QSG176: *Bluetooth*® Mesh Quick-Start Guide for SDK v2.x and v3.x

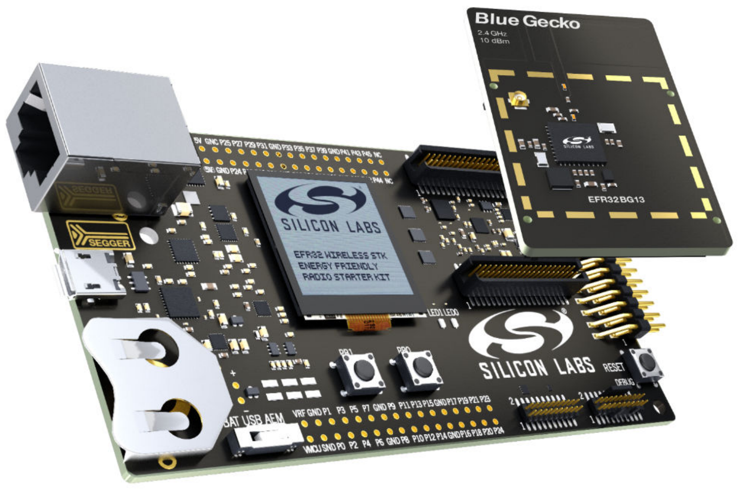
This document describes how to get started with Bluetooth mesh development using the Bluetooth Mesh Software Development Kit (SDK) and Simplicity Studio® 5 with a compatible wireless starter kit. It includes step-by-step instructions to demonstrate a basic Bluetooth mesh network. In this demo, three 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 node, 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.

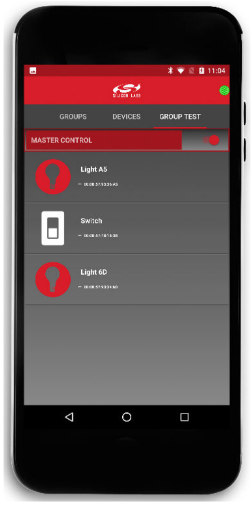
KEY POINTS

- Introducing the Bluetooth mesh development environment
- Using WSTKs to demonstrate a basic Bluetooth mesh network
- Starting Bluetooth mesh application development in Simplicity Studio

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 AN1200.1: Bluetooth® Mesh for iOS and Android ADK.

f you are developing with Bluetooth Mesh SDK v4.x for specification version 1.1, see *QSG183: Bluetooth Mesh Quick-Start Guide for SDK 4.x*. If you use Simplicity Studio 4 with Bluetooth Mesh SDK v1.x, find corresponding content in *QSG148: Getting Started with the Silicon Labs Bluetooth Mesh Demonstration Software in SDK v1.x*.





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 Kit(s)

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 described in section 4. Getting Started with the Bluetooth Mesh Demonstration Software, you need to have **three (3)** EFR32™ mainboards 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

Option 4: QTY(1) of PN: SLWSTK6021A kit: https://www.silabs.com/development-tools/wireless/efr32xg22-wireless-starter-kit AND either QTY(2) Option 1 or QTY(1) Option 2 or QTY(1) Option 3.

This demo requires either **EFR32BG24**, **EFR32MG24**, **EFR32MG21**, **EFR32BG13**, **EFR32MG13**, **EFR32BG12**, or **EFR32MG12** radio boards. **EFR32MG22** and **EFR32BG22** can be used but are suggested for Low Power Nodes only. If you already have the mainboards, 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. SLWSTK6021A can also be substituted, but is recommended only for Low Power Nodes.

1.2 Download Simplicity Studio

The Gecko SDK (GSDK) is the suite of Silicon Labs SDKs that includes the Bluetooth mesh SDK. To quickly get started with the GSDK and Bluetooth mesh, start by installing Simplicity Studio 5, which will set up your development environment and walk you through the installation of the GSDK. Go to: http://www.silabs.com/simplicity-studio to download the latest SSv55 version compatible with your computer's operating system. 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. Alternatively, Gecko SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.

This document focuses on development and use in the SSv5 environment. It assumes you are generally familiar with the SSv5 Launcher perspective. SSv5 installation and getting started instructions along with a set of detailed references can be found in the online Simplicity Studio 5 User's Guide, available on https://docs.silabs.com/ and through the SSv5 help menu.

1.3 Download the Mobile App

Download the 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

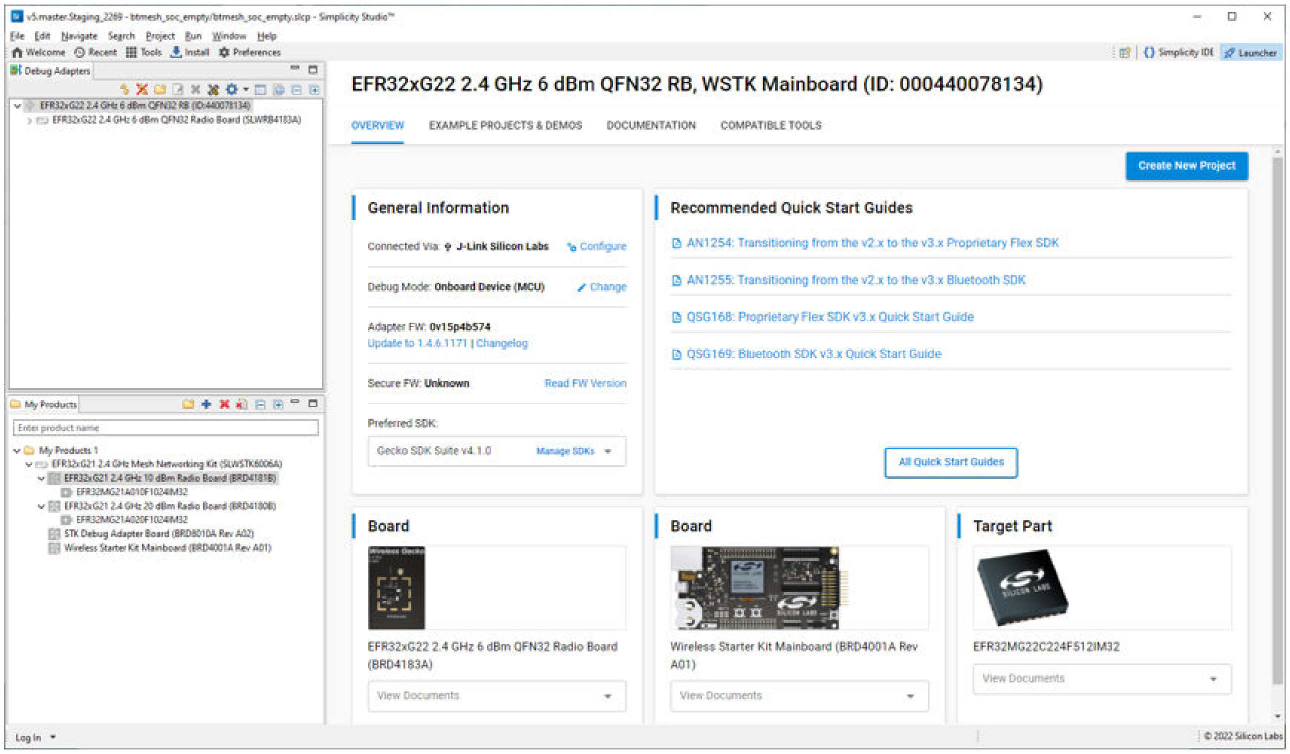
Google Play:

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

Note: The minimum requirement for the smartphone is Android 9 (API23) or iOS 13.

1.4 Documentation

Hardware-specific documentation may be accessed through links on the part Overview tab in Simplicity Studio 5.



EFR32xG22 2.4 GHz 6 dBm QFN32 RB (ID:440078134)
EFR32xG22 2.4 GHz 6 dBm QFN32 Radio Board (SLWRB4183A)

My Products 1
EFR32xG21 2.4 GHz Mesh Networking Kit (SLWSTK6006A)
EFR32xG21 2.4 GHz 10 dBm Radio Board (BRD4181B)
EFR32MG21A010F1024IM32
EFR32xG21 2.4 GHz 20 dBm Radio Board (BRD4180B)
EFR32MG21A020F1024IM32
STK Debug Adapter Board (BRD8010A Rev A02)
Wireless Starter Kit Mainboard (BRD4001A Rev A01)

EFR32xG22 2.4 GHz 6 dBm QFN32 RB, WSTK Mainboard (ID: 000440078134)

General Information

Connected Via: J-Link Silicon Labs
Debug Mode: Onboard Device (MCU)
Adapter FW: 0v15p4b574
Update to 1.4.6.1171 | Changelog
Secure FW: Unknown
Preferred SDK:
Gecko SDK Suite v4.1.0

Recommended Quick Start Guides

AN1254: Transitioning from the v2.x to the v3.x Proprietary Flex SDK
AN1255: Transitioning from the v2.x to the v3.x Bluetooth SDK
QSG168: Proprietary Flex SDK v3.x Quick Start Guide
QSG169: Bluetooth SDK v3.x Quick Start Guide

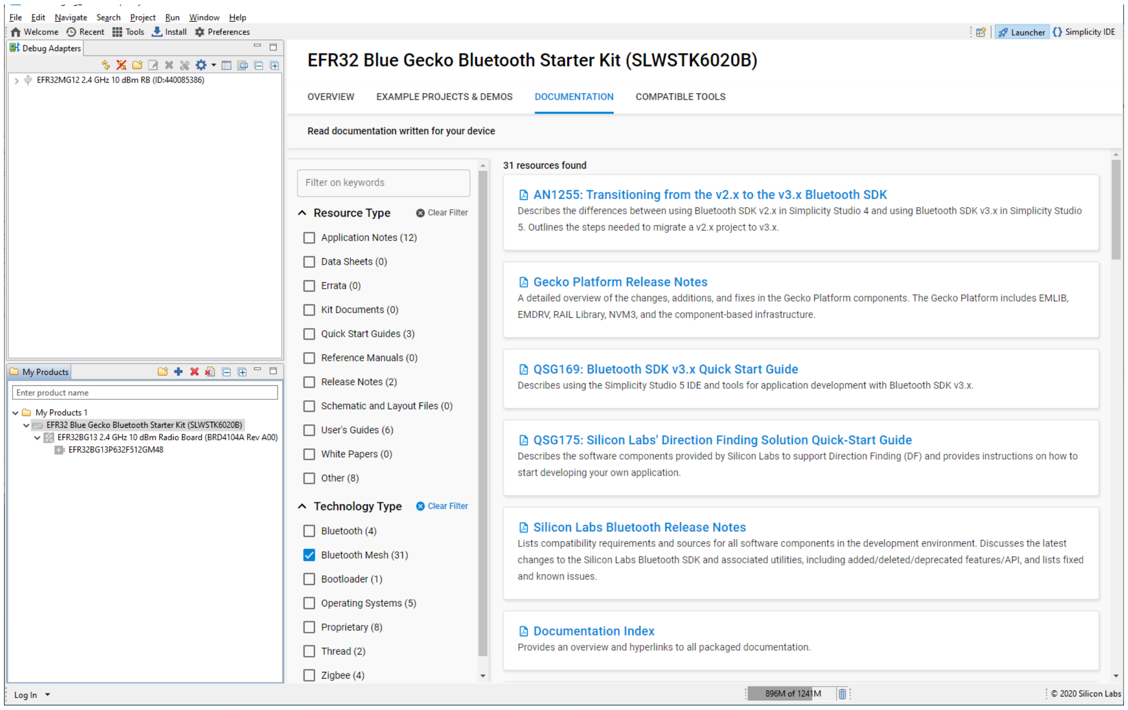
Board
EFR32xG22 2.4 GHz 6 dBm QFN32 Radio Board (BRD4183A)

Board
Wireless Starter Kit Mainboard (BRD4001A Rev A01)

Target Part
EFR32MG22C224F512IM32

© 2022 Silicon Labs

SDK documentation, User's Guides and other references are available through the Documentation tab.



