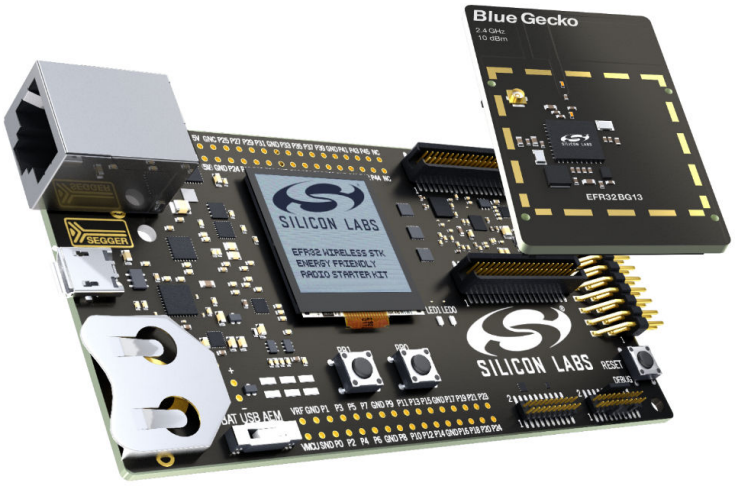
QSG148: Getting Started with the Silicon Labs *Bluetooth*® Mesh Lighting Demonstration in SDK 1.x

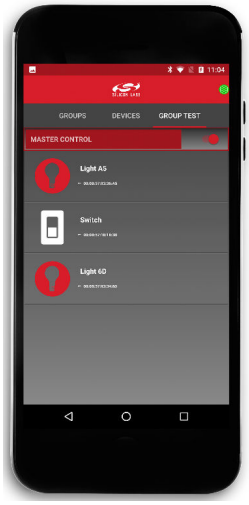
This document provides step-by-step instructions to demonstrate a basic Bluetooth mesh network. In this demo, three Wireless Starter Kit (WSTK)-based devices are provisioned as two Lights and one Switch. The mobile application allows the control of either the group of Lights or an individual Light. By pressing buttons on the Switch device, you can control the ON/OFF states and brightness for all lights in the same group. The demo is open-sourced and provides a good demonstration of a basic Bluetooth mesh network.

KEY POINTS

- Prerequisite for the demo
- Hardware set-up of WTSKs
- Bluetooth mesh SDK installation in Simplicity Studio
- Demo firmware installation
- Instructions for provisioning, configuring, and controlling network nodes using the Android smartphone application

The Bluetooth Mesh mobile app is intended to demonstrate the Silicon Labs Bluetooth Mesh technology together with the Bluetooth Mesh SDK sample apps. The mobile app is a reference app for the Bluetooth Mesh mobile ADK but it should not be taken as a starting point for customers to create their own mobile apps. For guidance on creating mobile apps with the Bluetooth Mesh mobile ADK, refer to AN1200: Bluetooth® Mesh for iOS and Android ADK.





1. Prerequisites

The Silicon Labs Bluetooth mesh lighting demonstration is designed to illustrate Bluetooth mesh operation without any need to configure or compile software. To get started with the Bluetooth mesh demo, obtain the following.

1.1 Order Development Kits

The Blue Gecko Bluetooth SoC Wireless Starter Kit is the easiest and fastest way to start the evaluation and development of your own Bluetooth mesh applications. To get started with the Bluetooth mesh demo, you need to have **three (3)** EFR32™ WSTK main boards and radio boards. These can be obtained by ordering any of the Wireless Starter Kit options below.

Option 1: QTY(3) of PN: SLWSTK6020B kits: www.silabs.com/products/development-tools/wireless/bluetooth/blue-gecko-bluetooth-low-energy-soc-starter-kit

Option 2: QTY(1) of PN: SLWSTK6000B kit: www.silabs.com/products/development-tools/wireless/mesh-networking/mighty-gecko-starter-kit

Option 3: QTY(1) of PN: SLWSTK6006A kit: www.silabs.com/products/development-tools/wireless/efr32xg21-wireless-starter-kit

This demo requires either **EFR32MG21**, **EFR32BG13**, **EFR32MG13**, **EFR32BG12**, or **EFR32MG12** radio boards. If you already have the WSTK Main Boards, you can purchase the required radio boards here.

Note: This document references the boards provided in PN: SLWSTK6020B. The radio board provided in SLWSTK6000B and SLWSTK6006A as well as the radio board mentioned above can be substituted for the EFR32BG13 board referenced in this document.

1.2 Download Simplicity Studio

Go to: http://www.silabs.com/simplicity-studio to download the latest Simplicity Studio version compatible with your computer's operating system.

1.3 Download Bluetooth Mesh by Silicon Labs Mobile App from iTunes or Google Play

iTunes:

https://itunes.apple.com/us/app/bluetooth-mesh-by-silicon-labs/id1411352948?mt=8

Google Play:

https://play.google.com/store/apps/details?id=com.siliconlabs.bluetoothmesh&hl=en

Note: The minimum requirement for the smartphone is Android 6 (API23).

1.4 Obtaining Support

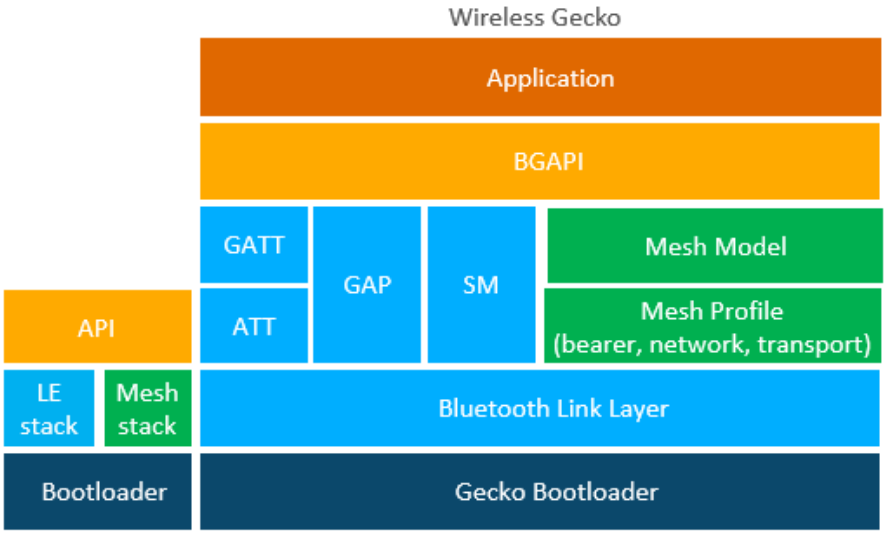
You can access the Silicon Labs support portal at https://www.silabs.com/support through Simplicity Studio Resources. Click the "Email-Support" link and log in with your self-registered credentials. Use the support portal to contact Customer Support for any questions you might have about the demonstration.

