QSG155: Using the Silicon Labs Dynamic Multiprotocol Demonstration Applications

This document shows how to use the dynamic protocol lighting demonstrations. Two demonstrations may be used, one incorporating Zigbee 3.0 and Bluetooth functionality and the other RAIL and Bluetooth functionality. RAIL (Radio Abstraction Interface Layer) is Silicon Labs' intuitive, easily-customizable radio interface layer that supports proprietary or standards-based wireless protocols. In both demonstrations, a dynamic multiprotocol light application can be controlled either from a protocol-specific switch application or from a Bluetooth-enabled smartphone app.

KEY POINTS

- Prerequisite for the demos
- Demo firmware installation
- Instructions for using the demos

1 Prerequisites

The Silicon Labs dynamic multiprotocol demonstrations are designed to illustrate dynamic multiprotocol operation without any need to configure or compile software. This document assumes that you have already obtained your hardware and installed the required software. You will need an EFR32 Mighty Gecko Wireless Starter Kit (SLWSTK6000B). For the demo you will use two WSTK main boards with demonstration-specific radio boards installed. If you already have the WSTK main boards, you can purchase the required radio boards here. Installing Simplicity Studio and the required SDKs are described in their respective quick start guides, listed below.

- Zigbee/Bluetooth Demonstration
  - Radio boards: EFR32MG12 SoC radio boards (SLWRB4162A).
  - SDK: Version 6.0.0.0 or higher of the EmberZNet SDK, as described in *QSG106: Getting Started with EmberZNet PRO*.
- RAIL/Bluetooth Demonstration
  - Radio boards: EFR32MG12 2.4 GHz/915 MHz dual band radio boards (SLWRB4164A)
  - SDKs: Version 2.8.0.0 or higher of the Bluetooth SDK (*QSG139: Bluetooth Development with Simplicity Studio*), which contains the light application, and version 2.2.0 or higher of the Flex SDK (*QSG138: Getting Started with the Silicon Labs Flex SDK for the Wireless Gecko (EFR32™) Portfolio*), which contains the RAIL switch application.

Note: This document describes using the precompiled images supplied with the SDKs. See section **Next Steps** for information on continuing with customizing the building the source code for these demonstration images.

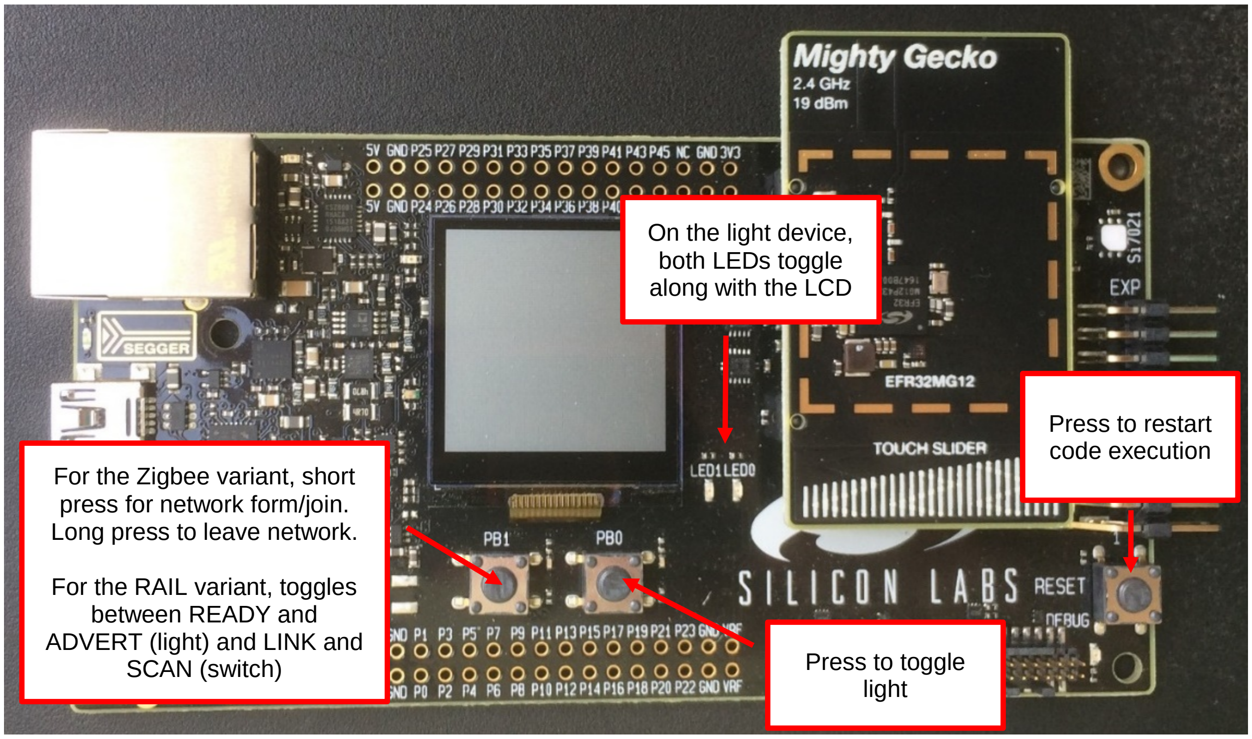
You should also download the **Wireless Gecko** by Silicon Labs smartphone application for Android or iOS. Note that there are several Silicon Labs apps - be sure to download **Wireless Gecko**, not Blue Gecko.