EFR32MG12 2.4 GHz 10 dBm RB (ID:440085386)

My Products 1
EFR32 Blue Gecko Bluetooth Starter Kit (SLWSTK6020B)
EFR32BG13 2.4 GHz 10 dBm Radio Board (BRD4104A Rev A00)
EFR32BG13P632F512GM48

EFR32 Blue Gecko Bluetooth Starter Kit (SLWSTK6020B)

Read documentation written for your device

Resource Type
- [ ] Application Notes (12)
- [ ] Data Sheets (0)
- [ ] Errata (0)
- [ ] Kit Documents (0)
- [ ] Quick Start Guides (3)
- [ ] Reference Manuals (0)
- [ ] Release Notes (2)
- [ ] Schematic and Layout Files (0)
- [ ] User's Guides (6)
- [ ] White Papers (0)
- [ ] Other (8)

Technology Type
- [ ] Bluetooth (4)
- [x] Bluetooth Mesh (31)
- [ ] Bootloader (1)
- [ ] Operating Systems (5)
- [ ] Proprietary (8)
- [ ] Thread (2)
- [ ] Zigbee (4)

31 resources found

AN1255: Transitioning from the v2.x to the v3.x Bluetooth SDK
Describes the differences between using Bluetooth SDK v2.x in Simplicity Studio 4 and using Bluetooth SDK v3.x in Simplicity Studio 5. Outlines the steps needed to migrate a v2.x project to v3.x.

Gecko Platform Release Notes
A detailed overview of the changes, additions, and fixes in the Gecko Platform components. The Gecko Platform includes EMLIB, EMDRV, RAIL Library, NVM3, and the component-based infrastructure.

QSG169: Bluetooth SDK v3.x Quick Start Guide
Describes using the Simplicity Studio 5 IDE and tools for application development with Bluetooth SDK v3.x.

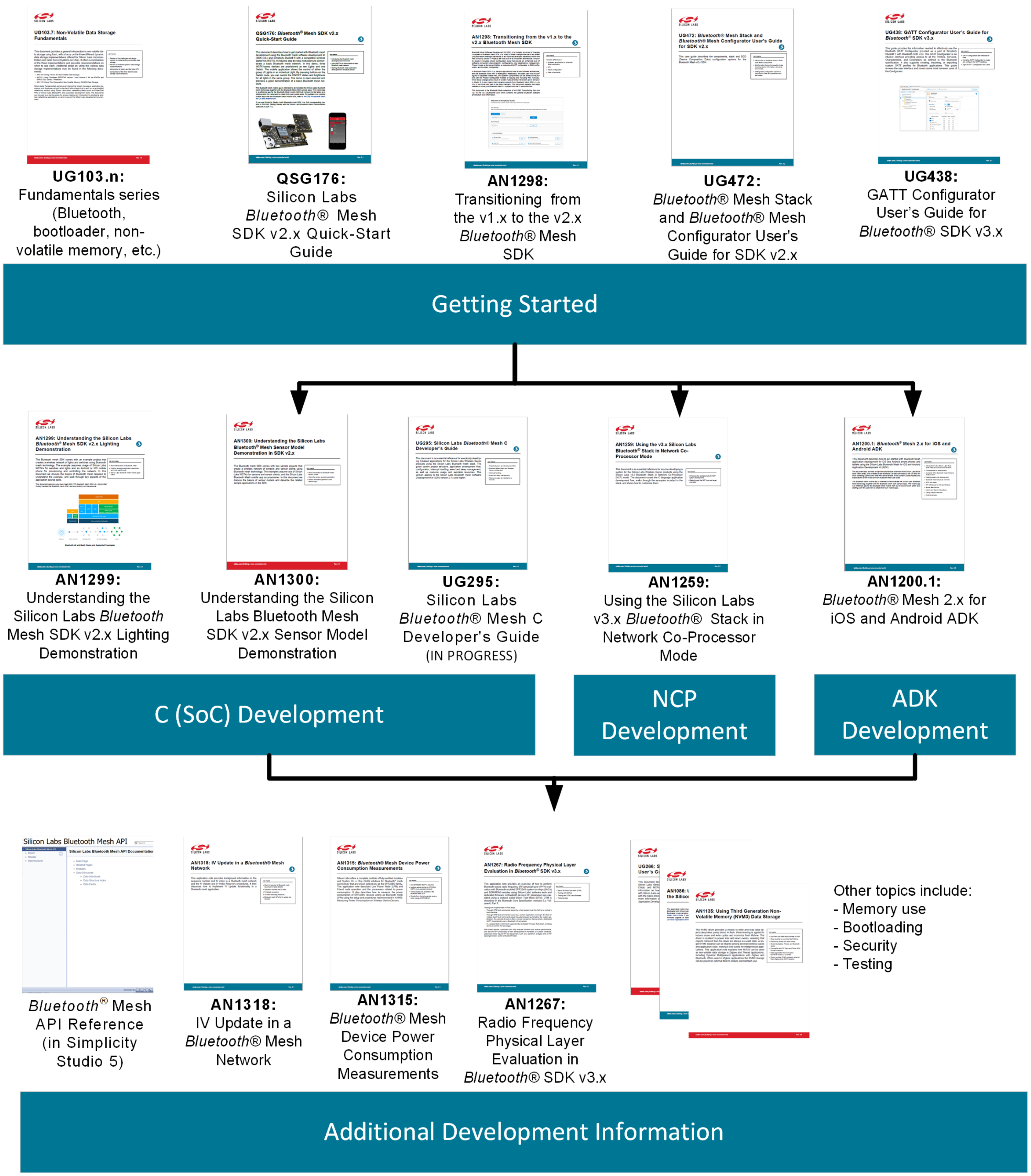
QSG175: Silicon Labs' Direction Finding Solution Quick-Start Guide
Describes the software components provided by Silicon Labs to support Direction Finding (DF) and provides instructions on how to start developing your own application.

Silicon Labs Bluetooth Release Notes
Lists compatibility requirements and sources for all software components in the development environment. Discusses the latest changes to the Silicon Labs Bluetooth SDK and associated utilities, including added/deleted/deprecated features/API, and lists fixed and known issues.

Documentation Index
Provides an overview and hyperlinks to all packaged documentation.

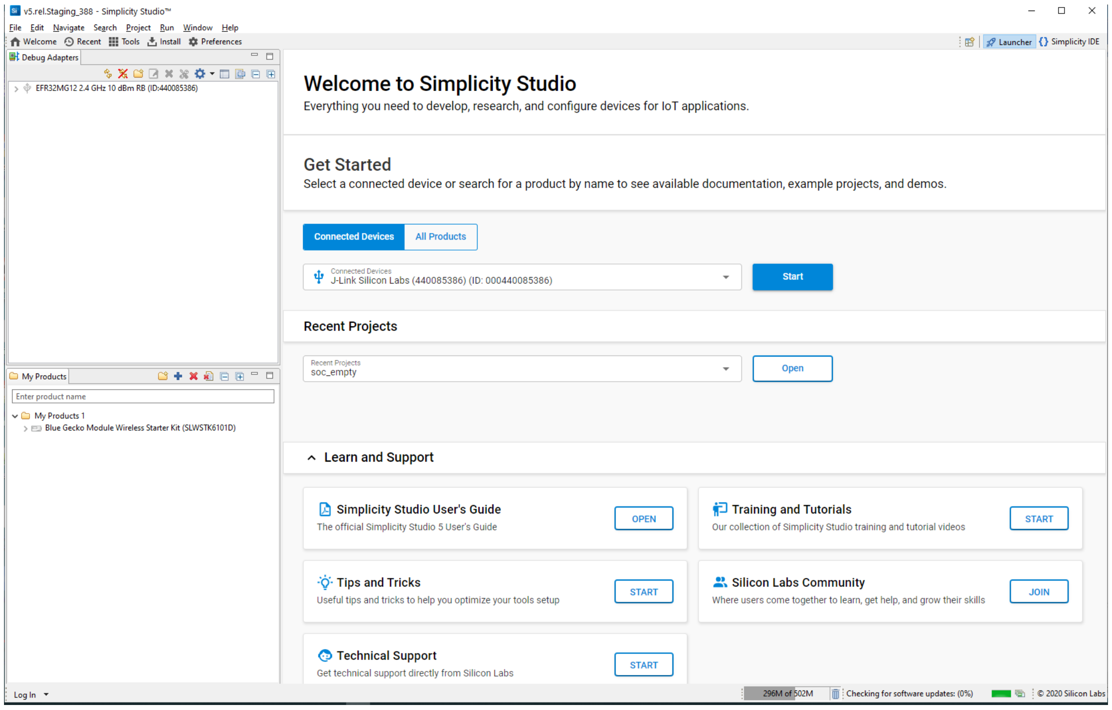
896M of 1241M
© 2020 Silicon Labs

Key documentation for the Bluetooth mesh SDK is summarized in the following figure:


UG103.n: Fundamentals series (Bluetooth, bootloader, non-volatile memory, etc.)
QSG176: Silicon Labs Bluetooth® Mesh SDK v2.x Quick-Start Guide
AN1298: Transitioning from the v1.x to the v2.x Bluetooth® Mesh SDK
UG472: Bluetooth® Mesh Stack and Bluetooth® Mesh Configurator User's Guide for SDK v2.x
UG438: GATT Configurator User's Guide for Bluetooth® SDK v3.x
Getting Started
AN1299: Understanding the Silicon Labs Bluetooth Mesh SDK v2.x Lighting Demonstration
AN1300: Understanding the Silicon Labs Bluetooth Mesh SDK v2.x Sensor Model Demonstration
UG295: Silicon Labs Bluetooth® Mesh C Developer's Guide (IN PROGRESS)
AN1259: Using the Silicon Labs v3.x Bluetooth® Stack in Network Co-Processor Mode
AN1200.1: Bluetooth® Mesh 2.x for iOS and Android ADK
C (SoC) Development
NCP Development
ADK Development
Bluetooth® Mesh API Reference (in Simplicity Studio 5)
AN1318: IV Update in a Bluetooth® Mesh Network
AN1315: Bluetooth® Mesh Device Power Consumption Measurements
AN1267: Radio Frequency Physical Layer Evaluation in Bluetooth® SDK v3.x
Other topics include:
- Memory use
- Bootloading
- Security
- Testing
Additional Development Information

1.5 Obtaining Support

You can access the Silicon Labs support portal at https://www.silabs.com/support through Simplicity Studio Resources, or on the Simplicity Studio 5 welcome page under Learn and Support.


v5.rel.Staging_388 - Simplicity Studio™
File Edit Navigate Search Project Run Window Help
Welcome Recent Tools Install Preferences
Launcher Simplicity IDE
Debug Adapters
My Products
Enter product name
Connected Devices
All Products
Start
Open
OPEN
START
JOIN
Log In

EFR32MG12 2.4 GHz 10 dBm RB (ID:440085386)

My Products 1
Blue Gecko Module Wireless Starter Kit (SLWSTK6101D)

Welcome to Simplicity Studio
Everything you need to develop, research, and configure devices for IoT applications.