2. About the Bluetooth Mesh SDK

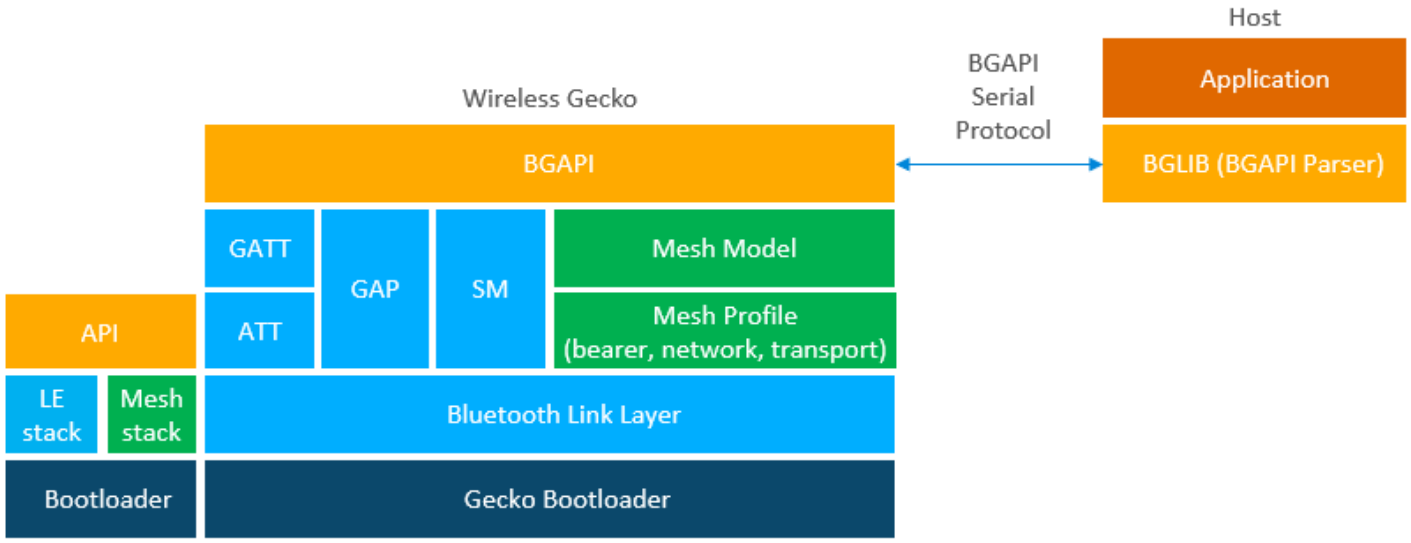
The Silicon Labs Bluetooth mesh stack is an advanced Bluetooth mesh protocol stack implementing the Bluetooth mesh standard. It can run alongside the Bluetooth Low Energy (LE) stack, using a common link layer, which allows using LE features in parallel. The Silicon Labs Bluetooth mesh stack is meant for Silicon labs Wireless Gecko SoCs and modules.

The Silicon Labs Bluetooth mesh stack provides multiple APIs for the developer to access the Bluetooth mesh functionality. Two modes are supported.

1. Standalone mode (also referenced as SoC mode), where both the Bluetooth mesh stack and the application run in a Wireless Gecko SoC or module. The application can be developed with the C programming language.



2. Network Co-Processor (NCP) mode, where the Bluetooth stack runs in a Wireless Gecko and the application runs on a separate host MCU. For this use case, the Bluetooth stack can be configured into NCP mode where the API is exposed over a serial interface such as UART.



2.1 Bluetooth Mesh Stack Features

The features of the Silicon Labs Bluetooth stack are listed in the following table. For details on the features of the Bluetooth **Low Energy** stack, refer to QSG139: Getting Started with *Bluetooth*® Software Development.

Table 2.1. Bluetooth Mesh Stack Features

Feature	Value and Comment
Bluetooth mesh version	Bluetooth mesh 1.0
Node types	Relay, Proxy, Friend, and Low Power Node (LPN)
Provisioning bearers	PB-ADV PB-GATT
GATT services	Proxy Provisioning
Security	ECDH AES-128 encryption, authentication, and obfuscation OoB authentication Replay protection Key refresh (blacklist)
Host (NCP) interfaces	4-wire UART with RTS/CTS control or 2-wire UART without RTS/CTSGPIOs for sleep and wake-up management Secure NCP option for data encryption between NCP target and host
Wi-Fi Coexistence	Using Packet Trace Arbitration (PTA)
Bootloaders	Secure Gecko Bootloader supporting authenticated and encrypted updates over OTA (over GATT) or UART and Secure Boot. The Gecko Bootloader also supports flash partitioning and both internal and external (SPI) flash.
Non-volatile memory	EFR32[B|M]G12, EFR32[B|M]G13: NVM3 or Persistent Store (PS). (Note: Example applications in the SDK use NVM3 by default.) EFR32[B|M]G21: NVM3

Table 2.2. Supported Models

Model	SIG Model ID	Example App
Model Group: NA		
Vendor	N/A	N/A
Model Group: Generic		
Generic OnOff Server Generic OnOff Client	0x1000 0x1001	SOC - BT Mesh Light SOC - BT Mesh Switch
Generic Level Server Generic Level Client	0x1002 0x1003	SOC - BT Mesh Light N/A
Generic Default Transition Time Server Generic Default Transition Time Client	0x1004 0x1005	SOC - BT Mesh Light N/A
Generic Power OnOff Server Generic Power OnOff Setup Server Generic Power OnOff Client	0x1006 0x1007 0x1008	SOC - BT Mesh Light SOC - BT Mesh Light N/A
Generic Power Level Server Generic Power Level Setup Server Generic Power Level Setup Client	0x1009 0x100A 0x100B	N/A N/A N/A