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

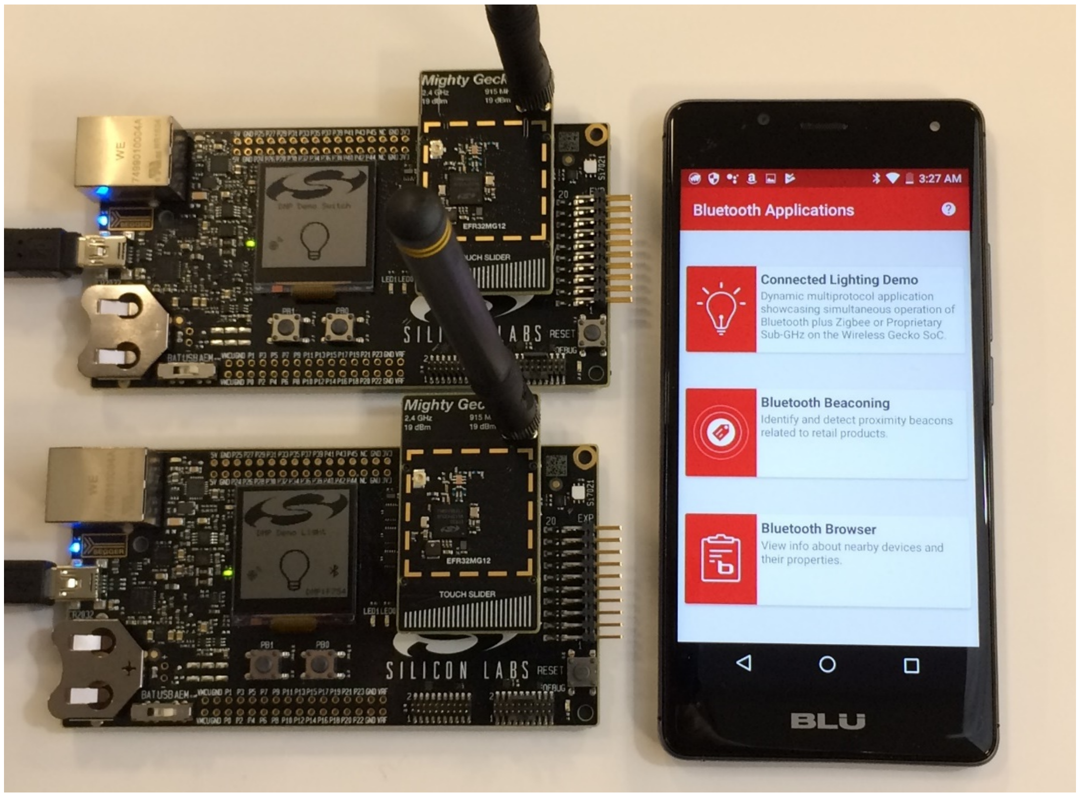
Use the support portal to contact Customer Support for any questions you might have about the demonstrations. You can access the Silicon Labs support portal at https://www.silabs.com/support through Simplicity Studio Resources. Click the “Email-Support” link and log in with your self-registered credentials.

2 Demonstration Devices

The following features on the WSTK are used in the demos.

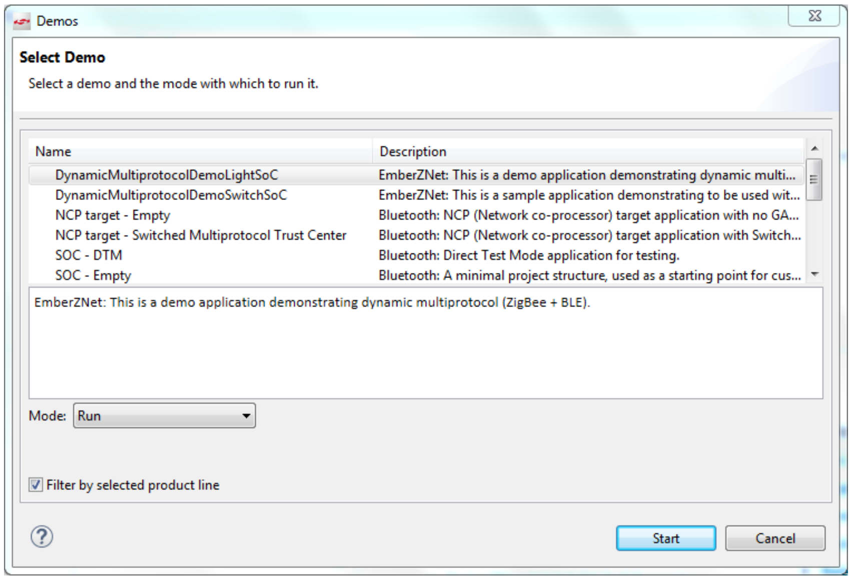

On the light device, both LEDs toggle along with the LCD
Press to restart code execution
For the Zigbee variant, short press for network form/join. Long press to leave network.
For the RAIL variant, toggles between READY and ADVERT (light) and LINK and SCAN (switch)
Press to toggle light
Mighty Gecko
PB1
PB0
LED1 LED0
RESET
TOUCH SLIDER
EFR32MG12
SILICON LABS

The following is a typical demo setup with two WSTKs and a smart phone (the sub-Gigahertz setup for RAIL/Bluetooth shown).


Bluetooth Applications
Connected Lighting Demo
Dynamic multiprotocol application showcasing simultaneous operation of Bluetooth plus Zigbee or Proprietary Sub-GHz on the Wireless Gecko SoC.
Bluetooth Beaconing
Identify and detect proximity beacons related to retail products.
Bluetooth Browser
View info about nearby devices and their properties.
BLU

3 Install the Demonstration Applications

Open Simplicity Studio. In general, to load a demo, click a device in the Devices pane, then click the demo. The Select Demo dialog opens. In the **Mode** drop-down, select **Run**. Click **Start**. The demo software downloads automatically to the selected device.



Select Demo

Select a demo and the mode with which to run it.

Name	Description
DynamicMultiprotocolDemoLightSoC	EmberZNet: This is a demo application demonstrating dynamic multi...
DynamicMultiprotocolDemoSwitchSoC	EmberZNet: This is a sample application demonstrating to be used wit...
NCP target - Empty	Bluetooth: NCP (Network co-processor) target application with no GA...
NCP target - Switched Multiprotocol Trust Center	Bluetooth: NCP (Network co-processor) target application with Switch...
SOC - DTM	Bluetooth: Direct Test Mode application for testing.
SOC - Empty	Bluetooth: A minimal project structure, used as a starting point for cus...