Get Started
Select a connected device or search for a product by name to see available documentation, example projects, and demos.

Connected Devices: J-Link Silicon Labs (440085386) (ID: 000440085386)

Recent Projects
Recent Projects: soc_empty

Learn and Support

Simplicity Studio User's Guide
The official Simplicity Studio 5 User's Guide

Training and Tutorials
Our collection of Simplicity Studio training and tutorial videos

Tips and Tricks
Useful tips and tricks to help you optimize your tools setup

Silicon Labs Community
Where users come together to learn, get help, and grow their skills

Technical Support
Get technical support directly from Silicon Labs

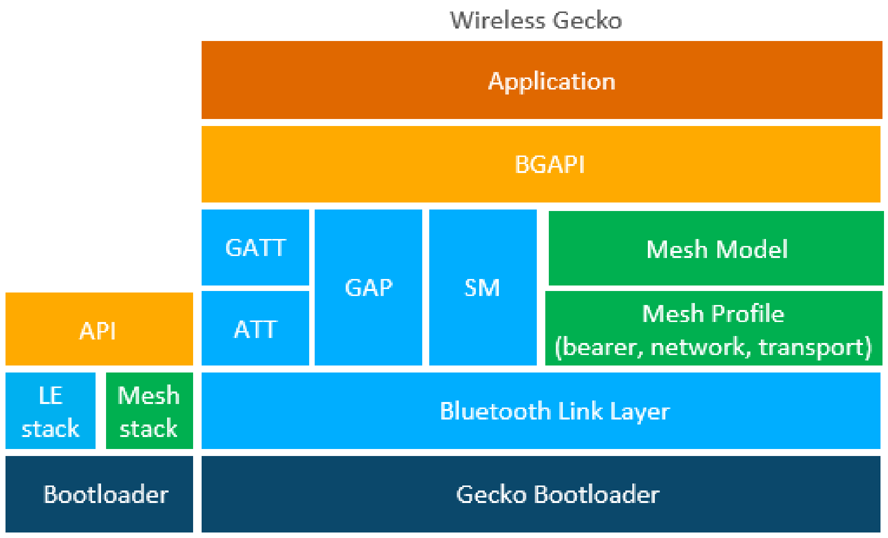
296M of 502M | Checking for software updates: (0%) | © 2020 Silicon Labs

2. About the Bluetooth Mesh SDK

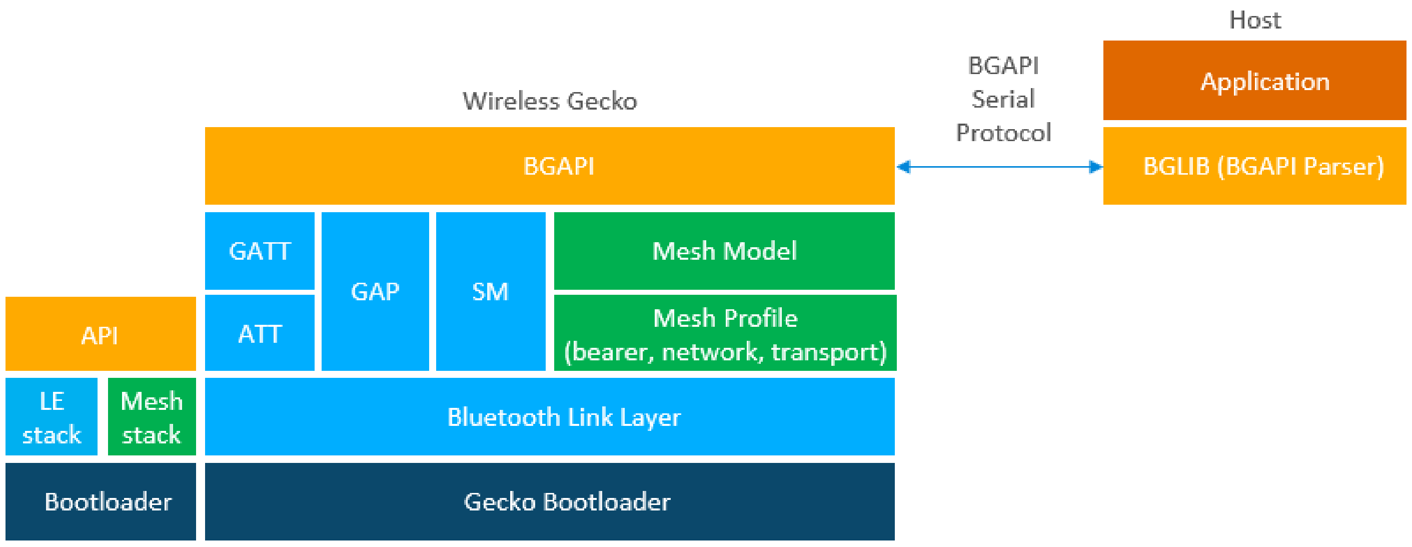
The Silicon Labs Bluetooth mesh stack is an advanced Bluetooth mesh protocol stack implementing the Bluetooth mesh standard. It can run alongside the Bluetooth Low Energy (LE) stack, using a common link layer, which allows using LE features in parallel. The Silicon Labs Bluetooth mesh stack is meant 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. For more information, see AN1259: Using the Silicon Labs Bluetooth® Stack v3.x and Higher in Network Co-Processor Mode.



2.1 Bluetooth Mesh Stack Features

The features of the Silicon Labs Bluetooth mesh stack are listed in the following table. For details on the features of the Bluetooth **Low Energy** stack, refer to QSG169: Bluetooth® SDK v3.x Quick-Start Guide

Table 2.1. Bluetooth Mesh Stack Features

Feature	Value and Comment
Bluetooth mesh version	Bluetooth mesh 1.0.1
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 (reject list)
Host (NCP) interfaces	4-wire UART with RTS/CTS control or 2-wire UART without RTS/CTS GPIOs 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, EFR32[B|M]G22, EFR32[B|M]G24: NVM3

Table 2.2. Supported Models

Model	SIG Model ID	Example App (1)
Model Group: NA		
Vendor	N/A	N/A
Model Group: Generic		
Generic OnOff Server Generic OnOff Client	0x1000 0x1001	SoC Light, SoC HSL Light SoC Switch, SoC Switch Low Power
Generic Level Server Generic Level Client	0x1002 0x1003	SoC Light, SoC HSL Light N/A
Generic Default Transition Time Server Generic Default Transition Time Client	0x1004 0x1005	SoC Light, SoC HSL Light N/A
Generic Power OnOff Server Generic Power OnOff Setup Server Generic Power OnOff Client	0x1006 0x1007 0x1008	SoC Light, SoC HSL Light SoC Light, SoC HSL Light N/A
Generic Power Level Server Generic Power Level Setup Server Generic Power Level Client	0x1009 0x100A 0x100B	N/A N/A N/A

Model	SIG Model ID	Example App [1]
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 Sensor Server SoC Sensor Server SoC Sensor Client
Model Group: Times and Scenes		
Time Server Time Setup Server Time Client	0x1200 0x1201 0x1202	SoC Light, SoC HSL Light SoC Light, SoC HSL Light N/A
Scene Server Scene Setup Server Scene Client	0x1203 0x1204 0x1205	SoC Light, SoC HSL Light SoC Light, SoC HSL Light SoC Switch, SoC Switch Low Power
Scheduler Server Scheduler Setup Server Scheduler Client	0x1206 0x1207 0x1208	SoC Light, SoC HSL Light SoC Light, SoC HSL Light N/A
Model Group: Lighting		
Light Lightness Server Light Lightness Setup Server Light Lightness Client	0x1300 0x1301 0x1302	SoC Light, SoC HSL Light SoC Light, SoC HSL Light SoC Switch, SoC Switch Low Power
Light CTL Server Light CTL Setup Server Light CTL Client Light CTL Temperature Server	0x1303 0x1304 0x1305 0x1306	SoC Light SoC Light SoC Switch, SoC Switch Low Power SoC Light
Light HSL Server Light HSL Setup Server Light HSL Client Light HSL Hue Server Light HSL Saturation Server	0x1307 0x1308 0x1309 0x130A 0x130B	SoC HSL Light SoC HSL Light N/A SoC HSL Light SoC HSL Light
Light LC Server Light LC Setup Server Light LC Client	0x130F 0x1310 0x1311	SoC Light, SoC HSL Light SoC Light, SoC HSL Light N/A

[1] In Simplicity Studio 5, example app names are preceded with 'Bluetooth Mesh -'.

2.2 Bluetooth Mesh Stack Limitations

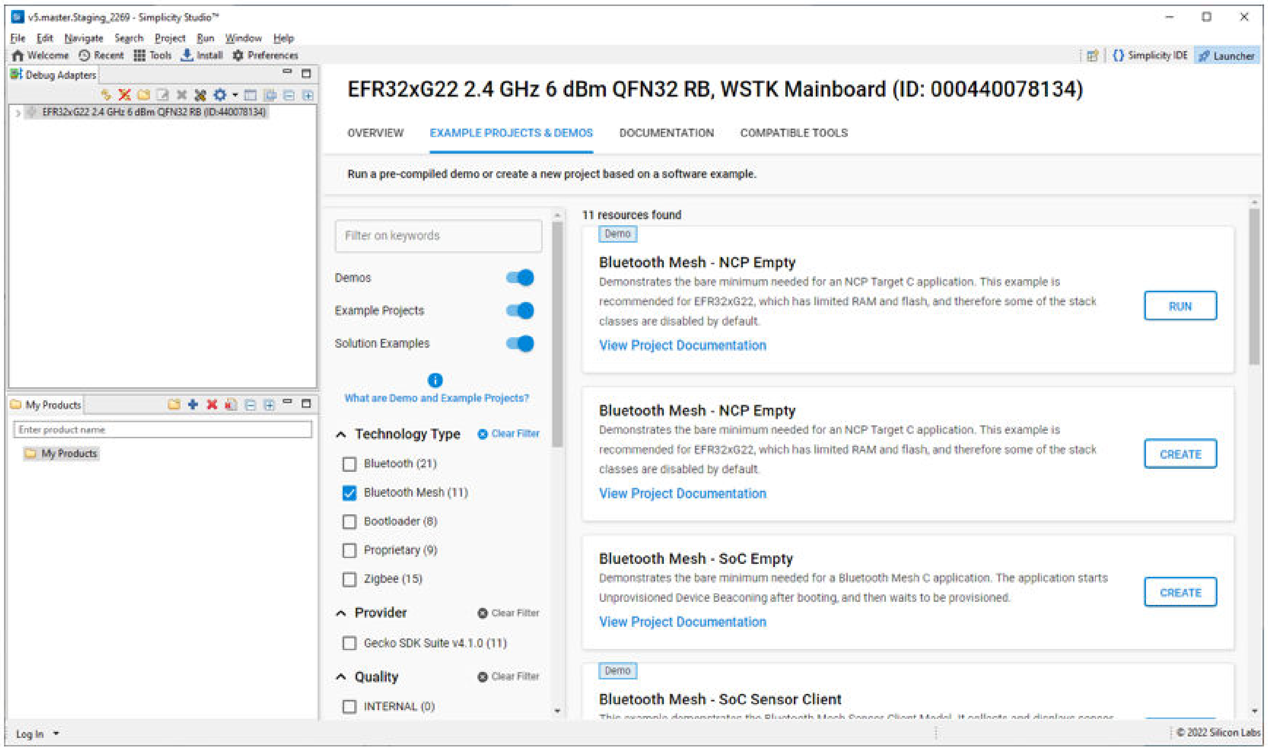
Component	Feature	Value and Comment
Mesh Node (EFR32)	Network Keys on a node(1)	Maximum of 7
	Application Keys on a node	Maximum of 8
	Number of nodes that can be communicated with	Maximum of 4096 (depending on available RAM and NVM3)
	Concurrent segmented messages being received	Maximum of 255 (depending 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 sessions	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. About Demos and Examples

Because starting application development from scratch is difficult, the Bluetooth Mesh SDK comes with a number of built-in demos and examples covering the most frequent use cases, as shown in the following figure. Demos are pre-built application images that you can run immediately. Software examples can be modified before building the application image. Demos with the same name as Software examples are built from their respective example. Click **View Project Documentation** to see additional information about some examples. This is also displayed on a **readme** tab when you create a project based on the example.