Model	SIG Model ID	Example App
Generic Battery Server Generic Battery Client	0x100C 0x100D	N/A N/A
Generic Location Server Generic Location Setup Server Generic Location Client	0x100E 0x100F 0x1010	N/A N/A N/A
Generic Admin Property Server	0x1011	N/A
Generic Manufacturer Property Server	0x1012	N/A
Generic User Property Server	0x1013	N/A
Generic Client Property Server	0x1014	N/A
Generic Property Client	0x1015	N/A
Model Group: Sensors		
Sensor Server Sensor Setup Server Sensor Client	0x1100 0x1101 0x1102	SOC - BT Mesh Sensor Server N/A SOC - BT Mesh Sensor Client
Model Group: Lighting		
Light Lightness Server Light Lightness Setup Server Light Lightness Client	0x1300 0x1301 0x1302	SOC - BT Mesh Light SOC - BT Mesh Light SOC - BT Mesh Switch
Light CTL Server Light CTL Setup Server Light CTL Temperature Server Light CTL Client	0x1303 0x1304 0x1306 0x1305	SOC - BT Mesh Light SOC - BT Mesh Light SOC - BT Mesh Light SOC - BT Mesh Light
Light LC Server Light LC Setup Server Light LC Client	0x130F 0x1310 0x1311	SOC - BT Mesh Light SOC - BT Mesh Light N/A
Time and Scenes		
Scene Server Scene Setup Server Scene Client	0x1203 0x1204 0x1205	SOC - BT Mesh Light SOC - BT Mesh Light SOC - BT Mesh Switch

2.2 Bluetooth Mesh Stack Limitations

Component	Feature	Value and Comment
Mesh Node (EFR32)	Network Keys on a node(1)	Maximum of 7
	Application Keys on a node	Maximum of 8
	Number of nodes that can be communicated with	Maximum of 4096 (depending on available RAM and NVM3)
	Concurrent segmented messages being received	Maximum of 255 (depending on available RAM)
	Concurrent segmented messages being sent	Maximum of 255 (depending on available RAM)
	Parallel provisioning sessions	Maximum of 1
	Faults reported on the health server	Maximum of 5
Mesh Provisioner (EFR32)	Maximum number of supported nodes	512
	Maximum number of network keys per node	Maximum of 7
	Maximum number of application keys per node	Maximum of 8
	Replay protection list size	Maximum of 4096 (depending on available RAM and NVM3. Network size limit is still 512)
	Parallel provisioning sessoins	1
	Concurrent key refresh operations	Maximum of 16
Mesh Provisioner (ADK)	Replay protection list size (max network node count)	32768
	Maximum number of network keys per node	Maximum of 7
	Maximum number of application keys per node	Maximum of 8
	Parallel provisioning sessions	1

(1) The node belongs to a single network but the network may have multiple network keys to encrypt the traffic.

3. Getting Started

3.1 Preparing the WSTK

The layout of the Wireless Starter Kit (WSTK) Main Board with attached EFR32BG13 radio board is shown in the following figure:

