Bluetooth® mesh SDK 4.2.1.0 GA

Gecko SDK Suite 4.2
May 3, 2023

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 5.3 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.

This release includes features supported by the Bluetooth mesh specification version 1.1.

These release notes cover SDK versions:

4.2.1.0 released May 3, 2023
4.2.0.0 released March 8, 2023
4.1.0.0-prev released February 1, 2023 (limited access)
4.0.0.0-prev released December 14, 2022 (limited access)
3.0.0.0-prev released June 20, 2022 (limited access)

KEY FEATURES

- Support for mesh draft specification 1.1:
  - Mesh Protocol
  - Mesh Binary Large Object Transfer Model (MBT)
  - Mesh Device Firmware Update Model (DFU)
- Reduced project flash consumption by optimizing Mesh stack code size
- Support added for xGM240P PCB Modules and BG22/BGM220 Explorer Kits
- Support added for GCC version 10.3-2021.10 and IAR version 9.20.4

Compatibility and Use Notices

For more 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, or if you are new to the Silicon Labs Bluetooth mesh SDK, see Using This Release.

Specification Compatibility:

This release contains an implementation of the Mesh Protocol 1.1, Mesh Model 1.1, Mesh Binary Large Object Transfer, and Mesh Device Firmware Update specifications. These specifications are not yet adopted Bluetooth SIG specifications, and no Bluetooth qualification program currently exists for these specifications. Therefore, these specifications are not approved for use in commercial production. Any experimental use of the features in these specifications are "AS IS", and no Bluetooth license rights are granted.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 9.20.4

- 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 10.3-2021.10, provided with Simplicity Studio.

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

Contents

1 New Items

1.1 New Features

Added in release 4.2.1.0

New Hardware:

Support for EFR32xG21 Rev C and Rev D

Added in release 4.2.0.0

The default Bluetooth address used for advertising Mesh data and secure network beacons has been changed to be a private address that is periodically reassigned to a new random value. This is because using a public address would not work well with Mesh 1.1 privacy-enhancing features, Private Network Beacons and Private GATT Proxy.

Added in release 4.0.0-prev

Support for Bluetooth mesh over advertising extensions (AE) has been added as a proprietary extension. This feature facilitates faster transmission of long messages, compared to the standard Bluetooth Mesh using regular advertisements. For instance, device firmware update using AE can result in greatly reduced transmission times compared to standard operation. Bluetooth Mesh over AE is, however, not compliant with the Bluetooth mesh specification. Products using Mesh over AE are not fully interoperable with products that are strictly compliant with the Bluetooth mesh specification.

Support for choosing the BLE advertising BGAPI the Bluetooth mesh stack uses internally has been added. Previously, the mesh stack used internally only the old BGAPI, which prevented the use of the newer BLE advertising BGAPIs in the application, since the old and the new APIs do not work interoperably. Now the BGAPI the stack uses can be specified by component selection in the project.

Example applications:

IV update is demonstrated through the embedded examples and **Bluetooth Mesh – Host Provisioner**.

Key refresh and key export to Network Analyzer is demonstrated with **Bluetooth Mesh – Host Provisioner.**

Tooling:

NCP Commander is now able to create a new mesh Network, provision and configure nodes into that network, and send mesh packets using the Generic Onoff Client and Generic Level Client models. This allows testing nodes with Generic Level and Onoff Server models and any other models bound to these.

New Hardware:

Support for xGM240 SIP Modules

Added in release 3.0.0-prev

The support for the following have been added based on specifications that are in Validation status and are not available for qualification. In order to obtain Bluetooth SIG qualification, customers will need to upgrade to a release which will be based on the adopted specification, which will be made available after the adoption.

1. Mesh Protocol 1.1 with the following features
   - Remote Provisioning (RPR)
   - Certificate Based Provisioning (CBP)
   - Private Beacons(PRB)
2. Mesh Binary Large Object Transfer Model (MBT)
3. Mesh Device Firmware Update Model (DFU)

New Example Applications

Embedded applications:

Bluetooth Mesh – SoC DFU Distributor: Demonstrates the Firmware Distributor role based on the BT Mesh Model specification. Distributor is responsible for delivering new firmware images to the Updating nodes and monitoring the progress of the firmware update.

B**luetooth Mesh - NCP Empty 1.1**: NCP mode target application including remote provisioning support. To be used with the BT Mesh Host Provisioner example found in GSDK folder app/bluetooth/example_host/btmesh_host_provisioner

Bluetooth Mesh - SoC Empty with Certificate-Based Provisioning support: Demonstrates the bare minimum needed for a Bluetooth Mesh C application that allows Certificate-Based Provisioning (CBP). The application starts Unprovisioned Device Beaconing after boot waiting to be provisioned to a Mesh Network.