Use the **Demos** and **Example Projects** switches to filter on only examples or only demos. Demos are also noted by the blue Demo tag in the upper left of the card. The Solution Examples filter is provided for future use.



EFR32xG22 2.4 GHz 6 dBm QFN32 RB, WSTK Mainboard (ID: 000440078134)

Run a pre-compiled demo or create a new project based on a software example.

11 resources found

Bluetooth Mesh - NCP Empty
Demonstrates the bare minimum needed for an NCP Target C application. This example is recommended for EFR32xG22, which has limited RAM and flash, and therefore some of the stack classes are disabled by default.
View Project Documentation

Bluetooth Mesh - NCP Empty
Demonstrates the bare minimum needed for an NCP Target C application. This example is recommended for EFR32xG22, which has limited RAM and flash, and therefore some of the stack classes are disabled by default.
View Project Documentation

Bluetooth Mesh - SoC Empty
Demonstrates the bare minimum needed for a Bluetooth Mesh C application. The application starts Unprovisioned Device Beaconing after booting, and then waits to be provisioned.
View Project Documentation

Bluetooth Mesh - SoC Sensor Client

Note: The demos and examples you see are determined by the part selected. If you are using a custom solution with more than one part, click the part you are working with to see only the items applicable to that part. Some functionality in these demos and examples depends on the features of the part, such as whether or not an LCD included on the mainboard.

To download and run a demo on your device, click **RUN** on the desired demo card. The demo automatically downloads to the selected device.

Bluetooth Mesh Examples and Demos

The following examples are provided. Examples with (*) in their names have a matching pre-built demo for some devices. Demos contain both a bootloader and the application. The compatible bootloader information provided for some examples is for later software development purposes. See section 5. Getting Started with Application Development for more information.

Bluetooth Mesh - NCP Empty(*): Bluetooth Mesh NCP (Network Co-Processor) target demonstrates the bare minimum needed for a Bluetooth mesh NCP Target C application, that makes it possible for the NCP Host Controller to access the Bluetooth mesh stack via UART. It provides access to the host layer via BGAPI and not to the link layer via HCI. The communication between the Host Controller and the target can be secured by installing the Secure NCP component. This example requires the BGAPI UART DFU Bootloader.

Bluetooth Mesh - SoC Empty: This example demonstrates the bare minimum needed for a Bluetooth mesh C application that supports Over-the-Air Device Firmware Upgrading (OTA DFU). The application starts Unprovisioned Device Beaconing after boot, and waits to be provisioned to a Mesh Network. This example can be used as a starting point for an SoC project and it can be customized by adding new components using the Project Configurator or by modifying the application (app.c). This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory.

Bluetooth Mesh - SoC Sensor Client(*): This example demonstrates the Bluetooth Mesh Sensor Client Model. It collects and displays sensor measurement data from remote device(s) (for example **Bluetooth Mesh - SoC Sensor Server**). The current status is displayed on the LCD (if one is present on the mainboard) and also sent to UART. CLI commands may substitute for button presses if the mainboard has only one button available. This example requires one of the Internal Storage Bootloader (single image) variants depending on device memory.

Bluetooth Mesh - SoC Sensor Server(*): This example demonstrates the Bluetooth Mesh Sensor Server Model and Sensor Setup Server Model. It measures temperature and people count (and also illuminance with some parts) and sends the measurement data to a remote device (for example, **Bluetooth Mesh - SoC Sensor Client**). The current status is displayed on the LCD (if one is present on the mainboard) and also sent to UART. CLI commands may substitute for button presses if the mainboard has only one button available. This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory.

Bluetooth Mesh - SoC Switch(*): This example is an out-of-the-box Software Demo optimized for user experience where the device acts as a switch. Button presses on the WSTK or CLI commands can control the state, lightness, and color temperature of the LEDs as well as scenes on a remote device (**Bluetooth Mesh - SoC Light**). The example also acts as an LPN and tries to establish friendship. The status messages are displayed on the LCD (if one is present on the mainboard) and also sent to UART. The example is based on the Bluetooth Mesh Generic On/Off Client Model, the Light Lightness Client Model, the Light CTL Client Model, and the Scene Client Model. This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory.

Bluetooth Mesh - SoC Switch Low Power(*): This example is an out-of-the-box Software Demo optimized for low current consumption where the device acts as a switch. It has disabled CLI, logging, and LCD. Button presses on the mainboard can control the state, lightness, and color temperature of the LEDs as well as scenes on a remote device (**Bluetooth Mesh - SoC Light**). The example also acts as an LPN and tries to establish friendship. The example is based on the Bluetooth Mesh Generic On/Off Client Model, the Light Lightness Client Model, the Light CTL Client Model, and the Scene Client Model. This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory

Bluetooth Mesh - SoC Light(*): This example is an out-of-the-box Software Demo where the LEDs of the device can be controlled by button presses on another device (**Bluetooth Mesh - SoC Switch**). The LEDs can be switched on and off, and the lighting intensity, color temperature, and Delta UV (on some devices shown only on the LCD or in UART logs) can also be set. The example also tries to establish friendship as a Friend node and prints its status to the LCD or UART (the target device determines if the feature is enabled and the output status). The example is based on the Bluetooth Mesh Generic On/Off Model, the Light Lightness Model, the Light CTL Server Model, and the Light LC Server Model. This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory.

Bluetooth Mesh - SoC HSL Light(*): This example is an out-of-the-box Software Demo where the LEDs of the device can be controlled by button presses on another device (Bluetooth Mesh - SoC Switch). The LEDs can be switched on and off, and their lighting intensity can also be set. Hue and saturation (shown only on the LCD or in UART logs, depending on the mainboard) can be set by the Light HSL Client model. The example also tries to establish friendship as a Friend node and prints its status to the LCD or UART. The example is based on the Bluetooth Mesh Generic On/Off Model, the Light Lightness Model, the Light HSL Server Model and the Light LC Server Model. This example requires one of the Internal Storage Bootloader (single image) variants, depending on device memory.

Interoperability Test Demo

The interoperability (IOP) tests check if the Bluetooth mesh stack running on the board is compatible with a smartphone or not. These are test procedures containing several test cases for each Bluetooth mesh operation. The demos are meant to be used with the Bluetooth mesh mobile application, and are currently supported only by SLWRB4104A EFR32BG13, SLWRB4181A EFR32MG21, and SLWRB4181B EFR32MG21.

Bluetooth Mesh - IOP Test - Friend node: Friend example for IOP test. This node acts as a friend for the low power node and caches messages sent to it when the low power node is sleeping.

Bluetooth Mesh - IOP Test - LPN node: Low power node example for IOP test. This node acts as a typical low power device and sleeps most of the time. It needs a friend node to cache messages and forward them when polled. This demo is also available for SLWRB4182A EFR32MG22.

Bluetooth Mesh - IOP Test - Proxy node: Proxy example for IOP test. This node forwards/relays messages between GATT and advertising bearers in the network.

Bluetooth Mesh - IOP Test - Relay node: Relay example for IOP test. This node acts as a relay. If a node is out of range for another node, it relays messages between the two, provided the relay node is in range for both.

NCP Host Examples

Two Bluetooth Mesh Network Co-Processor (NCP) host C application examples demonstrate accessing the Bluetooth mesh stack via UART. Access is to the NCP target stack layer via BGAPI and not to the link layer via HCI. The examples can be found in the GSDK install folder under *app/bluetooth/example_host*. The examples require a POSIX environment for compilation, such as Linux, macOS, or Windows MSYS2.

btmesh_host_empty: Minimal host-side project structure, used as a starting point for NCP host applications. Use it with the **Bluetooth Mesh - NCP Empty** NCP target application flashed to the radio board.

btmesh_host_provisioner: The Host Provisioner example demonstrates using a NCP node connected to a PC as a provisioner. Through this node the user can provision, configure and reset other nodes. The Bluetooth mesh network is created and handled by the NCP node so network management options are available as well. Use it with the **Bluetooth Mesh - NCP Empty** NCP target application flashed to the radio board.

Python-Based NCP Host Examples

Python-based NCP host examples can be accessed at https://github.com/SiliconLabs/pybgapi-examples. These examples are meant to be used with PyBGAPI (https://pypi.org/project/pybgapi/). Use with the **Bluetooth Mesh - NCP Empty** NCP target application flashed to the radio board

4. Getting Started with the Bluetooth Mesh Demonstration Software

To get started with Bluetooth mesh demo software, you should have downloaded Simplicity Studio 5 (SSv5) and the Bluetooth Mesh SDK as described in the Simplicity Studio 5 User's Guide, available online and through the SSv5 help menu.