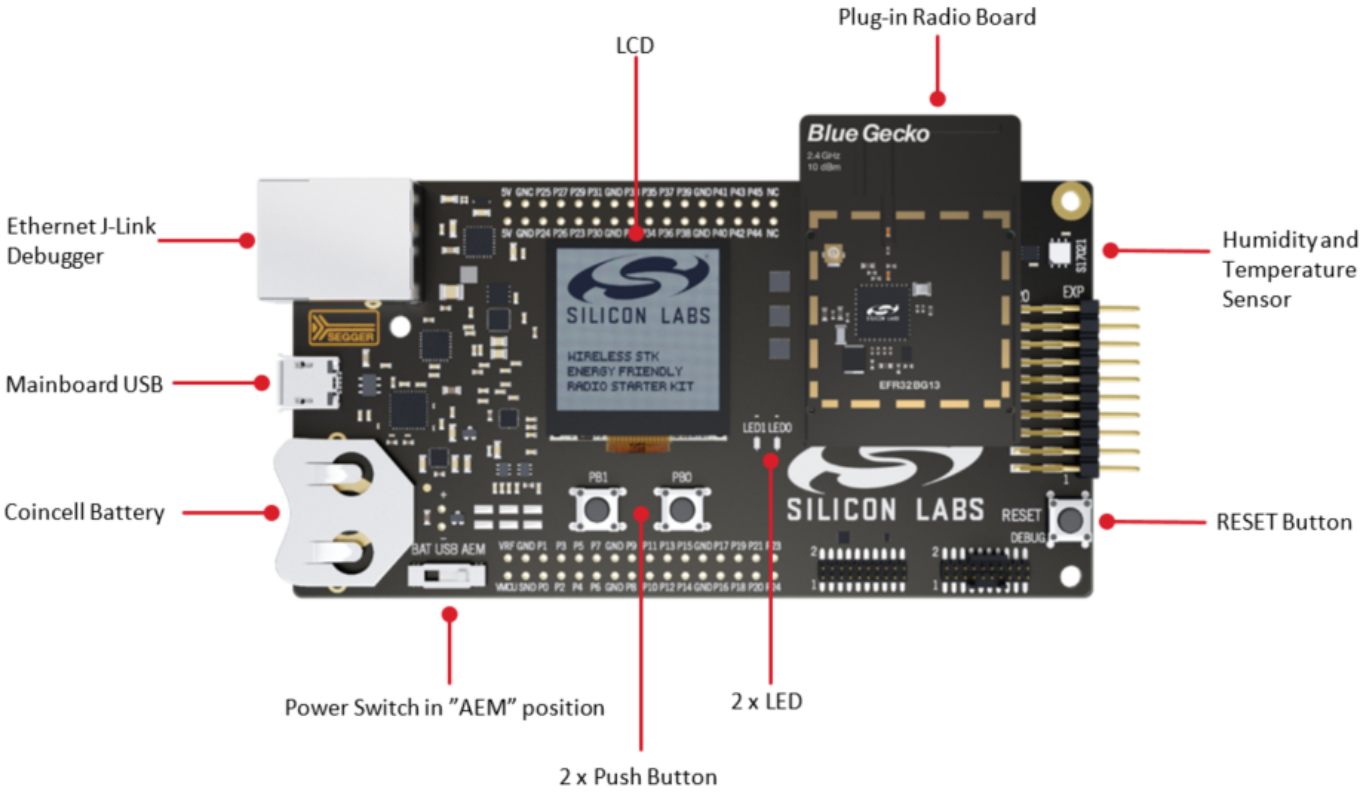


Figure 3.1. WSTK Main Board with Radio Board Attached

1. Connect a Blue Gecko Radio Board to the WSTK Main Board.

   Use radio board SLWRB4104A **EFR32BG13** 2.4 GHz (+10 dBm) for this demo experience.
2. Connect the WSTK to a PC using the "J-Link USB" connector and the cable provided with the starter kit.
3. If not already set, turn the Power switch to "**AEM**" position.
4. Repeat the above steps for the other two kits so all three kits are connected to your computer.

Verifying the Setup:

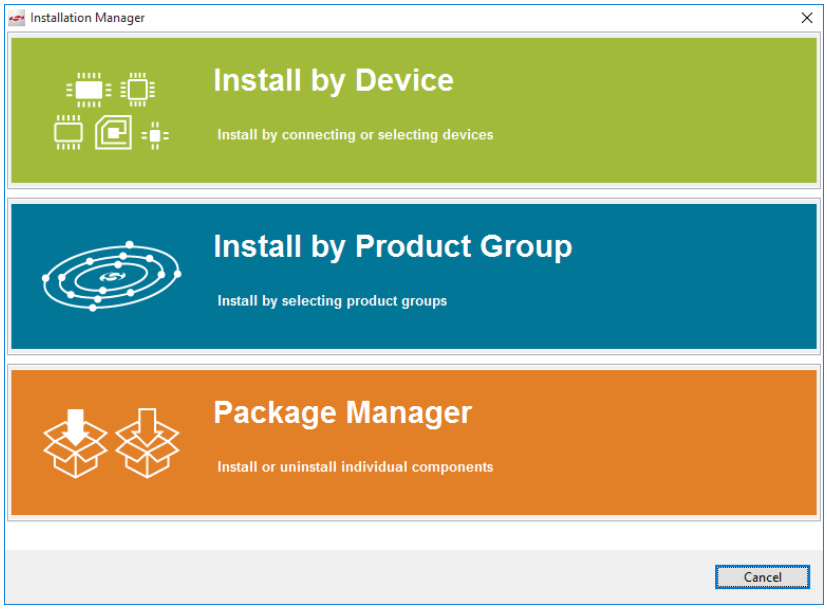
1. Check that the blue "USB Connection Indicator" LED (next to “J-Link USB”) turns on or starts blinking.
2. Check that the Main Board LCD display turns on and displays a Silicon Labs logo.

For more detailed information regarding the Starter Kit, refer to UG279: EFR32BG13 Blue Gecko Bluetooth Starter Kit User's Guide.

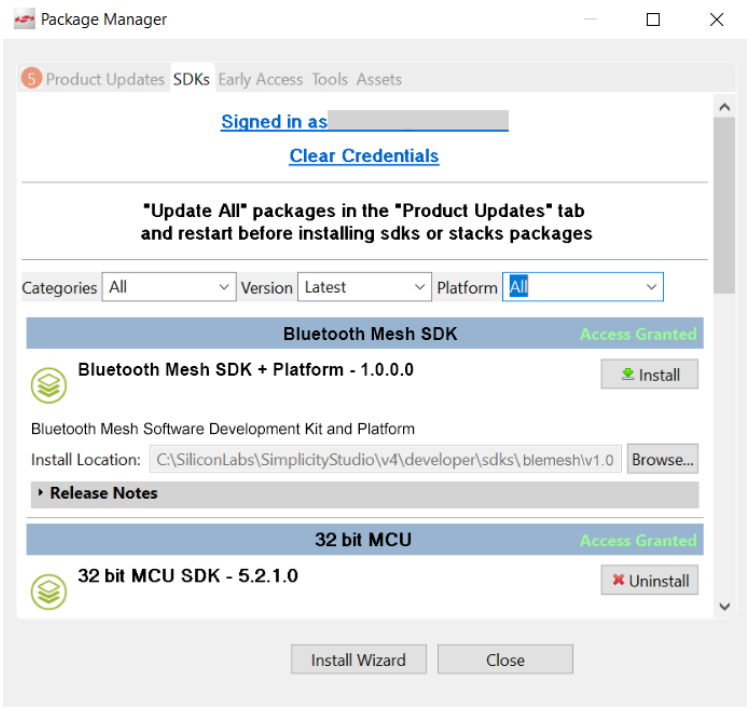
3.2 Open Simplicity Studio and Install Bluetooth Mesh SDK

Bluetooth mesh SDK is installed using the Simplicity Studio package manager.

1. Open Simplicity Studio and log in using your Silicon Labs account.
2. Click the Download Update icon (red/green down arrow under the menu bars), and click Package Manager.



3. Go to the SDKs tab to install Bluetooth mesh SDK.



4. In the Launcher screen, check if the preferred SDK is “Bluetooth mesh SDK + Platform”. If not, click the link provided to change the preferred SDK to “Bluetooth mesh SDK + Platform”.