EmberZNet: This is a demo application demonstrating dynamic multiprotocol (ZigBee + BLE).

Mode: Run

☑ Filter by selected product line

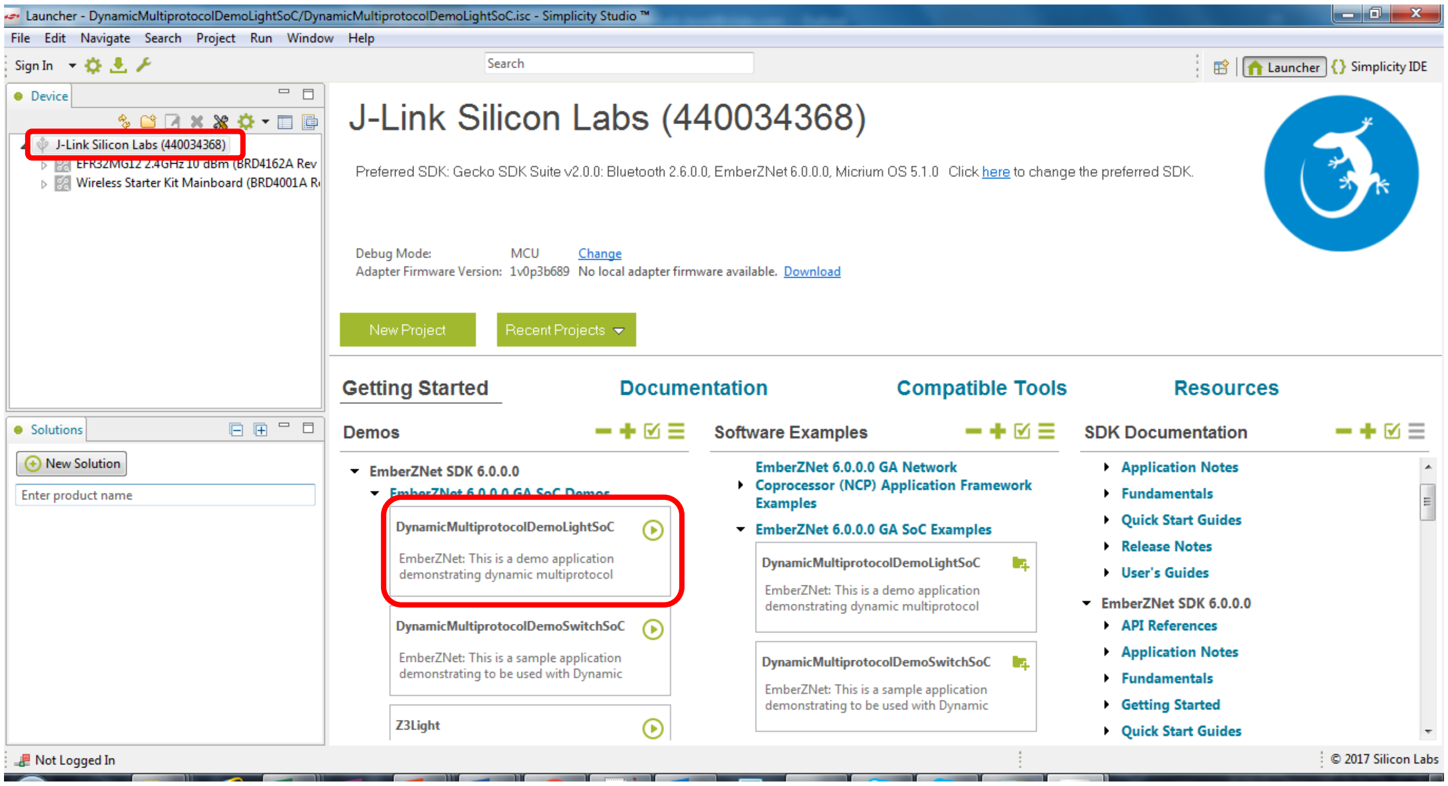
RAIL/Bluetooth

To run the RAIL/Bluetooth demo, in the Flex SDK, load one device with **RAIL: Switc**h. Then go to the Bluetooth SDK and load the other device with **SOC - Light -RAIL - DMP**.

Note: The Light application in the Flex SDK is not the multiprotocol application.

Zigbee/Bluetooth

To run the Zigbee/Bluetooth demo, load one device with **DynamicMultiprotocolDemoLightSoC** and one device with **DynamicMulti-protocolDemoSwitchSoC**.


Launcher - DynamicMultiprotocolDemoLightSoC/DynamicMultiprotocolDemoLightSoC.isc - Simplicity Studio ™
File Edit Navigate Search Project Run Window Help
Sign In
Search
Launcher
Simplicity IDE
Device
J-Link Silicon Labs (440034368)
EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev
Wireless Starter Kit Mainboard (BRD4001A R
Solutions
New Solution
Enter product name
Not Logged In
© 2017 Silicon Labs

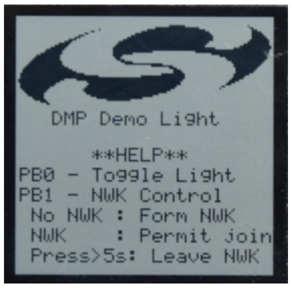
J-Link Silicon Labs (440034368)

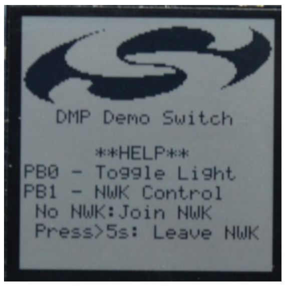
Preferred SDK: Gecko SDK Suite v2.0.0: Bluetooth 2.6.0.0, EmberZNet 6.0.0.0, Micrium OS 5.1.0 Click here to change the preferred SDK.