4.1 Preparing the Mainboard

The layout of the Wireless Starter Kit mainboard with attached EFR32BG13 radio board is shown in the following figure:

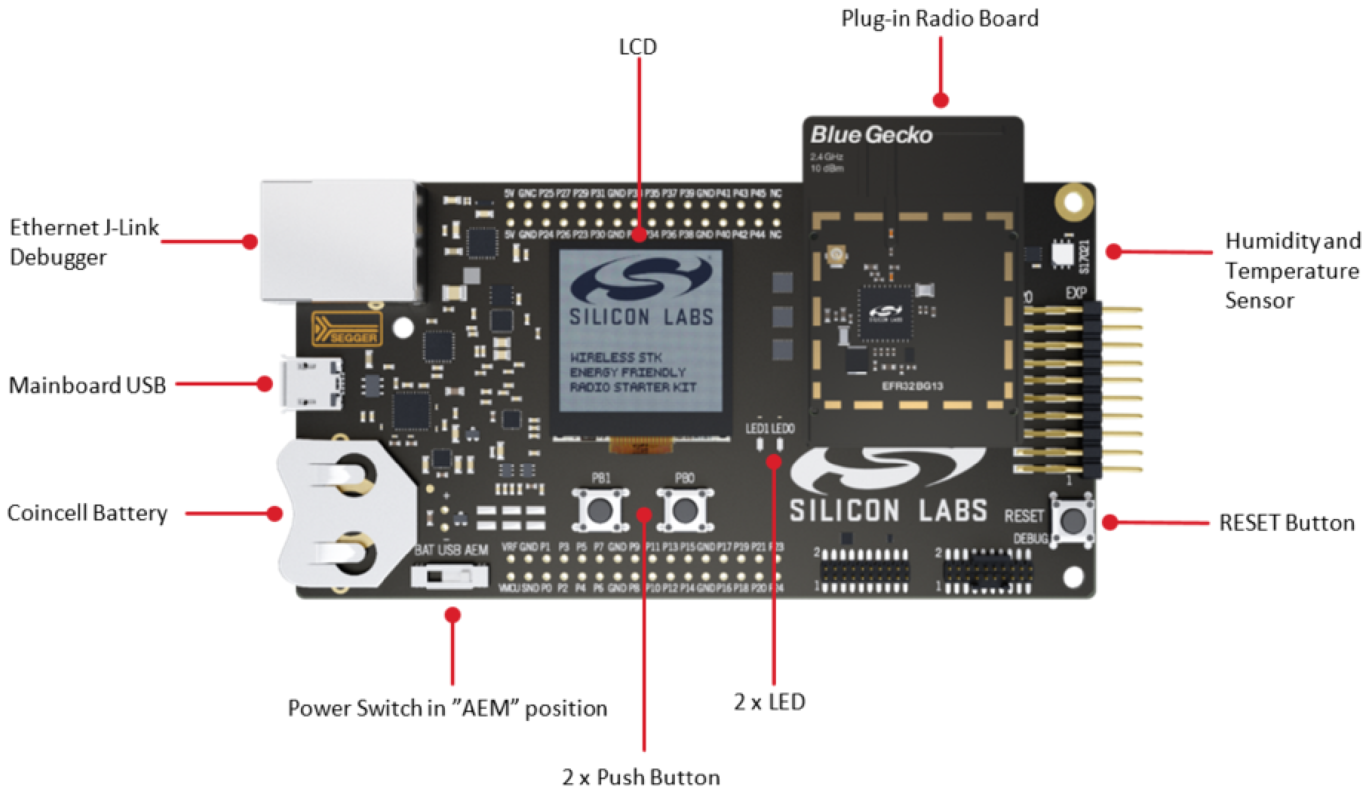


Figure 4.1. WSTK Mainboard with Radio Board Attached

1. Connect a Blue Gecko Radio Board to the mainboard.

   Use radio board SLWRB4104A **EFR32BG13** 2.4 GHz (+10 dBm) for this demo experience.
2. Connect the mainboard 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:

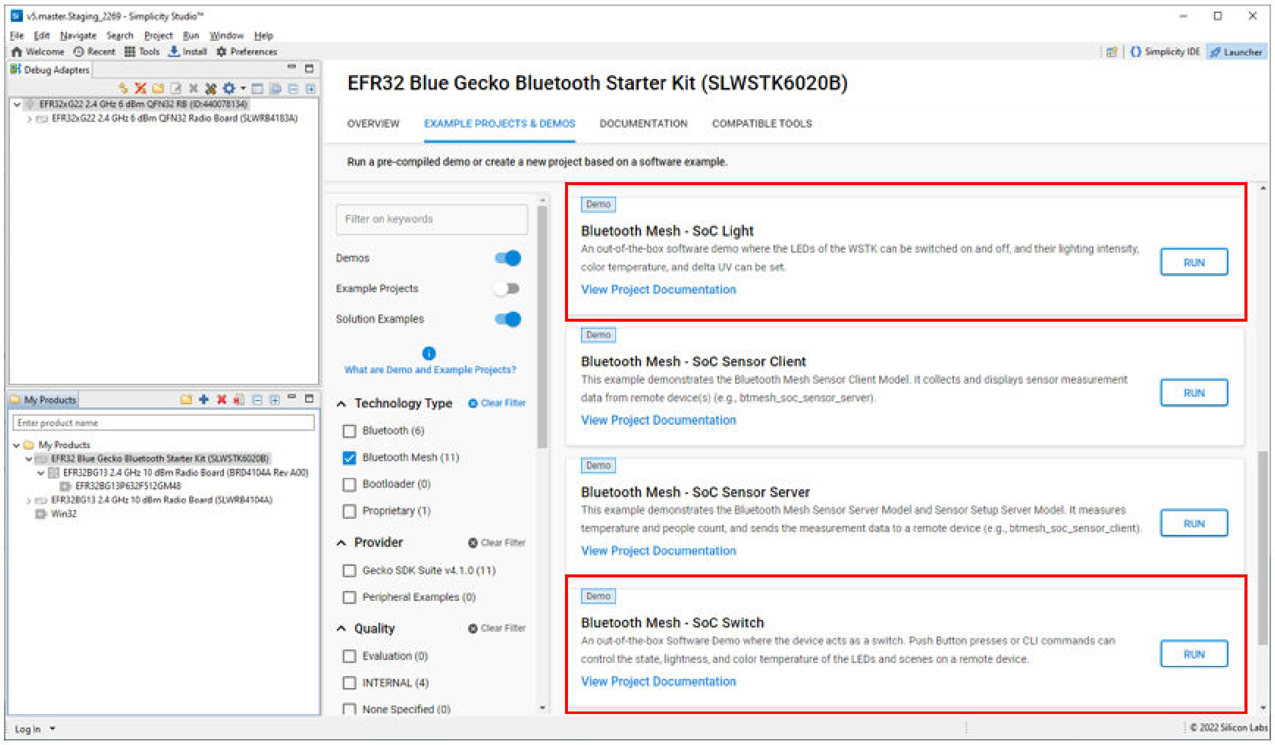
1. Check that the blue "USB Connection Indicator" LED (next to “J-Link USB”) turns on or starts blinking.
2. Check that the mainboard LCD display turns on and displays a Silicon Labs logo.

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

4.2 Install the Demonstration Firmware

When the devices are connected to your PC with a USB cable, you can see three devices listed in the **Debug Adapters** view 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 **Bluetooth Mesh – SoC Light** and one device with **Bluetooth Mesh – SoC Switch**. Go to the Example Projects & Demos tab. Filter as desired to show the demos. Click the target device in the Debug Adapters view and click [**RUN**] next to the desired demo.


v5.master.Staging_2269 - Simplicity Studio™
File Edit Navigate Search Project Run Window Help
Welcome Recent Tools Install Preferences
Simplicity IDE Launcher
Debug Adapters
My Products
Log In
OVERVIEW
EXAMPLE PROJECTS & DEMOS
DOCUMENTATION
COMPATIBLE TOOLS
Filter on keywords
What are Demo and Example Projects?
Clear Filter
Demo
RUN
View Project Documentation
© 2022 Silicon Labs

EFR32xG22 2.4 GHz 6 dBm QFN32 RB (ID:440078134)
EFR32xG22 2.4 GHz 6 dBm QFN32 Radio Board (SLWRB4183A)

Enter product name
My Products
EFR32 Blue Gecko Bluetooth Starter Kit (SLWSTK6020B)
EFR32BG13 2.4 GHz 10 dBm Radio Board (BRD4104A Rev A00)
EFR32BG13P632F512GM48
EFR32BG13 2.4 GHz 10 dBm Radio Board (SLWRB4104A)
Win32

EFR32 Blue Gecko Bluetooth Starter Kit (SLWSTK6020B)

Run a pre-compiled demo or create a new project based on a software example.

Demos: on
Example Projects: off
Solution Examples: on

Technology Type
- [ ] Bluetooth (6)
- [x] Bluetooth Mesh (11)
- [ ] Bootloader (0)
- [ ] Proprietary (1)

Provider
- [ ] Gecko SDK Suite v4.1.0 (11)
- [ ] Peripheral Examples (0)

Quality
- [ ] Evaluation (0)
- [ ] INTERNAL (4)
- [ ] None Specified (0)

Bluetooth Mesh - SoC Light
An out-of-the-box software demo where the LEDs of the WSTK can be switched on and off, and their lighting intensity, color temperature, and delta UV can be set.

Bluetooth Mesh - SoC Sensor Client
This example demonstrates the Bluetooth Mesh Sensor Client Model. It collects and displays sensor measurement data from remote device(s) (e.g., btmesh_soc_sensor_server).

Bluetooth Mesh - SoC Sensor Server
This example demonstrates the Bluetooth Mesh Sensor Server Model and Sensor Setup Server Model. It measures temperature and people count, and sends the measurement data to a remote device (e.g., btmesh_soc_sensor_client).

Bluetooth Mesh - SoC Switch
An out-of-the-box Software Demo where the device acts as a switch. Push Button presses or CLI commands can control the state, lightness, and color temperature of the LEDs and scenes on a remote device.

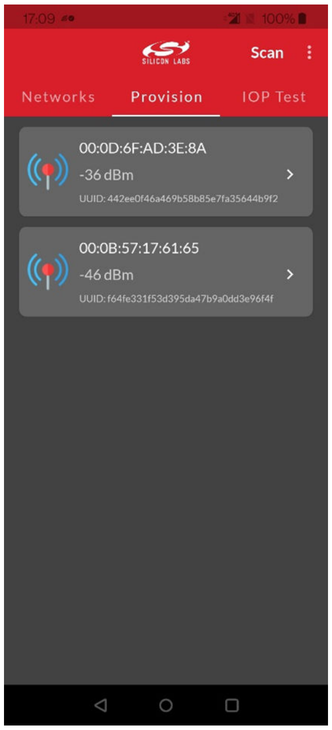
4.3 Use the Demo with an Android Smartphone

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.

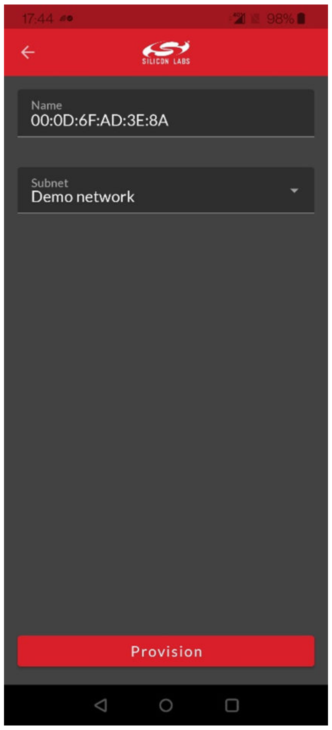
Follow the procedures below to set up and use the demonstration.

1. Go to provisioning view and tap scan to 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.



00:0D:6F:AD:3E:8A
-36 dBm
UUID: 442ee0f46a469b58b85e7fa35644b9f2

00:0B:57:17:61:65
-46 dBm
UUID: f64fe331f53d395da47b9a0dd3e96f4f

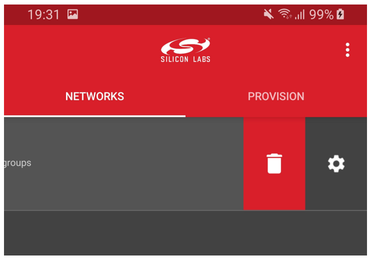


Name
00:0D:6F:AD:3E:8A

Subnet
Demo network

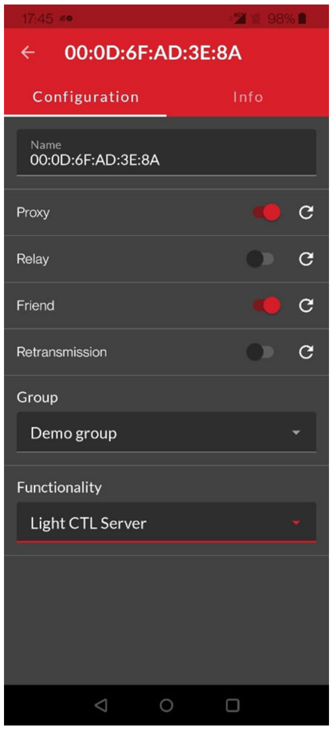
Note: The Android application has a pre-generated network and group, but you can add more groups to the application if you like.