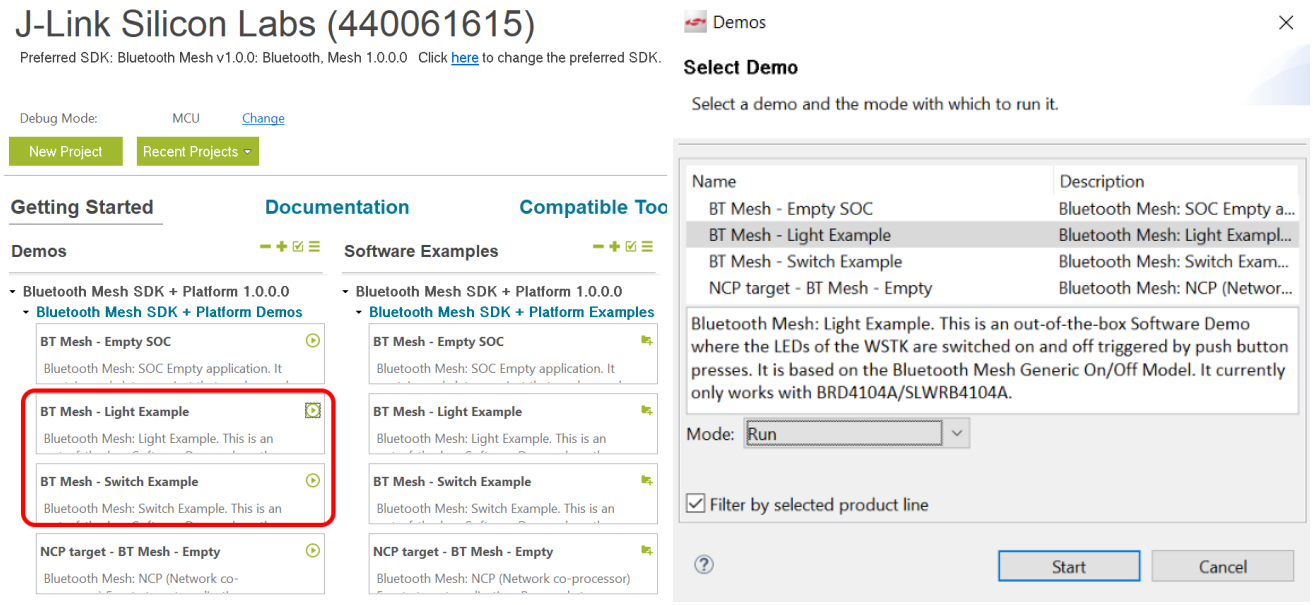
You can find more detailed instructions for Simplicity Studio in QSG139: Bluetooth Development with Simplicity Studio.

3.3 Install the Demonstration Firmware

When the devices are connected to your PC with a USB cable, you can see three devices listed in the **Device** window in Simplicity Studio. Select the J-link for a device to display demonstrations, examples, and documentation associated with the Bluetooth Mesh SDK.

For this demo, you need to flash two devices with **BT Mesh – Light Example** and one device with **BT Mesh – Switch Example**.

To install the firmware, click the demo. In the **Mode** drop-down in the next dialog, select **Run**. Click [**Start**].



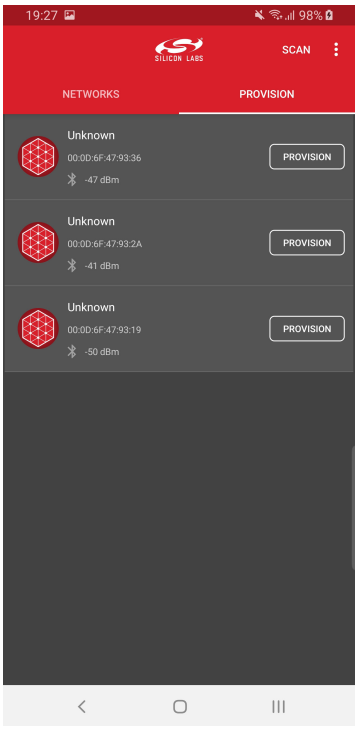
3.4 Use the Demo with an Android Smartphone

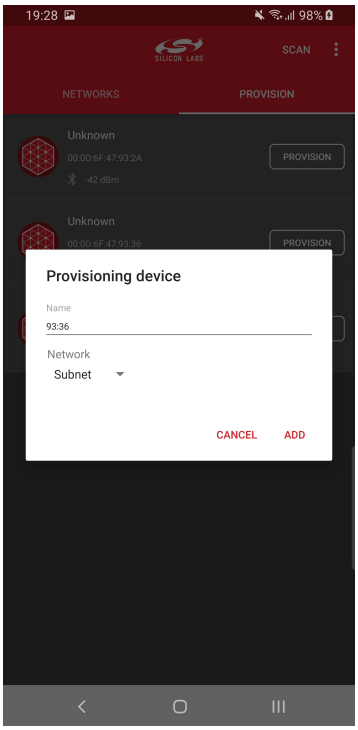
Make sure that all three devices have the status of "**unprovisioned"** on the device LCD screen before starting with the application.

Open the **Bluetooth Mesh** App by Silicon Labs on your Android phone.

Follow the procedures below to set up and use the demonstration.

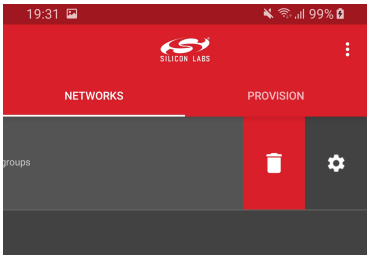
1. Go to provisioning view and search for unprovisioned devices.
2. Select the Bluetooth mesh device you want to provision and configure.
3. Enter the descriptive name for the device and the network you want to add it to.





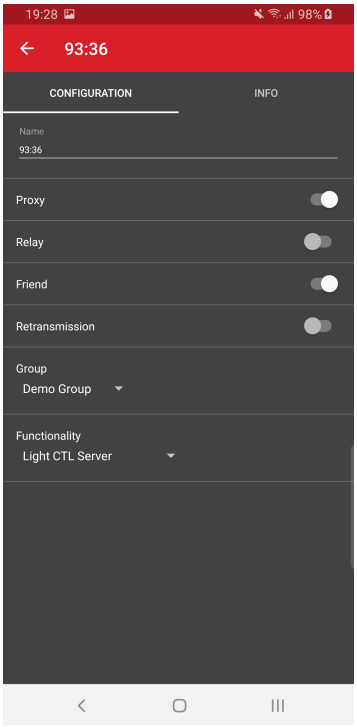
Note: The Android application has a pre-generated network and group, but you can add more groups to the application if you like.

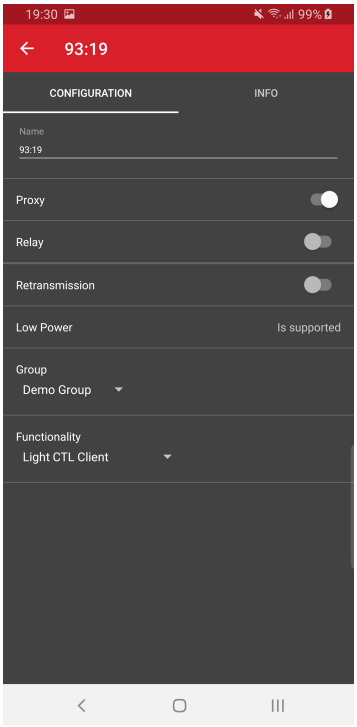
The network and node database can be erased by long-pressing the network in the main view and by pressing the trash icon.

