Bluetooth® mesh SDK 2.1.1.0 GA
Gecko SDK Suite 3.2
July 21, 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 5.2 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:

2.1.1.0 released July 21, 2021
2.1.0.0 released June 16, 2021

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.

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 10.2.0, provided with Simplicity Studio.

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

1. **New Items** .................................................................................................................................................................................. 2  
   1.1 **New Features** ...................................................................................................................................................................... 2  
   1.2 **New APIs** ............................................................................................................................................................................ 2  
2. **Improvements** ............................................................................................................................................................................. 3  
   2.1 **Changed APIs** ..................................................................................................................................................................... 3  
3. **Fixed Issues** .............................................................................................................................................................................. 4  
4. **Known Issues in the Current Release** .......................................................................................................................................... 5  
5. **Deprecated Items** ....................................................................................................................................................................... 6  
6. **Removed Items** .......................................................................................................................................................................... 7  
7. **Using This Release** .................................................................................................................................................................... 8  
   7.1 **Installation and Use** ............................................................................................................................................................. 8  
   7.2 **Security Information** ............................................................................................................................................................ 8  
   7.3 **Support** ............................................................................................................................................................................... 9
1 New Items

1.1 New Features

**Added in release 2.1.0.0**

**Secure Vault Integration**

Beginning with release 2.1.0.0, the Bluetooth Mesh SDK uses the Secure Vault Key Management functionality for storing mesh cryptographic keys when Secure Vault High devices are used.

The Secure Vault integration is visible to the customer in a number of ways on Series 2 devices:

- The layout of NVM3 data for cryptographic keys and their related metadata changes. Key migration functionality is provided for projects that have been created using SDK versions 2.0 or earlier. A one-time key migration needs to be done when the firmware on a device is updated.

- Key data visibility is intentionally limited on regular mesh nodes. An application on a regular mesh node is not permitted to view application or device key data using the `sl_btmesh_node_get_key()` BGAPI command, while an application on an embedded Provisioner node is allowed to do so.

For further information on key storage in Secure Vault please refer to AN1271: Secure Key Storage.

**Compiler Support**

Supported compilers have been updated to GCC version 10.2.0 and IAR version 8.50.9.

**New Example Applications**

HSL Lighting example (Bluetooth Mesh - SoC HSL Light) was added to demonstrate a light node that is controllable over HSL server models.

IOP demos (Bluetooth Mesh - IOP Test - *) were added for the Radio boards in the Pro Development Kits (SLWRB4104A, SLWRB4181A, SLWRB4181B, SLWRB4182A). The demos allow testing interoperability with mobile phones. The test requires four examples, each example representing one of the Mesh features: proxy, relay, friend, and LPN.

**New Components**

- HSL Server Component was added.
- Support for dynamic GATT database (a Bluetooth LE feature) was added.

1.2 New APIs

**Added in release 2.1.1.0**

To optimize buffer usage with Scene models, an optional API to enable compacted scene recall events has been added (ref. issue ID 706555). It is recommended to use the new API when a node has a large amount of models, or the amount of network traffic the node is expected to hear is high.

To activate the new API, use the BGAPI command `sl_btmesh_scene_server_enable_compact_recall_events()`. Afterwards, `sl_btmesh_evt_scene_server_compact_recall_events` will signal scene recall requests. To retrieve the cached model states after a scene recall request, use the command `sl_btmesh_generic_server_get_cached_state()`.

**Added in release 2.1.0.0**

Because of Secure Vault integration, the details of storing encryption keys and their related metadata has changed on Series 2 devices. A new BGAPI class for migrating encryption keys and the embedded Provisioner’s device database after a firmware update on Series 2 devices has therefore been added. It has the following commands:

- `sl_btmesh_migration_migrate_keys`
- `sl_btmesh_migration_migrate_ddb`
2 Improvements

2.1 Changed APIs

**Changed in release 2.1.1.0**

The event `sl_btmesh_evt_friend_friendship_terminated` will now be generated when a Configuration Client disables the node’s Friend feature when a friendship is active. Previously the termination of the friendship in this situation was implicitly signalled by the `sl_btmesh_evt_node_config_set` event. (Ref. issue ID 627811)

**Changed in release 2.1.0.0**

The following BGAPI commands in the prov class now return after parameter validation, and the actual requested operation takes place after a BGAPI response is given. The completion of the requested operation is signalled by a corresponding BGAPI event:

- `sl_btmesh_prov_add_ddb_entry()` – completion of addition is signalled by `sl_btmesh_evt_prov_add_ddb_entry_complete`
- `sl_btmesh_prov_delete_ddb_entry()` – completion of deletion is signalled by `sl_btmesh_evt_prov_delete_ddb_entry_complete`

The following BGAPI command in the prov class has an additional event that may be generated after it is called:

`sl_btmesh_prov_init` – in addition to `sl_btmesh_evt_prov_initialized`, `sl_btmesh_evt_prov_initialization_failed` may be generated.