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 device:

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.



00:0D:6F:AD:3E:8A

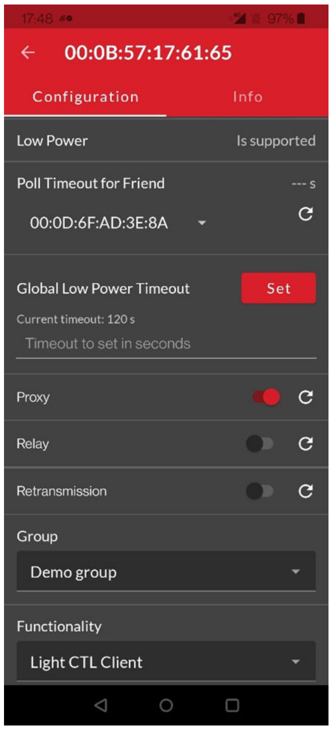
Configuration | Info

Name
00:0D:6F:AD:3E:8A

Proxy
Relay
Friend
Retransmission

Group
Demo group

Functionality
Light CTL Server



00:0B:57:17:61:65

Configuration | Info

Low Power — Is supported

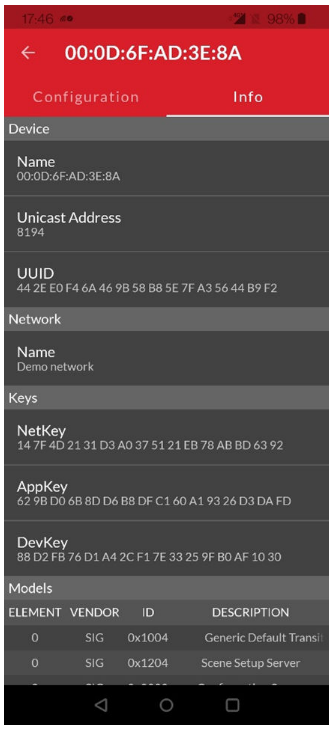
Poll Timeout for Friend — --- s
00:0D:6F:AD:3E:8A

Global Low Power Timeout — Set
Current timeout: 120 s
Timeout to set in seconds

Proxy
Relay
Retransmission

Group
Demo group

Functionality
Light CTL Client



00:0D:6F:AD:3E:8A

Configuration | Info

Device

Name
00:0D:6F:AD:3E:8A

Unicast Address
8194

UUID
44 2E E0 F4 6A 46 9B 58 B8 5E 7F A3 56 44 B9 F2

Network

Name
Demo network

Keys

NetKey
14 7F 4D 21 31 D3 A0 37 51 21 EB 78 AB BD 63 92

AppKey
62 9B D0 6B 8D D6 B8 DF C1 60 A1 93 26 D3 DA FD

DevKey
88 D2 FB 76 D1 A4 2C F1 7E 33 25 9F B0 AF 10 30

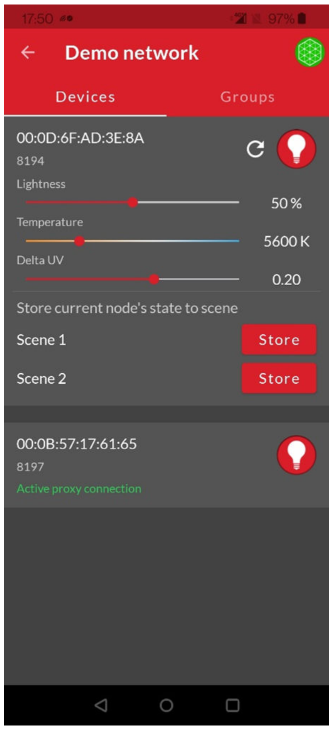
Models

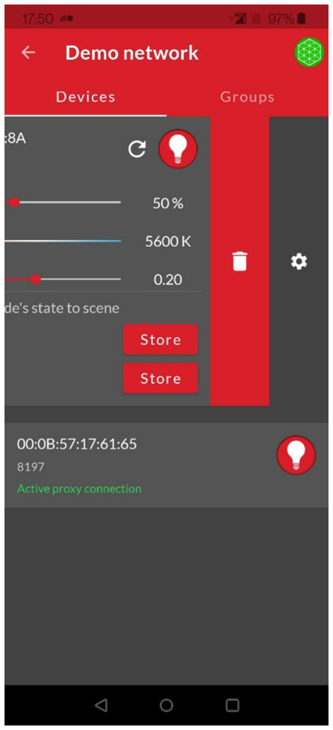
ELEMENT	VENDOR	ID	DESCRIPTION
0	SIG	0x1004	Generic Default Transit
0	SIG	0x1204	Scene Setup Server

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

On devices with only one button, such as Thunderboard BG22:

- Short press: Decrease Light Lightness by 10%, wraps back to 100% after 0%
- Medium press: Decrease CTL (temperature) value, wraps back to maximum after minimum
- Long press: Toggle sending Off message and On message
- Very long press (5 seconds or more): Recall scene 1

Button presses during startup (power-on or reset) execute the following actions:

- On devices with two buttons: PB0 = Full factory reset; PB1 = Node reset
- On devices with one button: PB0 = Full factory reset

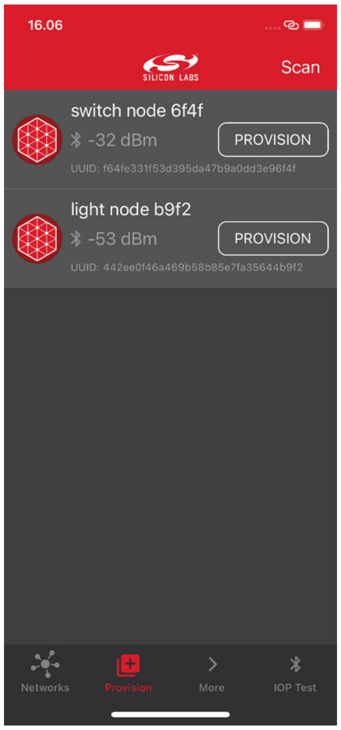
4.4 Use the Demo with an iOS Smartphone

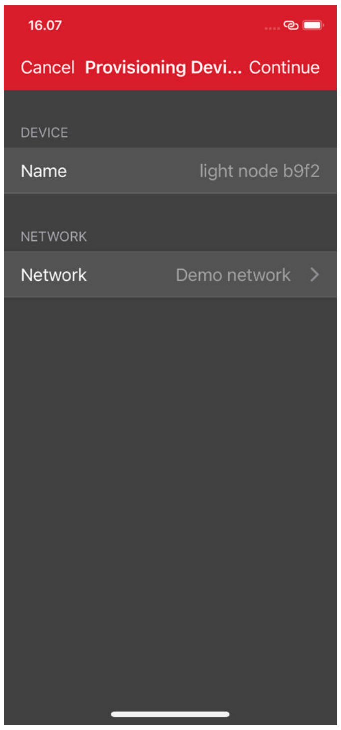
Make sure that all three devices have the status of **"unprovisioned"** on the device LCD screen before starting with the Mobile App.

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

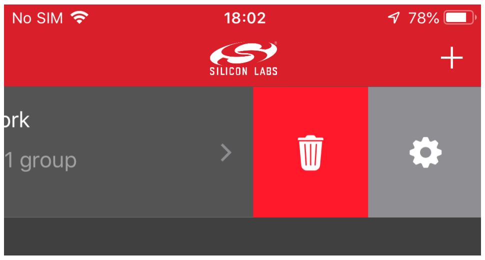
Follow the procedures below to set up and use the demonstration.

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.


16.06
Scan
switch node 6f4f
-32 dBm
PROVISION
UUID: f64fe331f53d395da47b9a0dd3e96f4f
light node b9f2
-53 dBm
PROVISION
UUID: 442ee0f46a469b58b85e7fa35644b9f2

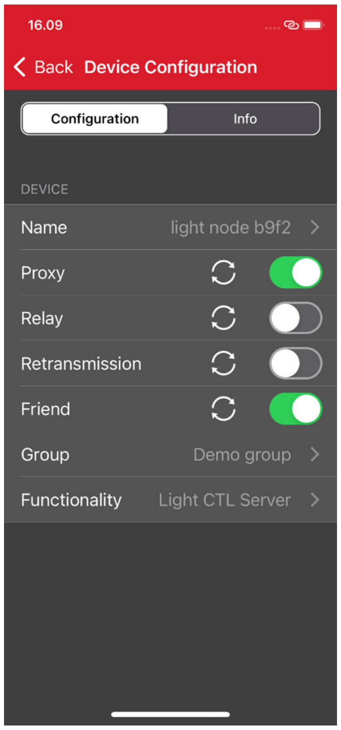

16.07
Cancel Provisioning Devi... Continue
DEVICE
Name light node b9f2
NETWORK
Network Demo network

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:

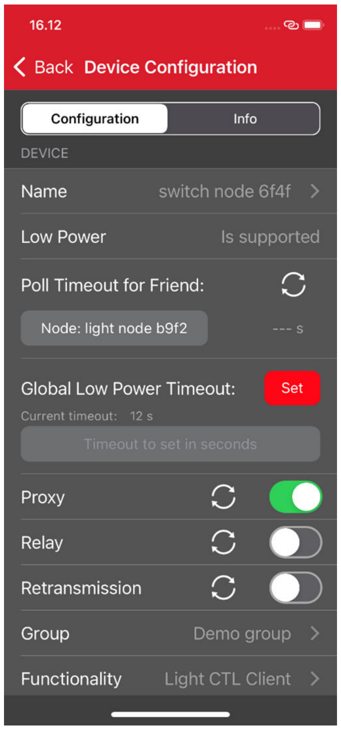
1. During provisioning select the network you want to add the device to.
2. During configuration select the Bluetooth mesh features (proxy, relay, low power and friend) that you want to enable.
    a. Notice that if you disable proxy, the node can no longer be directly accessed over GATT.
3. Select the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



16.09
Back Device Configuration
Configuration | Info

DEVICE

Name	light node b9f2
Proxy	on
Relay	off
Retransmission	off
Friend	on
Group	Demo group
Functionality	Light CTL Server