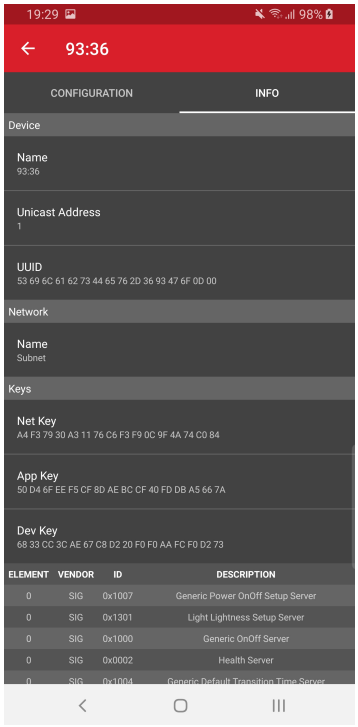


To configure the newly provisioned Bluetooth mesh:

1. Right after provisioning the Android application connects the proxy service on the node.
2. During configuration select the Bluetooth mesh features (proxy, relay, low power, and friend) that you want to enable.
    a. Notice that if you disable proxy, the node can no longer be directly accessed over GATT.
3. Select the functionality (mesh model) that you want to enable.
4. Select the group you want to add the device to.



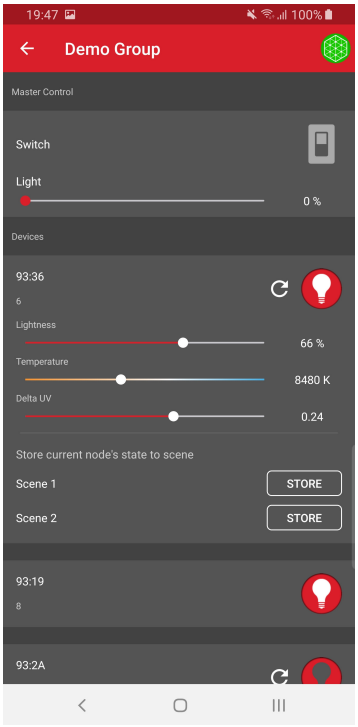


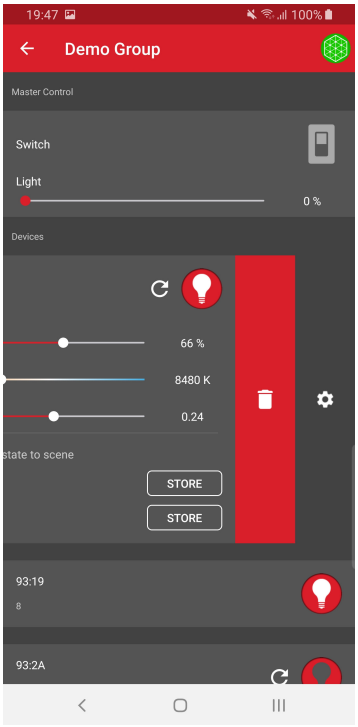


Note: The information view shows the Bluetooth mesh node features, such as Unicast address, UUID, and security keys as well as the supported mesh models. It can be used for debug purposes.

To control a Bluetooth mesh node with the Android application:

1. Select the network and group you want to control .
2. The application will show the available nodes in that group.
3. You can control the light:
    a. Pressing the light bulb icon will send an On/Off message.
    b. Moving the upper slider will send Light Lightness (dimming) messages.
    c. Moving the medium and lower sliders will send CTL (temperature and delta UV) messages.
    d. Pressing [**STORE**] stores the corresponding scene.
4. By going to devices view and either swiping or long-pressing a node you can then either delete or reconfigure the node.





Once the Android application has been used to provision a light bulb and a light switch to a network and group, the light switch (WSTK) can also be used to control the light bulb (WSTK) with the PB0 and PB1 buttons.

PB0 button:

- Short press: Decrease Light Lightness by 10%
- Medium press: Decrease CTL (temperature) value
- Long press: Send Off message
- Very long press (5 seconds or more): Recall scene 1

PB1 button:

- Short press: Increase Light Lightness by 10%
- Medium press: Increase CTL (temperature) value
- Long press: Send On message
- Very long press (5 seconds or more): Recall scene 2

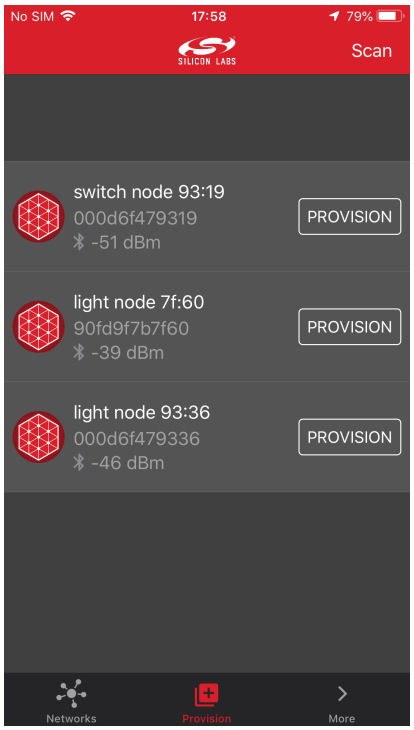
3.5 Use the Demo with an iOS Smartphone

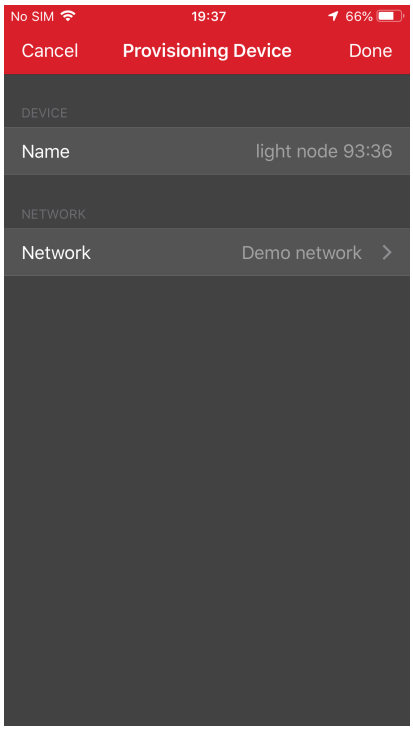
Make sure that all three devices have the status of **"unprovisioned"** on the device LCD screen before starting with the Mobile App.

