init(),
mesh_time_client_deinit(),
mesh_time_client_get_tai_utc_delta(),
mesh_time_client_get_time(),
mesh_time_client_get_time_role(),
mesh_time_client_get_time_zone(),
mesh_time_client_set_tai_utc_delta(),
mesh_time_client_set_time(),
mesh_time_client_set_time_role(),
mesh_time_client_set_time_zone(),
mesh_time_client_tai_utc_delta_status(),
mesh_time_client_time_status(),
mesh_time_client_time_role_status(),
mesh_time_client_time_zone_status().

Time server and setup server model commands and events added:

mesh_time_server_init(),
mesh_time_server_deinit(),
mesh_time_server_get_datetime(),
mesh_time_server_get_tai_utc_delta_new(),
mesh_time_server_get_time(),
mesh_time_server_get_time_role(),
mesh_time_server_get_time_zone_offset_new(),
mesh_time_server_set_time_zone_offset_new(),
mesh_time_server_set_time(),
mesh_time_server_set_time_role(),
mesh_time_server_set_time_zone_offset_new(),
mesh_time_server_tai_utc_delta_updated(),
mesh_time_server_time_updated(),
mesh_time_server_time_role_updated(),
mesh_time_server_time_zone_offset_updated().

Added in release 1.6.0.0

BGAPI commands and events for LC and scene models have been added

LC client model commands and events:

mesh_lc_client_init(),
mesh_lc_client_get_light_onoff(),
mesh_lc_client_get_mode(),
mesh_lc_client_get_om(),
mesh_lc_client_get_property(),
mesh_lc_client_set_light_onoff(),
mesh_lc_client_set_mode(),
mesh_lc_client_set_om(),
mesh_lc_client_set_property(),
mesh_lc_client_light_onoff_status(),
mesh_lc_client_mode_status(),

mesh_lc_client_om_status(),
mesh_lc_client_property_status()

LC server model commands and events:

mesh_lc_server_init(),
mesh_lc_server_deinit(),
mesh_lc_server_init_all_properties(),
mesh_lc_server_update_light_onoff(),
mesh_lc_server_update_mode(),
mesh_lc_server_update_om(),
mesh_lc_server_ambient_lux_level_updated(),
mesh_lc_server_light_onoff_updated(),
mesh_lc_server_linear_output_updated(),
mesh_lc_server_mode_updated(),
mesh_lc_server_occupancy_updated(),
mesh_lc_server_om_updated()
mesh_lc_server_set_publish_mask()

LC setup server model commands and events:

mesh_lc_setup_server_update_property(),
mesh_lc_setup_server_set_property()

Scene client model commands and events:

mesh_scene_client_init(),
mesh_scene_client_delete(),
mesh_scene_client_get(),
mesh_scene_client_get_register(),
mesh_scene_client_recall(),
mesh_scene_client_store(),
mesh_scene_client_register_status(),
mesh_scene_client_status()

Scene server model commands and events:

mesh_scene_sever_init(),
mesh_scene_server_deinit(),
mesh_scene_server_get(),
mesh_scene_server_publish(),
mesh_scene_server_recall(),
mesh_scene_server_register_get()

Scene setup server model commands and events:

mesh_scene_setup_server_init(),
mesh_scene_setup_server_delete(),
mesh_scene_setup_server_publish(),
mesh_scene_setup_server_store()

2 Improvements

2.1 Changed APIs

Changed in release 1.6.3.0

Two BGAPI events have been changed

Changed friend events:

mesh_friendship_established(): netkey_index field added,
mesh_friendship_terminated(): lpn_address and netkey_index fields added.

Changed in release 1.6.2.0

BGAPI commands for model-specific initialization of generic client and server models have been added. These commands can be used in place of the already-existing initialization commands and may result in reduced size of the firmware image of a SoC project that uses only a subset of generic models.

Eleven generic client model commands have been added:

mesh_generic_client_init_battery(),
mesh_generic_client_init_common(),
mesh_generic_client_init_ctl(),
mesh_generic_client_init_default_transition_time(),
mesh_generic_client_init_level(),
mesh_generic_client_init_lightness(),
mesh_generic_client_init_location(),
mesh_generic_client_init_on_off(),
mesh_generic_client_init_power_level(),
mesh_generic_client_init_power_on_off(),
mesh_generic_client_init_property().