16.12
Back Device Configuration
Configuration | Info

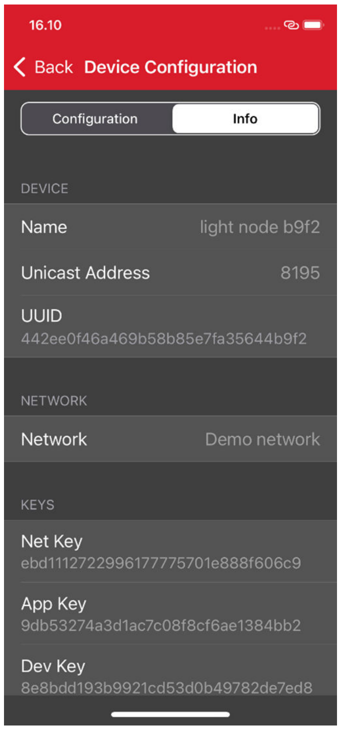
DEVICE

Name	switch node 6f4f
Low Power	Is supported

Poll Timeout for Friend:
Node: light node b9f2 --- s

Global Low Power Timeout: Set
Current timeout: 12 s
Timeout to set in seconds

Proxy	on
Relay	off
Retransmission	off
Group	Demo group
Functionality	Light CTL Client



16.10
Back Device Configuration
Configuration | Info

DEVICE

Name	light node b9f2
Unicast Address	8195
UUID	442ee0f46a469b58b85e7fa35644b9f2

NETWORK

Network	Demo network

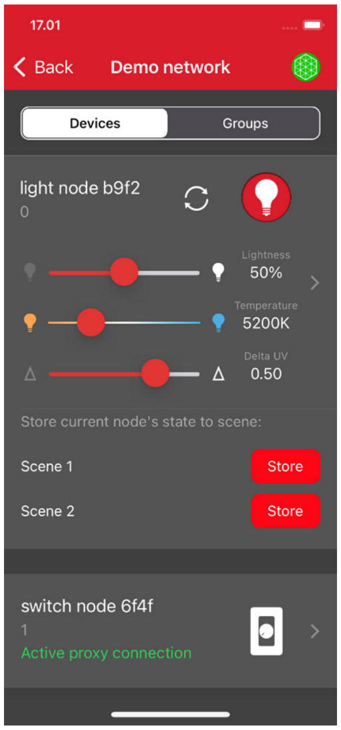
KEYS

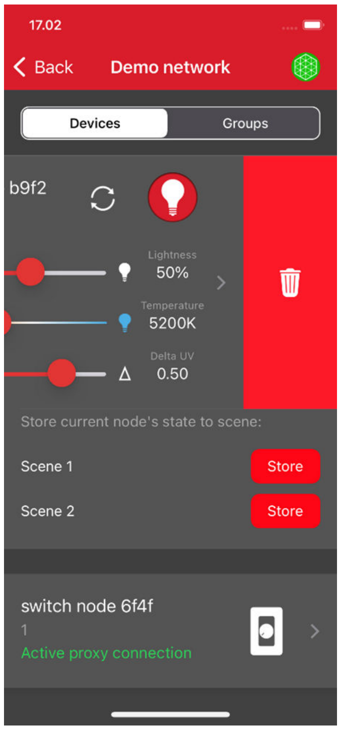
Net Key	ebd1112722996177775701e888f606c9
App Key	9db53274a3d1ac7c08f8cf6ae1384bb2
Dev Key	8e8bdd193b9921cd53d0b49782de7ed8

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

On devices with only one button, such as Thunderboard BG22:

- Short press: Decrease Light Lightness by 10%, wraps back to 100% after 0%
- Medium press: Decrease CTL (temperature) value, wraps back to maximum after minimum
- Long press: Toggle sending Off message and On message
- Very long press (5 seconds or more): Recall scene 1

Button presses during startup (power-on or reset) execute the following actions:

- On devices with two buttons: PB0 = Full factory reset; PB1 = Node reset
- On devices with one button: PB0 = Full factory reset

5. Getting Started with Application Development

The most common starting point for application development is the **Bluetooth Mesh - SoC Empty** example. This example demonstrates the bare minimum needed for a Bluetooth mesh C application that allows Over-the-Air Device Firmware Upgrading (OTA DFU). The application starts Unprovisioned Device Beaconing after boot waiting to be provisioned to a Mesh Network.

Note: All Bluetooth mesh devices must be loaded with the Gecko Bootloader as well as the application. While you are getting started, the easiest way to do this is to load any of the precompiled demo images that come with the bootloader configured as part of the image. When you flash your application it overwrites the demo application, but the bootloader remains. Subsequently you may wish to build your own bootloader, as described in *UG266: Silicon Labs Gecko Bootloader User's Guide for GSDK 3.2 or Lower* and *UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 or Higher*. If the application requires a specific bootloader type, it is noted in the example description in section 3. About Demos and Examples.

New Project creation is done through three dialogs:

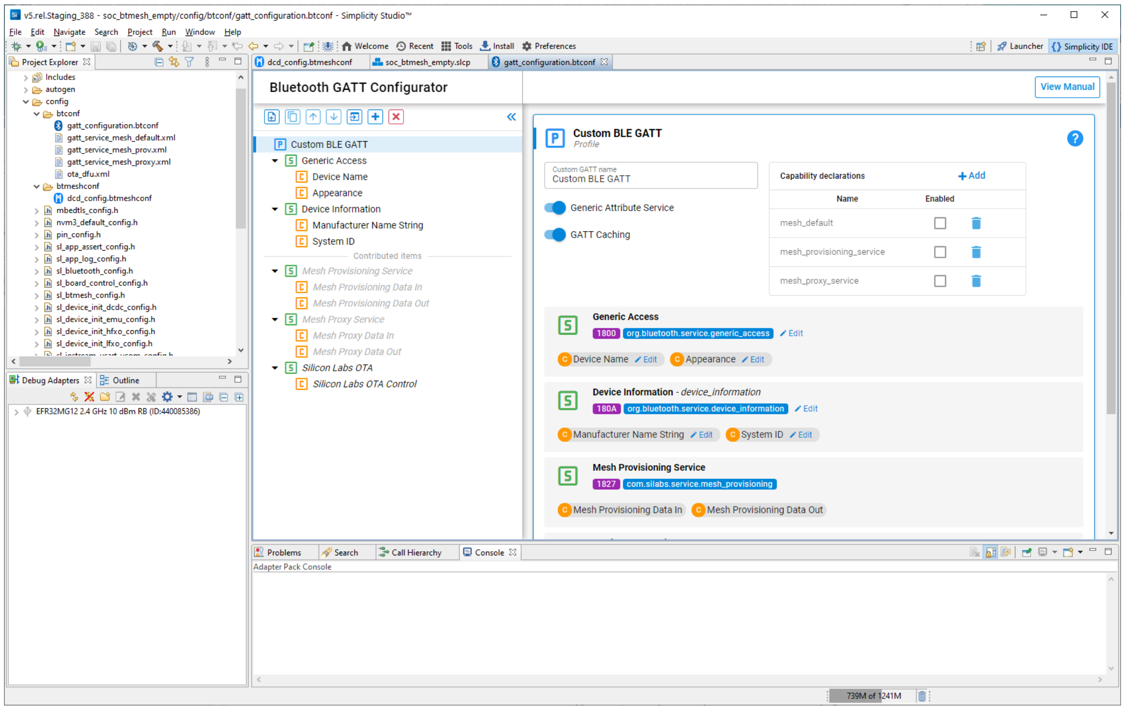
- Target, SDK, and Toolchain
- Examples
- Configuration

You can start a project from different locations in the Launcher Perspective, as described in the Simplicity Studio 5 User's Guide. While you are getting started, we suggest starting from the File menu, as that takes you through all three of the above dialogs.

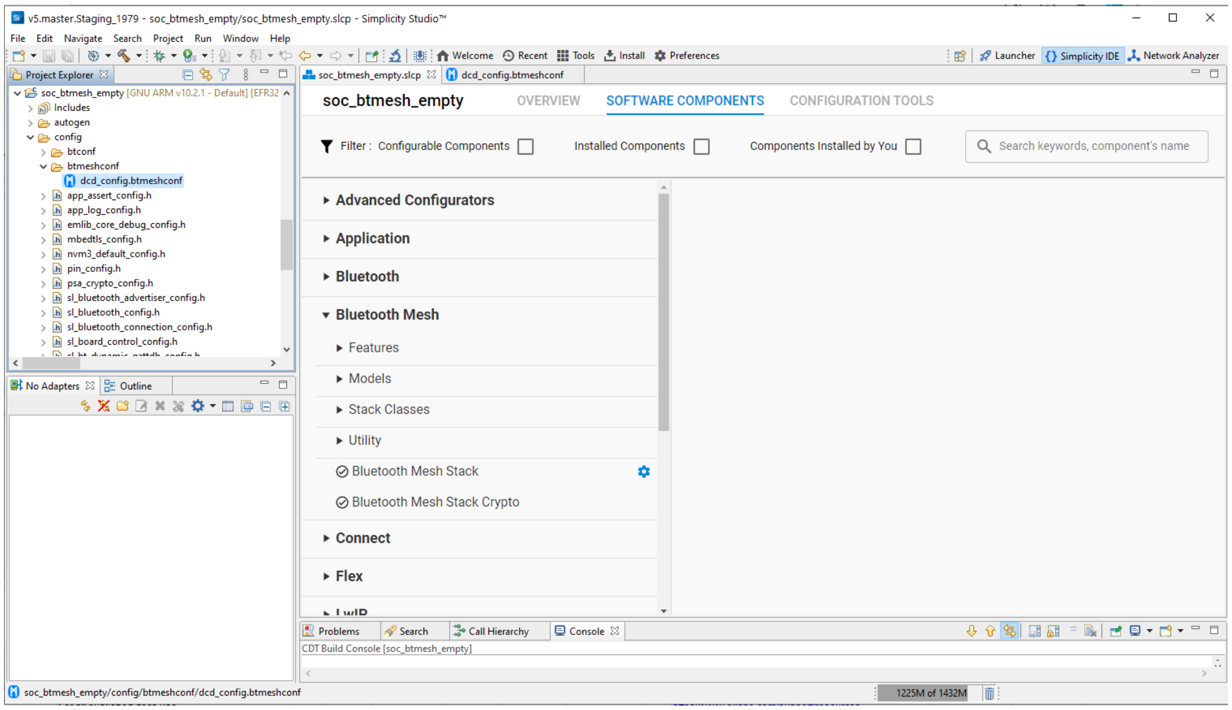
1. Select New >> Silicon Labs Project Wizard.
2. Review your SDK and toolchain. If you wish to use IAR instead of GCC, be sure to change it here. Once you have created a project it is difficult to change toolchains. Click [**NEXT**].
3. On the Example Project Selection dialog, filter on Bluetooth Mesh and select **Bluetooth Mesh - SoC Empty**. Click [**NEXT**].
4. On the Project Configuration dialog, rename your project if you wish. Note that if you change any linked resource, it is changed for any other project that references it. While you are getting started the default choice to include project files but link to the SDK is best. Click [**FINISH**].

When you create a Bluetooth mesh project, three tabs open automatically: the GATT Configurator (gatt_configuration.btconf), .the slcp or Project Configurator (<projectname>.slcp), and the Mesh Configurator (dcd_config.btmeshconf). If the example has documentation, the project opens on a readme tab. Note that a Simplicity IDE perspective control is now included in the upper right of the screen.