Bluetooth Mesh - SoC CSR Generator: Certificate generating firmware example. Software is generating the device EC key pair, the signing request for the device certificate, and other related data. The generated data can be read out by the Central Authority.

NCP Host applications:

Bluetooth Mesh – Host Provisioner was extended with **Remote Provisioning** and **Certifiate Based Provisioning** support.

New Components

- **BLOB Storage:** Provides an API for storing binary large objects (BLOB) via the bootloader.
- **BLOB Transfer Client:** Provides BT Mesh BLOB (binary large object) Transfer Client functionality, defined in mesh specification.
- **BLOB Transfer Server:** Provides BT Mesh BLOB (binary large object) Transfer Server functionality, defined in mesh specification.
- **DFU distributor:** Wrapper software component for DFU distributor role in Bluetooth Mesh applications.
- **Firmware Update Client**: Provides BT Mesh Firmware Update Client functionality, defined in mesh specification.
- **Firmware Update Server**: Provides BT Mesh Firmware Update Server functionality, defined in mesh specification.
- **Firmware Distribution Server:** Provides BT Mesh Firmware Distribution Server functionality, defined in mesh specification.
- **Remote Provisioning:** Used to support remote provisioning client functionality of provisioning devices into a mesh network by interacting with a mesh node that supports the Remote Provisioning Server model.
- **Remote Provisioning Server:** Used to support remote provisioning server functionality of provisioning a remote device over the mesh network and to perform the Node Provisioning Protocol Interface procedures.
- **DFU updating node:** Wrapper software component for DFU updating node role in Bluetooth Mesh applications.

New Documents

- QSG183: Bluetooth Mesh SDK Quick-Start Guide for SDK v4.x
- AN1319: Bluetooth Mesh Device Firmware Update
- AN1370: Bluetooth Mesh Device Firmware Update Example Walkthrough
- AN1368: Bluetooth Mesh Remote Provisioning
- AN1405: Bluetooth Mesh on Advertising Extensions

Added in release 3.0.0.0

New Development Tools

More user-friendly filtering for Software Examples in Simplicity Studio.

New Hardware Support

Support was added for xGM240P PCB Modules and BG22/BGM220 Explorer Kits.

1.2   New APIs

Added in release 4.2.0.0

Test API has been ameded with a new call, **sl_btmesh_test_send_private_beacons,** that can be used for sending private network beacons for testing purposes.

Test API has been amended with a new call, **sl_btmesh_test_adv_use_random_address**, that can be used for setting the type of Bluetooth address used for Mesh advertising.

Added in release 4.1.0-prev

Test API has been ameded with a new call, **sl_btmesh_test_update_key_refresh_phase** that can be used for testing various key refresh phases locally without a configuration client.

Added in release 4.0.0-prev

Bluetooth mesh over AE adds client and server vendor models using Silicon Labs vendor messages for over-the-air configuration of the feature, as well as the corresponding BGAPI classes **sl_btmesh_silabs_config_client** and **sl_btmesh_silabs_config_server**.

Node API has been amended with multiple additions:

- APIs for saving the proxy solicitation RPL state, and for examining whether saving the RPL is needed, have been added (**sl_btmesh_node_save_proxy_solicitation_rp**l, and **sl_btmesh_node_get_proxy_solicitation_rpl_status**).
- APIs for reading and writing unprovisioned device URI data to persistent storage have been added (**sl_btmesh_node_set_oob_uri**, and **sl_btmesh_node_get_oob_uri**).
- An API for locally accessing the Models Metadata page, **sl_btmesh_node_get_local_model_metadata_page**, and the corresponding events, **sl_btmesh_node_local_model_metadata_page** and **sl_btmesh_node_local_model_metadata_page_end**.

Provisioner's control over the key refresh procedure has been improved by additional APIs, **sl_btmesh_prov_set_key_refresh_failure**, **sl_btmesh_prov_phase_timeout_get**, and **sl_btmesh_prov_phase_timeout_set**. Also, the Provisioner can now specify the amount of parallel key refresh configuration requests by setting the configuration variable **SL_BTMESH_CONFIG_LIMIT_PROV_CONCURRENT_KR**. Note that the value should be less than or equal to the overall number of parallel configuration client requests.

Configuration client's control over configuration requests has been improved by an additional API **sl_btmesh_config_client_set_request_timeout_for_node**, and an event **sl_btmesh_config_client_obo_ack_received**, which is generated when a friend node acknowledges a transport message that has an LPN as the destination.