Open the **Bluetooth Mesh** App by Silicon Labs on your iOS phone.

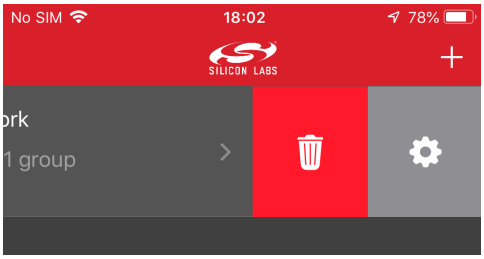
Follow the procedures below to set up and use the demonstration.

1. Create a Bluetooth mesh network.
2. Select the network and create a group.
3. Go to the provisioning view and search for unprovisioned devices.
4. Select the Bluetooth mesh device you want to provision and configure.



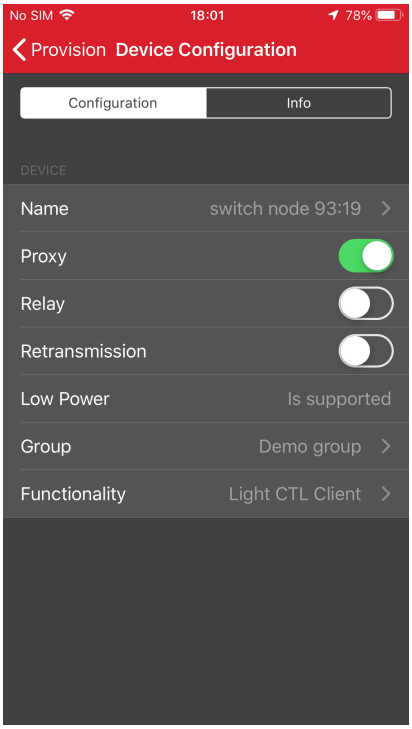


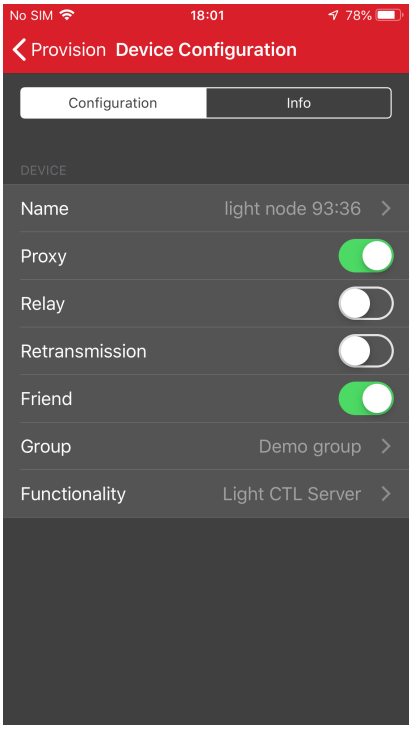
The network and node database can be erased by left-swiping the network in the main view and then pressing the trash icon.

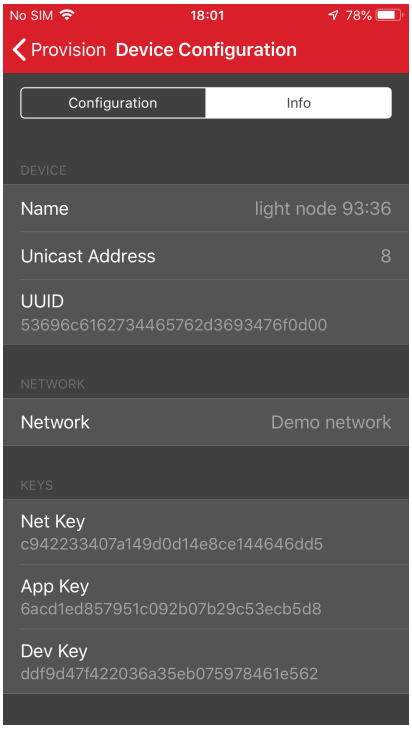


To provision a Bluetooth mesh device and configure the node:

1. During provisioning select the network you want to add the device to.
2. During configuration select the Bluetooth mesh features (proxy, relay, low power and friend) that you want to enable.
    a. Notice that if you disable proxy, the node can no longer be directly accessed over GATT.
3. Select the group you want to add the device to.
4. Finally select the functionality (mesh model) that you want to enable.



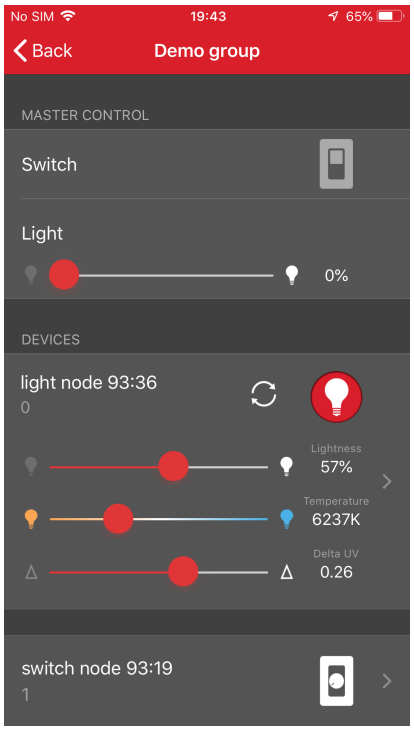


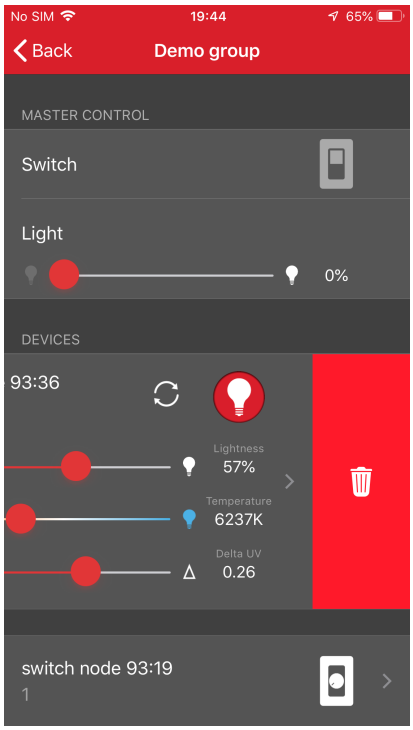


Note: The information view shows the Bluetooth mesh node features, such as Unicast address, UUID, and security keys as well as the supported mesh models. It can be used for debug purposes.