Debug Mode: MCU Change
Adapter Firmware Version: 1v0p3b689 No local adapter firmware available. Download

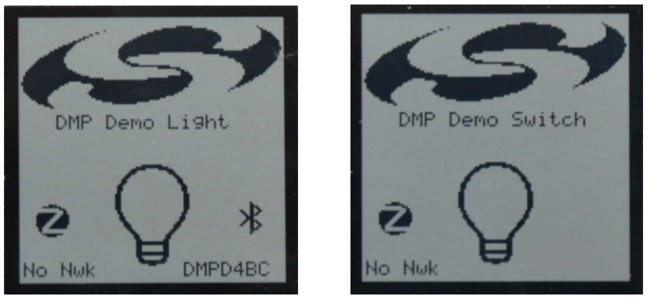
New Project
Recent Projects
Getting Started
Documentation
Compatible Tools
Resources
Demos
EmberZNet SDK 6.0.0.0
EmberZNet 6.0.0.0 GA SoC Demos
DynamicMultiprotocolDemoLightSoC
EmberZNet: This is a demo application demonstrating dynamic multiprotocol
DynamicMultiprotocolDemoSwitchSoC
EmberZNet: This is a sample application demonstrating to be used with Dynamic
Z3Light
Software Examples
EmberZNet 6.0.0.0 GA Network Coprocessor (NCP) Application Framework Examples
EmberZNet 6.0.0.0 GA SoC Examples
DynamicMultiprotocolDemoLightSoC
EmberZNet: This is a demo application demonstrating dynamic multiprotocol
DynamicMultiprotocolDemoSwitchSoC
EmberZNet: This is a sample application demonstrating to be used with Dynamic
SDK Documentation
Application Notes
Fundamentals
Quick Start Guides
Release Notes
User's Guides
EmberZNet SDK 6.0.0.0
API References
Application Notes
Fundamentals
Getting Started
Quick Start Guides

When the Zigbee/Bluetooth demo applications download, they first show a help menu on the LCD, and then show the main display. The help menus are also displayed for approximately 10 seconds if you restart code execution by pressing the Reset button. Note that resetting the application breaks the Bluetooth connection between devices, if one has been formed, and interrupts communication on the device network, if one has been formed. The Help menus are:

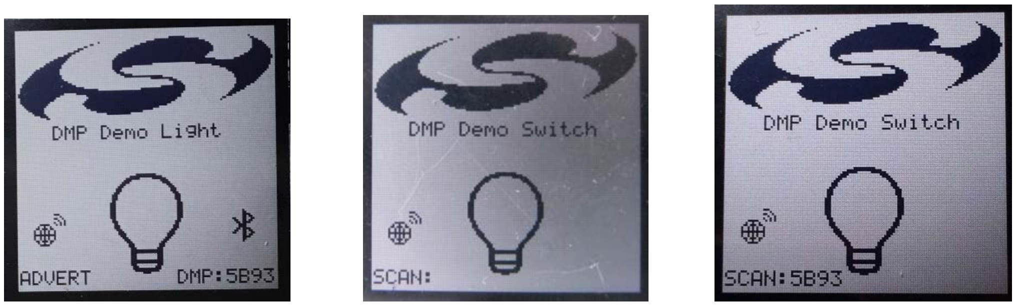




After installation is complete, the LCDs show the application “Light Bulb” display. The RAIL Switch application briefly scans for a nearby switch and then shows the short ID of the switch with the strongest signal. Note the logos of the different protocols.



Bluetooth/Zigbee

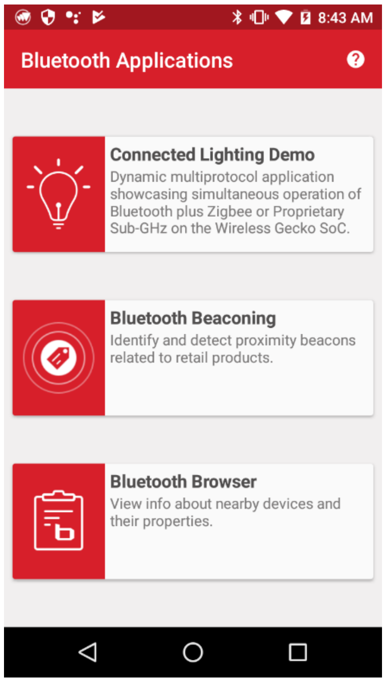


Bluetooth/RAIL

4 Use the Zigbee/Bluetooth Demonstration

4.1 With the Bluetooth Smartphone App Alone

When both devices are ready, open the app and tap **Connected Lighting Demo**.



8:43 AM

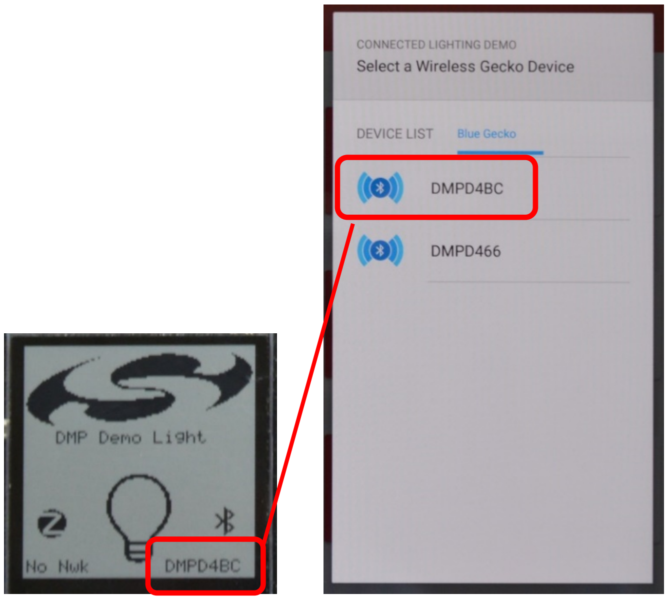
Bluetooth Applications

Connected Lighting Demo
Dynamic multiprotocol application showcasing simultaneous operation of Bluetooth plus Zigbee or Proprietary Sub-GHz on the Wireless Gecko SoC.