Health client model API has been amended with explicit initialization and deinitialization, so that health client model code will be pulled into a project only when it is needed: **sl_btmesh_health_client_init** and **sl_btmesh_health_client_deinit**.

Test command for adjusting PB-ADV timing paramerers has been provided as **sl_btmesh_test_set_adv_provisioning_bearer_timing**. Note that the default timing values are those that the specification defines; timings should be adjusted only for debugging purposes.

Added in release 3.0.0-prev

Multiple new BGAPI classes have been added to support the Mesh 1.1 specification release new features. For the details of the added classes please see the API reference; the additions are summarized below.

Support for Mesh binary large object transfer adds BGAPI classes for MBT client model and MBT server model.

Support for Mesh device firmware update adds BGAPI classes for firmware update client model, firmware update server model, firmware distribution client model, firmware distribution server model, and firmware standalone updater model, as well as a supporting BGAPI class for device firmware image cache manipulation.

Support for Mesh 1.1 remote provisioning adds BGAPI classes for remote provisioning client model and remote provisioning server model.

Support for Mesh 1.1 private beacons adds BGAPI classes for private beacon client model and private beacon server model.

Support for Mesh 1.1 minor enhancements adds BGAPI classes for SAR configuration client model, SAR configuration server model, large composition data client model, large composition data server model, on-demand private proxy client model, on-demand private proxy server model, and solicitation PDU RPL configuration client model.

Furthermore, existing BGAPI classes have been amended as listed below.

Mesh 1.1 enhanced provisioning algorithm support adds one command to node BGAPI, **sl_btmesh_node_set_provisioning_algorithm()**, and defines flag enumeration values for both Mesh 1.0 and 1.1 provisioning algorithms.

Mesh 1.1 certificate-based provisioning support adds one command to node BGAPI, **sl_btmesh_node_init_provisioning_records()**, and three commands as well as two events to provisioner BGAPI: **sl_btmesh_prov_init_provisioning_records()**, **sl_btmesh_prov_get_provisioning_records_list()**, **sl_btmesh_prov_get_provisioning_record_data()**, **sl_btmesh_provisioning_records_list()**, and **sl_btmesh_provisioning_record_data()**.

Support for the Node Provisioning Protocol Interface procedures adds two events to the node BGAPI, **sl_btmesh_node_address_updated()** and **sl_btmesh_node_dcd_updated().**

Proxy solicitation support adds one new BGAPI command to the proxy BGAPI: **sl_btmesh_proxy_send_solicitation()**.

New BGAPI commands have been added to the test BGAPI to support private beacon testing: **sl_btmesh_test_get_private_identity()** and **sl_btmesh_test_set_private_identity()**.

2 Improvements

The supported complier versions have been updated. GCC version 10.3-2021.10 and IAR version 9.20.4 are now supported.

The flash footprint of the mesh stack implementation has been reduced by optimization of structures and removal of unnecessary dependencies between components. Exact reduction depends on the features used by the project.

3   Fixed Issues

Fixed in release 4.2.1.0

ID #	Description
1130439	Corrected memory leak of OOB data when canceling a DFU distribution
1132170	Corrected an error in Device UUID generation that could lead to a crash on bootup in an unlikely scenario.

Fixed in release 4.2.0.0

ID #	Description
858513, 1086009	Corrected HSL transition timing issue
1085641	Corrected authentication data truncation to 128 bits when 256 bit data is stored on device and Mesh 1.0 provisioning is in use.
1086255	Fixed status message generation when firmware distribution is cancelled.
1095921	Fixed GATT advertising with Private Node Identity.
1097975	Fixed Private Network Beacon generation.
1098624	Use non-public address for Mesh advertisement bearer so that Mesh 1.1 privacy features are not undone.
1102663	Fixed MBT block size calculation.
1106639	Fixed Private GATT Proxy Solicitation handling.
1106642	Fixed remote provisioning message endianness handling issue.
1107501	Use up to date ATT MTU with GATT Proxy communications.

Fixed in release 4.1.0-prev

ID #	Description
1072206	Added a test command for local key refresh phase update, as the command to update key locally moves the KR phase for that key to phase 1. Note that this was mislabeled as '1072236' in the 4.0.0-prev release notes.
1078021, 1084904	Multiple corrections to remote provisioning protocol implementation.
1078517	Corrected friend functionality initialization when device is unprovisioned.
1079777, 1081836, 1084062, 1084907, 1085645	Mulitple corrections to MBT protocol implementation.
1080846	Corrected Hue-related Generic Level wrap-around behavior.
1085578	Corrected a regression to node identity advertisement.
1085686	Corrected reporting of security updates (IV index update and key refresh) from Friend to LPN during a friendship.