To control a Bluetooth mesh node with the iOS application:

1. Select the network and group you want to control.
2. The application will show the available nodes in that group.
3. You can control the light:
    a. Pressing the light bulb icon will send an On/Off message.
    b. Moving the upper slider will send Light Lightness (dimming) messages.
    c. Moving the medium and lower sliders will send CTL (temperature and delta UV) messages.
    d. Pressing [**STORE**] stores the corresponding scene.
4. By going to the Devices view and tapping a node name you can reconfigure the node. To remove the node from the network, left-swipe it and press the trash icon.





Once the iOS application has been used to provision a light bulb and a light switch to a network and group, the light switch (WSTK) can also be used to control the light bulb (WSTK) with the PB0 and PB1 buttons.

PB0 button:

- Short press: Decrease Light Lightness by 10%
- Medium press: Decrease CTL (temperature) value
- Long press: Send Off message
- Very long press (5 seconds or more): Recall scene 1

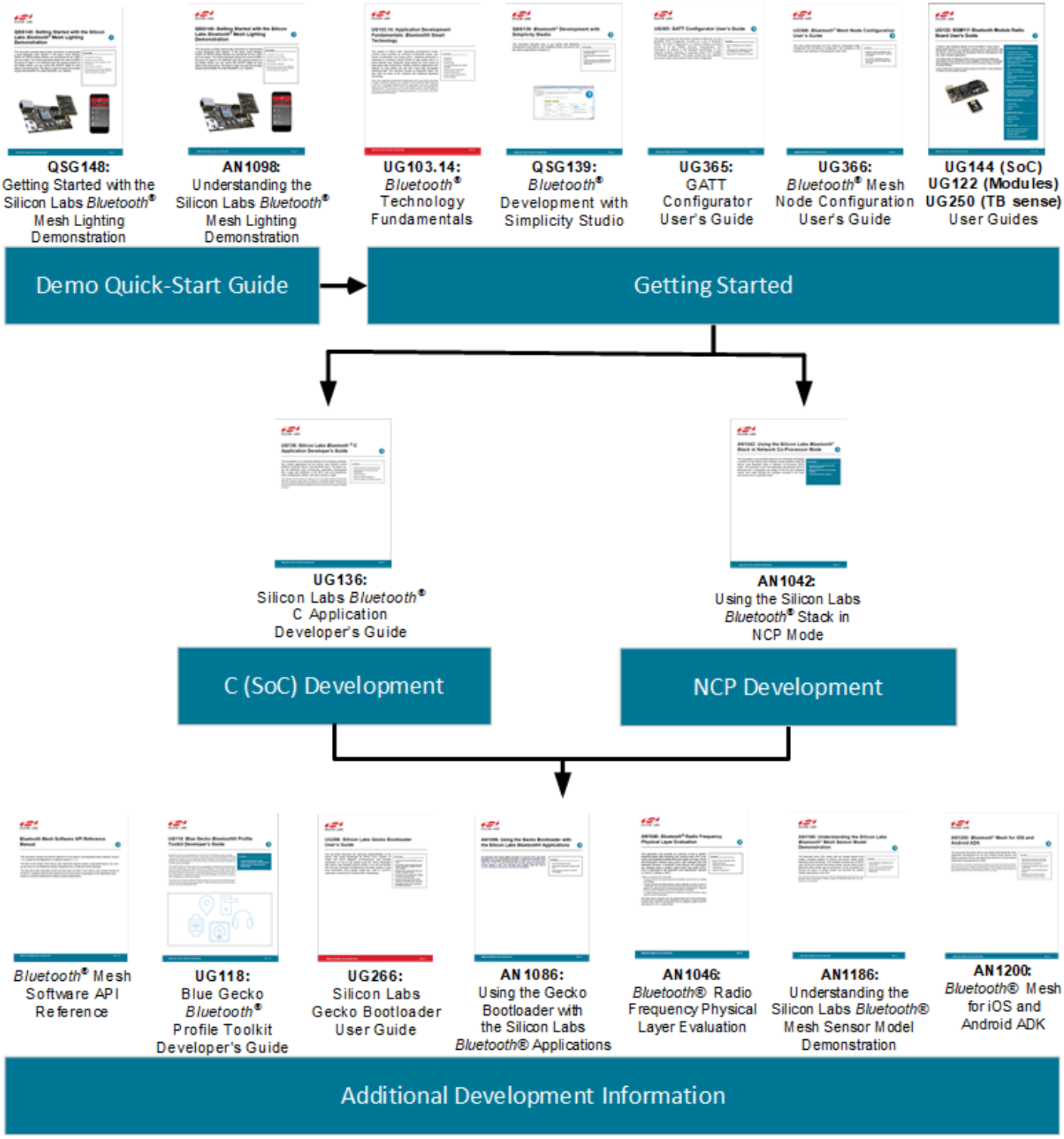
PB1 button:

- Short press: Increase Light Lightness by 10%
- Medium press: Increase CTL (temperature) value
- Long press: Send On message
- Very long press (5 seconds or more): Recall scene 2

4. Next Steps

To understand how the demo works, see *AN1098: Understanding the Silicon Labs Bluetooth Mesh Lighting Demonstration*.

Explore the other documentation provided by Silicon Labs to get started with customizing your own Bluetooth mesh applications. SDK-specific documentation is provided under **SDK Documentation** on the Getting Started tab of the Launcher perspective.



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

SILICON LABS

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

http://www.silabs.com