Bluetooth Beaconing
Identify and detect proximity beacons related to retail products.

Bluetooth Browser
View info about nearby devices and their properties.

Tap the Bluetooth light device. If you see more than one, tap the one with the matching ID.



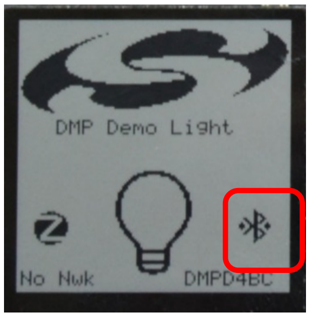
CONNECTED LIGHTING DEMO
Select a Wireless Gecko Device

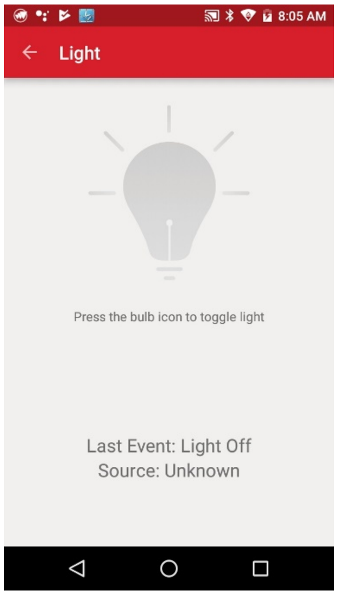
DEVICE LIST   Blue Gecko

DMPD4BC

DMPD466

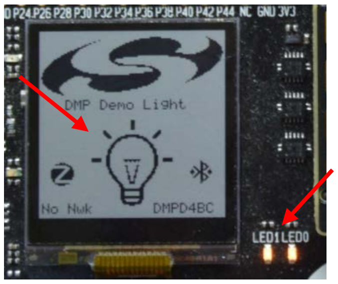
The app display changes to the Light Bulb, and the Bluetooth icon on the light's LCD changes as well, indicating a connection.

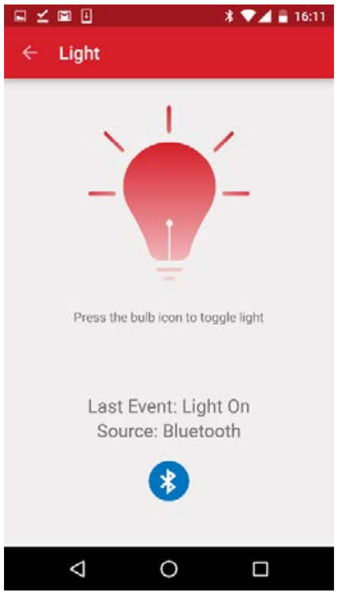




Last Event: Light Off
Source: Unknown

Tap the bulb icon on the smartphone app to toggle the light. The app display, the LCD display, and the LEDs all turn on. The app shows **Last Event: Light On** and **Source: Bluetooth.**

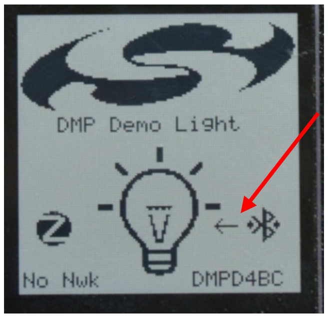




Last Event: Light On
Source: Bluetooth

Tap again to turn the light off. The app shows **Last Event: Light Off** and **Source: Bluetooth**.

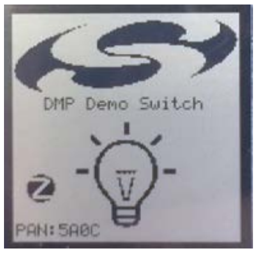
As you toggle the light with the app, an arrow is briefly displayed on the light LCD to indicate that the source of the command is Bluetooth.

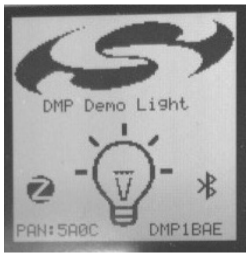


On the light device, press PB0 to toggle the light. The app display now shows **Source: Button.**

4.2 With the Switch

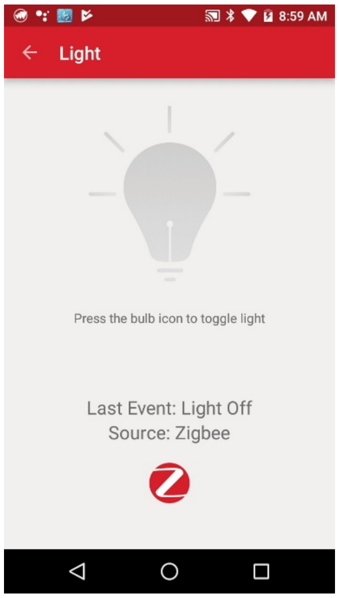
In order to operate the light from the Zigbee switch application, you need to form a Zigbee network. Press PB1 on the light WSTK. The display changes from **No Nwk** to **Forming** and then to a flashing PAN ID. While the PAN ID is flashing the light is in permit join mode. Press PB1 on the switch. The display changes to **Joining** and, after a brief delay, to the network's PAN ID.





Note: If you are in a busy environment, the switch might join to another network. Make sure the PAN IDs on both the light and the switch are the same. If not, press and hold PB1 on the switch for more than 5 seconds to take it off the network. Press PB1 on the light to put it back into permit join mode. Once the PAN ID is flashing, press PB1 on the switch.

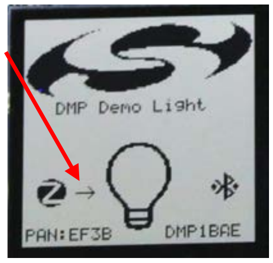
Once the switch is connected, press PB0 on the switch to toggle the light. The app shows **Source: Zigbee**.