Fixed in release 4.0.0-prev

ID #	Description
757936	Fixed an issue with missing generic and lighting model statuses when the request TID was equal to an already processed TID.
774974	Fixed remote provisioning issue for extended scan in device mode.
822120	Fixed missing timeout for identity and private identity beaconing.
833418	Fixed generic level initial value for levels bound to other models.
843168	Fixed an overly restrictive check for ATT MTU when provisioning.
845510	Fixed an issue with Distributor not being able to distribute images while having an active friendship.
845522	Fixed an issue with Distributor not being able to start a transfer when verification of firmware fails.
853178	Fixed a potential out-of-memory issue by correcting a component inclusion to be conditional on use.
854491	Fixed a timeout problem with upload.

ID #	Description
1013958, 1022225	Allowed the inclusion of either old or new advertisement BGAPI into the Mesh stack.
1031411	Fixed handling of errors when provisioning link is not yet open.
1061551	Fixed health server publication handling of TTL=1.
1062373	Fixed a regression in GATT proxy service advertisements.
1064324	Fixed a regression with Series 1 Provisioner's factory reset.
1074421	Fixed an issue with starting a new upload when previous one failed.
00293822, 00262939, 00261494, 00258424	Examples demonstrating IV Updates in order to avoid sequence number exhaustion.

Fixed in release 3.0.2.0

ID #	Description
465318	Fixed the issue with periodic publishing and publish retransmissions not working simultaneously.
1015385	Fixed GATT proxy server advertisement restart after proxy client disconnection.
1017565, 650825	Publish retransmissions issue fixed for generic models and lighting models, as well as Time Server model.
1024154	Perform a full reset of Series 2 devices when Config Reset message is processed and the device is reset.
1024849	Fixed an issue with Scheduler actions not triggering after a power cycle.
1024851	Fixed an issue with Scheduler repeated events being one hour late.
1032627	Discard broken advertisement indications before attempting to decrypt instead of after.

Fixed in release 3.0.1.0

ID #	Description
818000	Corrected an issue with Light LC model PTS tests where reported lightness values were off by a small amount.
844593, 846010, 846598, 849377	Increased call stack of all applications to avoid running out of call stack when GATT proxy or GATT provisioning is in use.

Fixed in release 3.0.0.0

ID #	Description
764197	Set the default friend queue size to be a power of two.
818395	Fixed a potential crash in a situation where persistent storage contains more data than the project is configured to handle (e.g., after a firmware update without a factory reset).
818523	Corrected the check that the key used for publication is bound to the model to include virtual address publication as well.
831921	Fixed a regression with cleaning up the provisioning session after link failure.
833535	Updated the list of property IDs to contain the full current list of properties.

4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
401550	No BGAPI event for segmented message handling failure.	Application needs to deduce failure from timeout / lack of application layer response; for vendor models an API has been provided.
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.	
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.
841360, 1102630	Poor performance of segmented message transmission over GATT bearer.	Ensure that the underlying BLE connection's Connection interval is short; ensure that ATT MTU is large enough to fit a full Mesh PDU; tune the minimum connection event length to allow multiple LL packets to be transmitted per connection event.

5 Deprecated Items

None

6   Removed Items

Removed in release 3.0.0.0

The deprecated BGAPI command **sl_btmesh_node_erase_mesh_nvm()** has been removed. Use **sl_btmesh_node_reset()** instead.

7 Using This Release

This release contains the following

- Silicon Labs Bluetooth mesh stack library
- Bluetooth mesh sample applications

If you are a first time user, see *QSG176: Silicon Labs Bluetooth Mesh SDK v2.x Quick-Start Guide*.

7.1 Installation and Use

The Bluetooth mesh 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/Sili-conLabs/gecko_sdk for more information.

The GSDK default install location has changed with Simplicity Studio 5.3 and higher.

- 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

This version of the stack is integrated with Secure Vault Key Management. When deployed to Secure Vault High devices, mesh encryption keys are protected using the Secure Vault Key Management functionality. The table below shows the protected keys and their storage protection characteristics.

Key	Exportability on a node	Exportability on Provisioner	Notes
Network key	Exportable	Exportable	Derivations of the network key exist only in RAM while network keys are stored on flash
Application key	Non-exportable	Exportable	
Device key	Non-exportable	Exportable	In Provisioner's case, applied to Provisionerr's own device key as well as other devices' keys

Keys that are marked as "Non-Exportable" can be used but cannot be viewed or shared at runtime.

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 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.

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

www.silabs.com