Eleven generic server model commands have been added:

mesh_generic_server_init_battery(),
mesh_generic_server_init_common(),
mesh_generic_server_init_ctl(),
mesh_generic_server_init_default_transition_time(),
mesh_generic_server_init_level(),
mesh_generic_server_init_lightness(),
mesh_generic_server_init_location(),
mesh_generic_server_init_on_off(),
mesh_generic_server_init_power_level(),
mesh_generic_server_init_power_on_off(),
mesh_generic_server_init_property().

BGAPI commands for Provisioner have been added to allow additional manipulation of the Provisioner’s device database and key refresh procedure state.

Six Provisioner commands have been added:

mesh_prov_ddb_update_netkey_index(),
mesh_prov_flush_key_refresh_state(),
mesh_prov_get_key_refresh_phase(),
mesh_prov_key_refresh_resume(),
mesh_prov_key_refresh_start_from_phase(),
mesh_prov_key_refresh_suspend(),

<u>Changed in release 1.6.1.0</u>

One new BGAPI command has been added

Added LC server model command:

mesh_lc_server:_set_regulator_interval()

<u>Changed in release 1.6.0.0</u>

Four new BGAPI commands and five new BGAPI events have been added

Added mesh node commands:

mesh_node_rssi()
mesh_node_set_beacon_reporting()
mesh_node_stop_unprov_beaconing()

Added generic server event:

mesh_generic_server_state_recall()

Added mesn node events:

mesh_node_beacon_received()
mesh_node_heartbeat()
mesh_node_neartbeat_start()
mesh_node_heartbeat_stop()

One new BGAPI errorcode has been added:

bg_err_mesh_no_data_available

2.2 Changed Documents

None

3 Fixed Issues

Fixed in release 1.6.3.0

ID #	Description
412675	Communicate locally-originated IV index updates properly over established GATT proxy connections
468395	Fixed unit conversion issues in LC Server PI regulator calculations
468761	Increased buffers in mesh_lib helper library to handle 36-octet properties
468987	Fixed LC Server lightness transitions when returning from manual override
475082	Fixed Current Scene state clearing when model state changes after scene recall
475280, 475281	Fixes of issues in LC Server lightness transitions controlled by the PI regulator
475738	Fix Provisioner's managing of its own device key
477086	Fixed Scene Recall behavior for models that are extended by LC Server model
477583	Fixed buffer handling issue in handling segmented friendship messages
479841	Fixed persistent storage handling of lightness range and color temperature range values
480444	Fixed model reallocation when firmware update adds models to the project
480507	Fixed health client initialization on a non-Provisioner node
481974	Fixed generic model-specific initialization calls
482008	Fixed BGAPI return value for mesh_generic_server_init_power_level()

Fixed in release 1.6.2.0

ID #	Description
466717	Fixed handling of extended scan response events in the stack
469269	Fixed generic level unintentional wrap around from minimum to maximum value when using move requests

Fixed in release 1.6.1.0

ID #	Description
434160	Added missing handling of segmented control messages
444559	Optimized model configuration storage management on bootup
448945	Made handling of disconnections during PB-GATT provisioning more robust
454060	Fixed Scene Server model storage allocation
455878	Fixed LC Server ambient lux level reporting
456186	Fixed handling of segmented messges once a preceding segmented message reception was cancelled
456976	Fixed buffer allocation for stack internal control messages
458876	Fixed error code given on provisioning failure due to malformed message
459204	Fixed relaying of PDUs with RFU destination addresses
461448	Fixed emitted heartbeat count when using local test interface

Fixed in release 1.6.0.0

ID #	Description
4975, 439811	GCC linking with link-time optimization disabled
270215	Added BGAPI for heartbeat monitoring in the application
417988	Added BGAPI for received secure network beacon monitoring in the application
421689	Key index and key value storing made into an atomic operation
447682	Fixed local loopback interface
449257	Fixed project conversion to C++ in Studio

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
467080	BTMESH_HEAP_SIZE macro does not take model counts into account	Manually edit the macro to handle model counts if some models are present more than once
470417	Polymorphic GATT database capability settings clash with Mesh proxy and provisioning services	
483973	**PTI traces taken with xG21 hardware are flawed**	**Capture traces using other hardware**
485543	**Current scene number is zeroed on scene recall**	

5 Deprecated Items

None

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. You can 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.

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

8 Legal

8.1 Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications.

Application examples described herein are for illustrative purposes only.

Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

8.2 Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world’s most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave and others are trademarks or registered trademarks of Silicon Labs.

ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings.

Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.