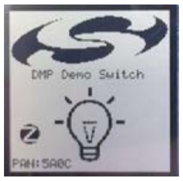
Press the bulb icon to toggle light

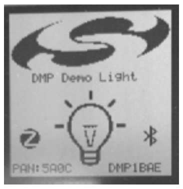
Last Event: Light Off
Source: Zigbee

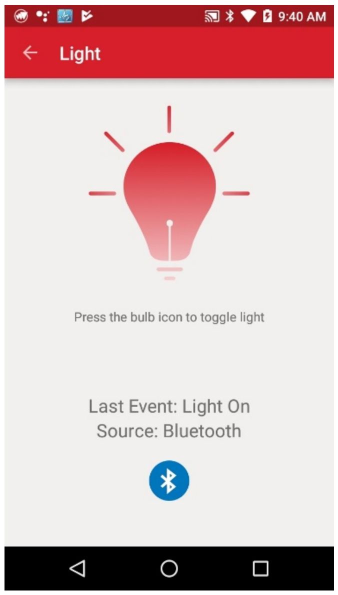
Again, a briefly-displayed arrow shows the source of the command on the Light LCD.



Now if you press the bulb icon on the app, both the light and the switch displays change.







Light

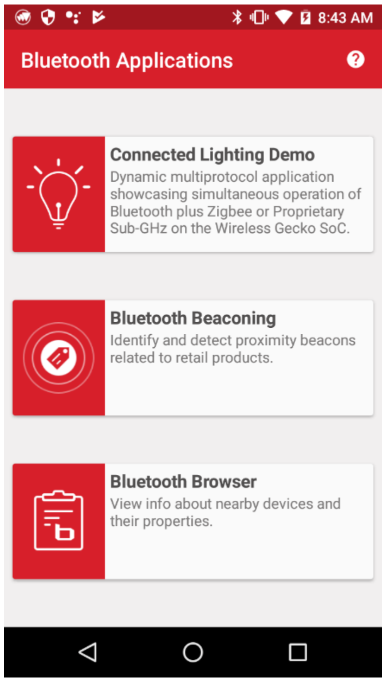
Press the bulb icon to toggle light

Last Event: Light On
Source: Bluetooth

5 Use the RAIL/Bluetooth Demonstration

5.1 With the Bluetooth Smartphone App Alone

When both devices are ready, open the app and tap **Connected Lighting Demo**.



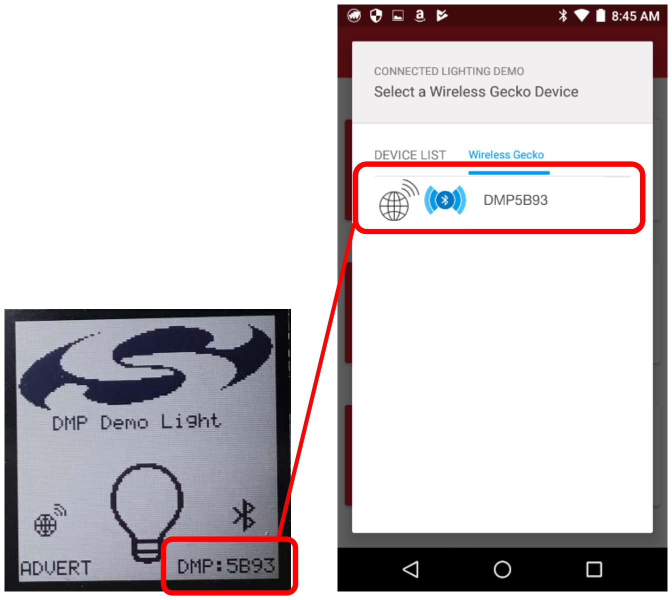
Bluetooth Applications

Connected Lighting Demo
Dynamic multiprotocol application showcasing simultaneous operation of Bluetooth plus Zigbee or Proprietary Sub-GHz on the Wireless Gecko SoC.

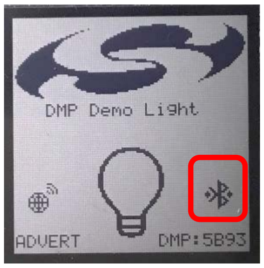
Bluetooth Beaconing
Identify and detect proximity beacons related to retail products.

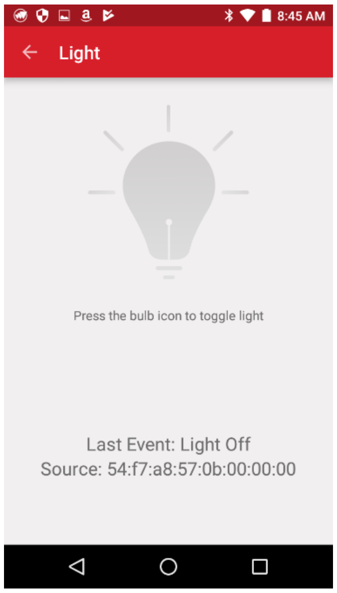
Bluetooth Browser
View info about nearby devices and their properties.

Tap the Bluetooth light device. If you see more than one, tap the one with the matching ID.



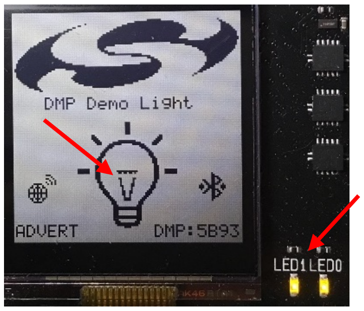
The app display changes to the Light Bulb, and the Bluetooth icon on the light's LCD changes as well, indicating a connection.