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()`
### 3 Fixed Issues

**Fixed in release 2.1.1.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>692961</td>
<td>Fixed the node becoming unresponsive when relay retransmissions were enabled when under heavy load</td>
</tr>
<tr>
<td>713152</td>
<td>Fixed a problem where limited precision of calculation caused rounding errors in the binding between Light Lightness Actual and Light Lightness Linear</td>
</tr>
</tbody>
</table>

**Fixed in release 2.1.0.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3878</td>
<td>Application should ignore GATT events for Mesh characteristics</td>
</tr>
<tr>
<td>342521</td>
<td>Math library does not grow the size of the image unnecessarily</td>
</tr>
<tr>
<td>358019</td>
<td>Corrected result code given when model publication with friendship credentials is requested but friendship is not supported</td>
</tr>
<tr>
<td>404070</td>
<td>Corrected result code given when provisioner command to create a network key is called on a non-provisioner device</td>
</tr>
<tr>
<td>454332</td>
<td>The LE GAP API should be used for device local name advertising</td>
</tr>
<tr>
<td>464907</td>
<td>Removed unnecessary ‘heartbeat started’ BGAPI event when configuration client disables heartbeat on a node</td>
</tr>
<tr>
<td>653405</td>
<td>Out-of-the-box switch sample application current consumption is now at the expected level</td>
</tr>
<tr>
<td>654477</td>
<td>DCD correctly decoded by Network Analyzer</td>
</tr>
<tr>
<td>660048</td>
<td>Button Press UC component doesn’t require IO Stream component unnecessarily</td>
</tr>
<tr>
<td>687105</td>
<td>BT Mesh Commands work with NCP Target example and NCP Commander</td>
</tr>
<tr>
<td>690803</td>
<td>Fixed duplicate vendor model IDs in code generator</td>
</tr>
<tr>
<td>690862</td>
<td>SoC empty example now starts beaconing on xG22 hardware</td>
</tr>
<tr>
<td>707497</td>
<td>Corrected PSA cryptographic context allocation</td>
</tr>
<tr>
<td>707524</td>
<td>Fixed a regression with IV recovery guard timer, not allowing another recovery too soon</td>
</tr>
<tr>
<td>710381</td>
<td>Fixed lighting default state handling when a non-default range for the corresponding model was set</td>
</tr>
<tr>
<td>711359</td>
<td>Fixed parameter checking for provisioning session creation BGAPI call</td>
</tr>
</tbody>
</table>
## 4 Known Issues in the Current Release

Issues in bold were added since the previous release.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>401550</td>
<td>No BGAPI event for segmented message handling failure</td>
<td>Application needs to deduce failure from timeout / lack of application layer response</td>
</tr>
<tr>
<td>418636</td>
<td>Issues with mesh_test local configuration state API (node identity, relay, network retransmission)</td>
<td></td>
</tr>
<tr>
<td>454059</td>
<td>A large number of key refresh state change events are generated at the end of KR process, and that may flood NCP queue</td>
<td>Increase NCP queue length in the project</td>
</tr>
<tr>
<td>454061</td>
<td>Slight performance degradation compared to 1.5 in round-trip latency tests was observed</td>
<td></td>
</tr>
<tr>
<td>624514</td>
<td>Issue with re-establishing connectable advertising if all connections have been active and GATT proxy is in use</td>
<td>Allocate one more connection than is needed</td>
</tr>
<tr>
<td>650825</td>
<td>Issue with retransmissions when a model is publishing periodically</td>
<td>Set up retransmissions in the model state and trigger periodic publishing by an application timer</td>
</tr>
<tr>
<td>676798</td>
<td>Automatic LPN polling may miss friendship termination deadline due to oscillator inaccuracy</td>
<td>Ensure application timer-based explicit polling takes place before friendship terminates, if the timeout for friendship termination is long</td>
</tr>
<tr>
<td>683518</td>
<td>Friendship termination event is not immediately seen when a friendship clear message is received</td>
<td>The event will be raised once friendship timeout expires</td>
</tr>
</tbody>
</table>
5 Deprecated Items

The following BGAPI command in the node class has been deprecated:

\texttt{sl_btmesh_node_erase_mesh_nvm()} – use \texttt{sl_btmesh_node_reset()} instead.
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 QSG176: Silicon Labs Bluetooth Mesh SDK v2.x Quick-Start Guide.

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 QSG176: Silicon Labs Bluetooth Mesh SDK v2.x Quick-Start Guide.

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 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.

<table>
<thead>
<tr>
<th>Key</th>
<th>Exportability on a node</th>
<th>Exportability on Provisioner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network key</td>
<td>Exportable</td>
<td>Exportable</td>
<td>Derivations of the network key exist only in RAM while network keys are stored on flash</td>
</tr>
<tr>
<td>Application key</td>
<td>Non-exportable</td>
<td>Exportable</td>
<td></td>
</tr>
<tr>
<td>Device key</td>
<td>Non-exportable</td>
<td>Exportable</td>
<td>In Provisioner’s case, applied to Provisioner’s own device key as well as other devices’ keys</td>
</tr>
</tbody>
</table>

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

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®, 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.