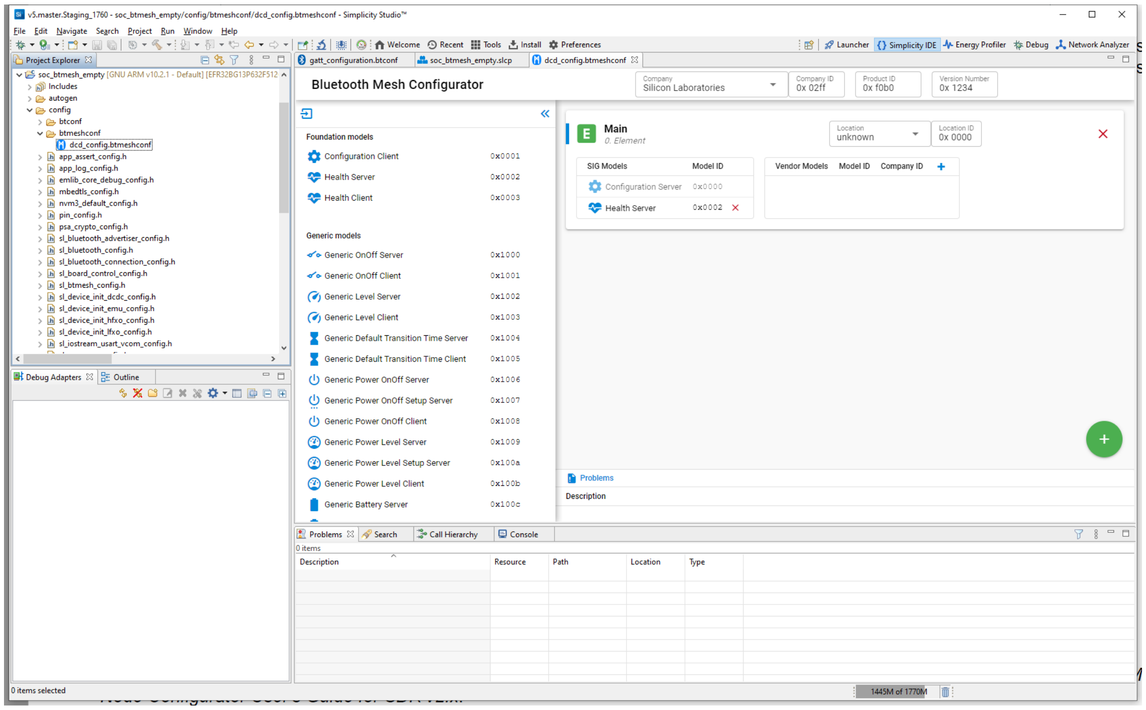
GATT configuration is the same for both Bluetooth and Bluetooth mesh projects. *UG438: GATT Configurator User's Guide for Bluetooth SDK v3.x* describes how to configure the GATT database.


v5.rel.Staging_388 - soc_btmesh_empty/config/btconf/gatt_configuration.btconf - Simplicity Studio™
File Edit Navigate Search Project Run Window Help
Welcome Recent Tools Install Preferences
Launcher Simplicity IDE
Project Explorer
Includes
autogen
config
btconf
gatt_configuration.btconf
gatt_service_mesh_default.xml
gatt_service_mesh_prov.xml
gatt_service_mesh_proxy.xml
ota_dfu.xml
btmeshconf
dcd_config.btmeshconf
mbedtls_config.h
nvm3_default_config.h
pin_config.h
sl_app_assert_config.h
sl_app_log_config.h
sl_bluetooth_config.h
sl_board_control_config.h
sl_btmesh_config.h
sl_device_init_dcdc_config.h
sl_device_init_emu_config.h
sl_device_init_hfxo_config.h
sl_device_init_lfxo_config.h
dcd_config.btmeshconf
soc_btmesh_empty.slcp
gatt_configuration.btconf
Bluetooth GATT Configurator
View Manual
Custom BLE GATT
Generic Access
Device Name
Appearance
Device Information
Manufacturer Name String
System ID
Contributed items
Mesh Provisioning Service
Mesh Provisioning Data In
Mesh Provisioning Data Out
Mesh Proxy Service
Mesh Proxy Data In
Mesh Proxy Data Out
Silicon Labs OTA
Silicon Labs OTA Control
Custom BLE GATT
Profile
Custom GATT name
Custom BLE GATT
Capability declarations
+Add
Name
Enabled
mesh_default
mesh_provisioning_service
mesh_proxy_service
Generic Attribute Service
GATT Caching
Generic Access
1800
org.bluetooth.service.generic_access
Edit
Device Name
Appearance
Device Information - device_information
180A
org.bluetooth.service.device_information
Manufacturer Name String
System ID
Mesh Provisioning Service
1827
com.silabs.service.mesh_provisioning
Mesh Provisioning Data In
Mesh Provisioning Data Out
Debug Adapters
Outline
EFR32MG12 2.4 GHz 10 dBm RB (ID:440085386)
Problems
Search
Call Hierarchy
Console
Adapter Pack Console
739M of 1241M

The Project Configurator and its associated Component Editor provide access to components. All the Bluetooth mesh functionality is provided as components. You can customize projects by installing or uninstalling the components based on the use cases and requirements, and then configuring installed components using the Component Editor. The Bluetooth Mesh Component group shows the components specific to Bluetooth mesh projects.

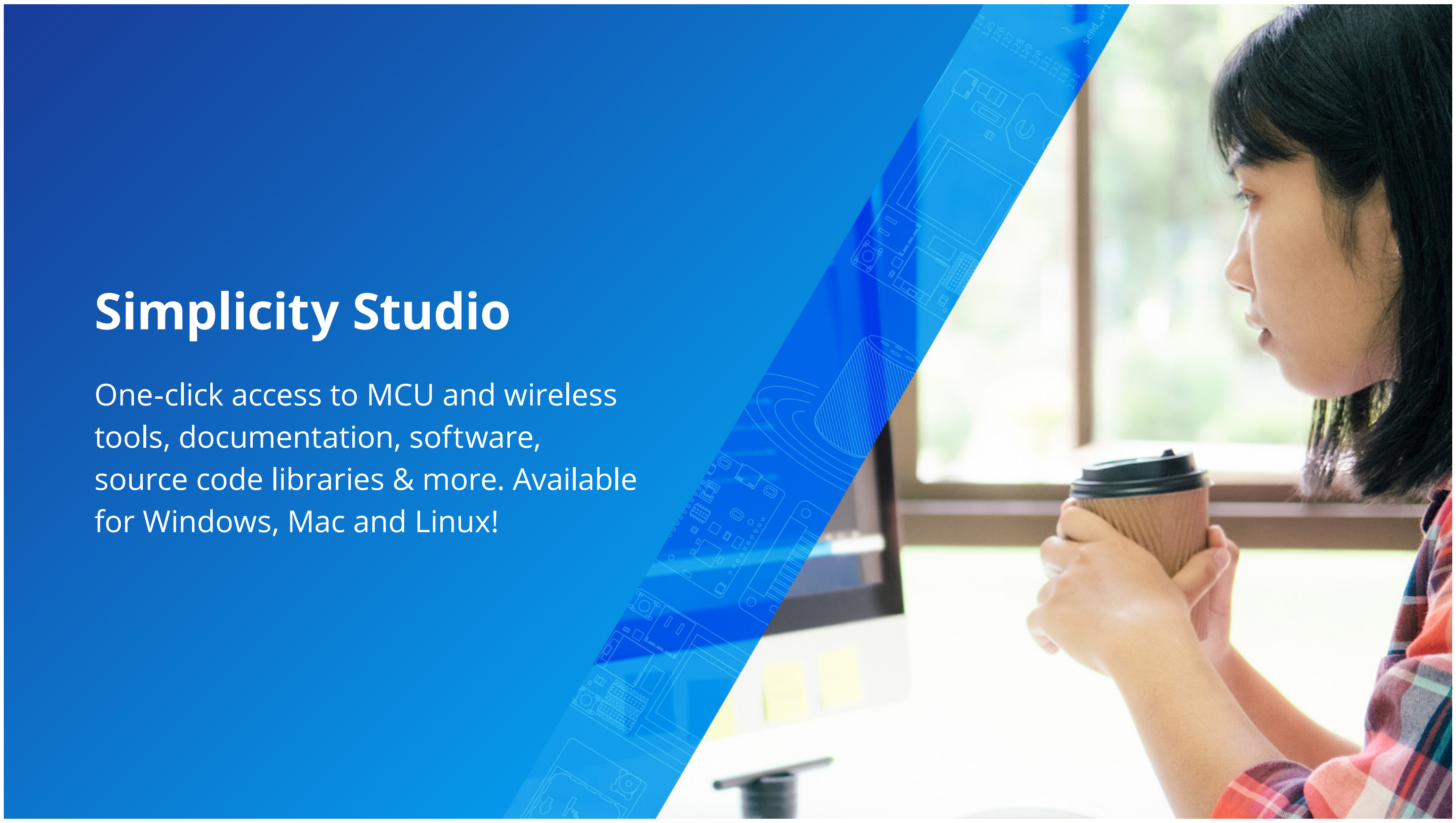

v5.master.Staging_1979 - soc_btmesh_empty/soc_btmesh_empty.slcp - Simplicity Studio™
soc_btmesh_empty
OVERVIEW
SOFTWARE COMPONENTS
CONFIGURATION TOOLS
Filter : Configurable Components
Installed Components
Components Installed by You
Search keywords, component's name
Advanced Configurators
Application
Bluetooth
Bluetooth Mesh
Features
Models
Stack Classes
Utility
Bluetooth Mesh Stack
Bluetooth Mesh Stack Crypto
Connect
Flex
Project Explorer
dcd_config.btmeshconf
CDT Build Console [soc_btmesh_empty]
soc_btmesh_empty/config/btmeshconf/dcd_config.btmeshconf

The Bluetooth Mesh Configurator provides access to Device Composition Data (DCD). This contains information about a Bluetooth mesh node, the elements it includes, and the supported models. DCD exposes the node information to a configuration client so that it knows the potential functionalities the node supports and, based on that, can configure the node.


v5.master.Staging_1760 - soc_btmesh_empty/config/btmeshconf/dcd_config.btmeshconf - Simplicity Studio™
Bluetooth Mesh Configurator
Company
Silicon Laboratories
Company ID
0x 02ff
Product ID
0x f0b0
Version Number
0x 1234
Foundation models
Configuration Client 0x0001
Health Server 0x0002
Health Client 0x0003
Generic models
Generic OnOff Server 0x1000
Generic OnOff Client 0x1001
Generic Level Server 0x1002
Generic Level Client 0x1003
Generic Default Transition Time Server 0x1004
Generic Default Transition Time Client 0x1005
Generic Power OnOff Server 0x1006
Generic Power OnOff Setup Server 0x1007
Generic Power OnOff Client 0x1008
Generic Power Level Server 0x1009
Generic Power Level Setup Server 0x100a
Generic Power Level Client 0x100b
Generic Battery Server 0x100c
Main
0. Element
Location
unknown
Location ID
0x 0000
SIG Models
Model ID
Configuration Server 0x0000
Health Server 0x0002
Vendor Models
Model ID
Company ID
Problems
Description

For more details on node configuration using the Project Configurator and Bluetooth Mesh Configurator, see *UG472: Bluetooth® Mesh Node Configurator User's Guide for SDK v2.x and Higher*.

To build and debug your project click Debug (bug icon) in the upper left corner of the Simplicity IDE perspective. It will build and download your project, and open the Debug perspective. Click Play to start running you project on the device.


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

www.silabs.com