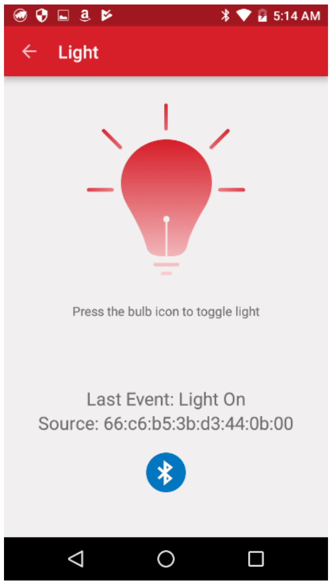




Last Event: Light Off
Source: 54:f7:a8:57:0b:00:00:00

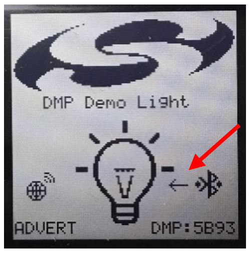
Tap the bulb icon on the smartphone app to toggle the light. The app display, the LCD display, and the LEDs all turn on. The app shows **Last Event: Light On. Source** is the MAC address of the device sending the command, in this case the smartphone.





Last Event: Light On
Source: 66:c6:b5:3b:d3:44:0b:00

Tap again to turn the light off. The app shows **Last Event: Light Off** and **Source:** Smartphone MAC address. As you toggle the light with the app, an arrow is briefly displayed on the light LCD to indicate that the source of the command is Bluetooth.



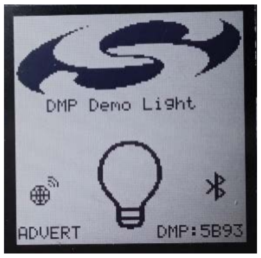
On the light device, press PB0 to toggle the light. The app display now shows the source as the MAC address of the light device.

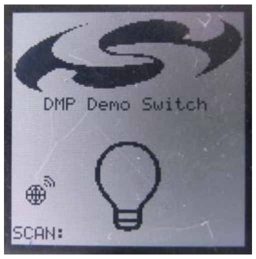
Note that the light demo has two modes, Advertise (ADVERT) and READY. These are for operation by the switch device, as described next. The smartphone app toggles the light regardless of the mode.

5.2 With the Switch

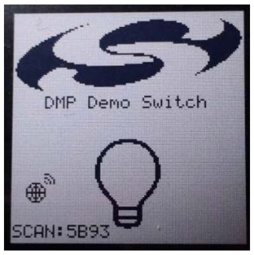
In order to operate the light from the RAIL Switch application, you first need to link the two devices.

The RAIL switch starts in SCAN mode to look for lights in the vicinity. The light starts in ADVERTISE mode and broadcasts its address to nearby RAIL Switches.

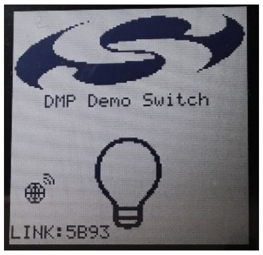




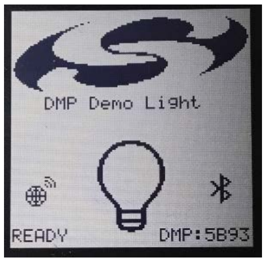
Once a RAIL switch receives a light's advertisement packet it displays its short ID in the bottom left corner. If more lights are advertising at the same time, the switch displays the ID of the Light with the strongest signal:



Press PB1 on the RAIL switch to store the light's ID and put the device into LINK mode:



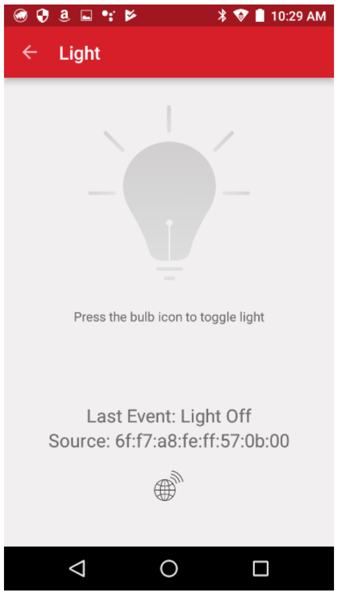
Finally, press PB1 on the Light to place it to READY mode, in which it sends out the status of the light periodically and accepts toggle commands from the linked Switch:



You should now be able to toggle the light using the linked switch.

Note that more than one switch can be linked to a single light, but one switch cannot control more than one light.

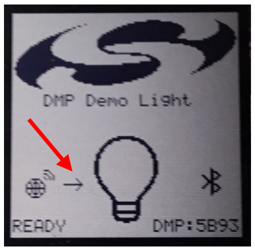
The app shows the source as the MAC address of the switch.



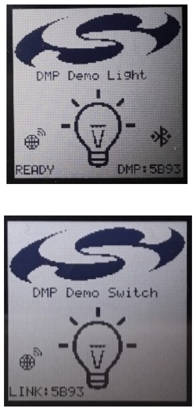
Press the bulb icon to toggle light

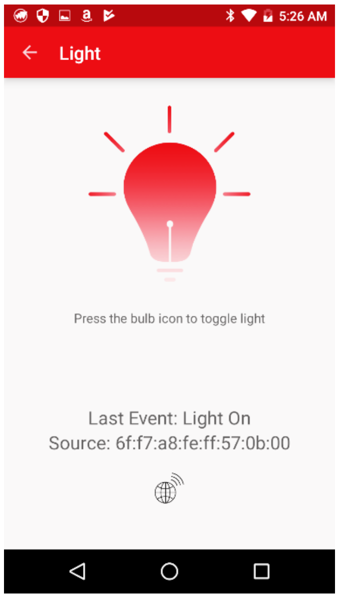
Last Event: Light Off
Source: 6f:f7:a8:fe:ff:57:0b:00

Again, a briefly-displayed arrow shows the source of the command on the Light LCD.



Now if you press the bulb icon on the app, both the light and the switch displays change.





Press the bulb icon to toggle light

Last Event: Light On
Source: 6f:f7:a8:fe:ff:57:0b:00

6 Next Steps

*UG305: Dynamic Multiprotocol User's Guid*e contains details on the functionality underlying dynamic multiprotocol applications. It also describes how the Radio Scheduler, a key component of the dynamic multiprotocol solution, works.

The source code for the smartphone application is also available. Contact Technical Support if you are interested.

6.1 Zigbee/Bluetooth

The EmberZNet SDK contains the code used to produce the demo images. *UG305: Dynamic Multiprotocol User's Guid*e contains a summary procedure for these examples, and *QSG106: Getting Started with EmberZNet* describes in detail how to generate, build, and flash example code. UG305 also describes how to change configuration settings to make an EmberZNet application into a dynamic multiprotocol application, and contains details on the functionality underlying this demo.

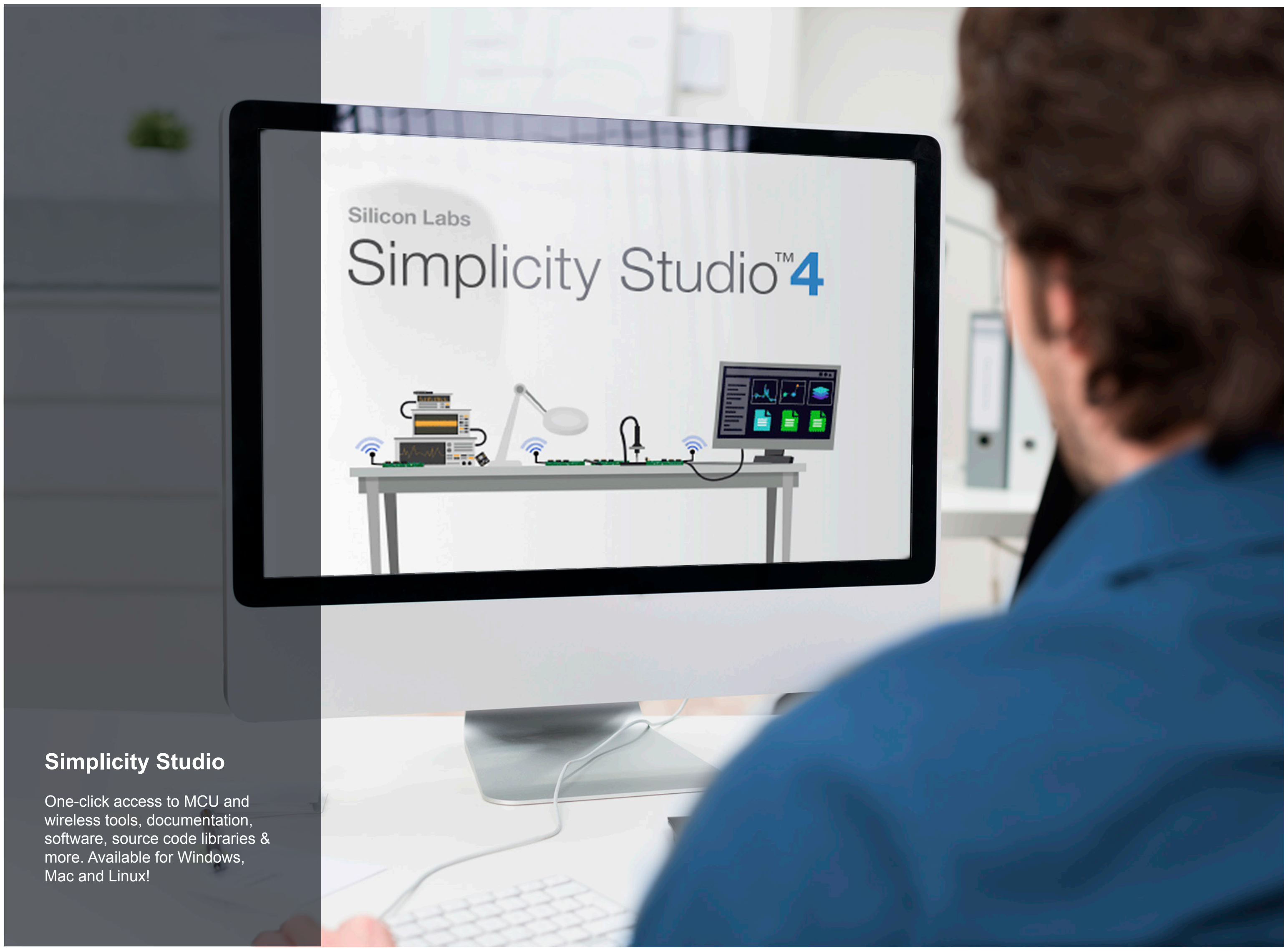
Note that you must download not only the EmberZNet SDK (version 6.0.0.0 or higher) but also the Bluetooth SDK (version 2.6.0.0 or higher) and the Micrium OS kernel in order to fully explore the examples. You must also install and use IAR-EWARM as your compiler.

The Zigbee dynamic multiprotocol solution is currently only supported for SoC architectures (not NCP) and always-on devices. Support for NCP architecture and sleep mode support are not yet available. Please contact Silicon Labs Sales for more information on our multi-protocol software roadmap.

6.2 RAIL/Bluetooth

The Flex SDK (version 2.2.0.0 or higher) contains the code used to produce the Switch demo image. The Bluetooth SDK (version 2.8.0.0 or higher) contains the code used to produce the RAIL/Bluetooth dynamic multiprotocol light example. *UG305: Dynamic Multiprotocol User's Guid*e contains summary procedures for these examples, and *QSG138: Getting Started with the Silicon Labs Flex SDK for the Wireless Gecko (EFR32™) Portfolio* and *QSG139: Bluetooth Development with Simplicity Studio*), respectively, describe in detail how to make and flash example applications.

Note that you must download not only the Flex and Bluetooth SDKs, but also the Micrium OS kernel in order to fully explore the examples.



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 and Community
community.silabs.com

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

http://www.silabs.com